The Journal of Supercomputing (2022) 78:13145–13204
https://doi.org/10.1007/s11227-022-04338-1

# Classification of resource management approaches in fog/edge paradigm and future research prospects: a systematic review

**Puneet Kansal[1] · Manoj Kumar[1] · Om Prakash Verma[1]**

Accepted: 25 January 2022 / Published online: 16 March 2022


## Abstract
The fog paradigm extends the cloud capabilities at the edge of the network. Fog computing-based real-time applications (Online gaming, 5G, Healthcare 4.0, Industrial IoT, autonomous vehicles, virtual reality, augmented reality, and many more) are growing at a very fast pace. There are limited resources at the fog layer compared to the cloud, which leads to resource constraint problems. Edge resources need to be utilized efficiently to fulfill the growing demand for a large number of IoT devices. Lots of work has been done for the efficient utilization of edge resources. This paper provided a systematic review of fog resource management literature from the year 2016–2021. In this review paper, the fog resource management approaches are divided into 9 categories which include resource scheduling, application placement, load balancing, resource allocation, resource estimation, task offloading, resource provisioning, resource discovery, and resource orchestration. These resource management approaches are further subclassified based on the technology used, QoS factors, and data-driven strategies. Comparative analysis of existing articles is provided based on technology, tools, application area, and QoS factors. Further, future research prospects are discussed in the context of QoS factors, technique/algorithm, tools, applications, mobility support, heterogeneity, AI-based, distributed network, hierarchical network, and security. A systematic literature review of existing survey papers is also included. At the end of this work, key findings are highlighted in the conclusion section.

**Keywords** Resource management in fog or edge computing · Resource scheduling · Application placement · Load balancing · Resource allocation · Resource estimation · Task offloading · Resource provisioning · Resource discovery · Resource orchestration

✉ Puneet Kansal
puneet.coe@gmail.com

[1] Delhi Technological University, Delhi, India

## 1 Introduction

Fog computing architecture is a combination of distributed resources spread in-between cloud to end devices that provide services of storage, computing, networking, and control near to the origin of data [1]. The efficient management of fog resources can significantly improve the quality of experience (QoE) while minimizing the cost for the service providers and end-users [2].

In this paper, we will use the terms edge and fog interchangeably, although the computing industry distinguishes between edge and fog computing [3]. Edge paradigm consists of network devices that are usually one hop away from the data generation sources, while fog paradigm is spread in-between end devices to cloud data centers. We can say that edge computing is a subset of fog computing as shown in Fig. 1.

IoT devices such as sensors, smart CCTV cameras, smart gadgets, and other smart devices are continually growing at a very fast pace. According to International Data Corporation (IDC), 41.6 billion connected IoT devices will generate 79.4 zettabytes (ZB) of data in 2025. The current centralized cloud architecture was not designed to handle such a huge amount of data. Fog is an extension of the cloud which provides resources at the network edge. Although fog resources are limited in size as compared to cloud resources, they can play a very significant role in processing big data for real-time applications [4, 5]. Emerging 5G technology, autopilot cars, healthcare 4.0, industry 4.0 are just a dream without the realization of fog/edge computing. As the number of real-time applications is growing

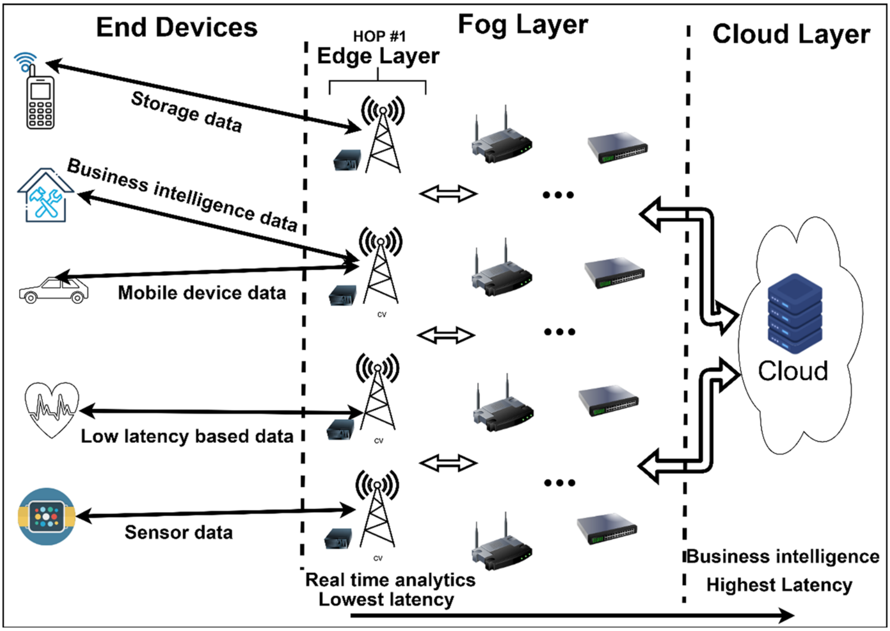


**Fig. 1** Three-tier edge/fog paradigm

and resources at the fog layers are limited, this becomes essential to manage fog resources efficiently.

There are many challenges and issues in resource management at the fog layer due to the heterogeneous, dynamic, and distributed nature of fog devices [6]. Unlike cloud resources, fog nodes are highly heterogeneous—different edge devices have different processor architectures [7]. Task scheduling, application placement, and load balancing approach used in cloud architecture cannot be applied directly on the fog network due to the highly heterogeneous nature of fog nodes. The fog layer helps in executing low-latency-based applications by providing resources near to the end-user. But unlike cloud computing, fog nodes are resource constraints—micro-data centers, routers, switches, and gateways have limited capabilities. Therefore, optimally allocating fog resources is one of the most tedious tasks. However, edge resources have more storage and processing capabilities than end devices. IoT devices usually have low battery, limited computation, and storage capacity, especially to support real-time applications such as online gaming, virtual reality, and augmented reality-based applications [8]. Therefore, efficiently offloading end-user tasks to the appropriate fog node is a must, to realize the proper working of these applications. One another big issue with resource management in fog computing is the dynamic nature of fog nodes and end devices (workload keeps changing and the number of fog nodes does not remain static in a fog network) [9]. However, it does not mean fog nodes and end devices cannot be static. Resource provisioning at the fog layer helps in resource scaling for these dynamic resource requirements. Even end devices can be mobile, which adds complexity to the resource management and orchestration in fog computing. Usually, routers, switches, and mini servers at mobile stations are used as fog nodes. Identifying new fog nodes and dead nodes is a tedious job needed to perform in real time. Fog nodes are spread all over the network, and discovering these distributed nodes is another big challenge.

This work divided the resource management approaches based on data-driven approaches into 9 subcategories: resource scheduling, application placement, load balancing, resource allocation, resource estimation, task offloading, resource provisioning, resource discovery, and resource orchestration. After dividing the research papers into these categories, we have rigorously reviewed them based on the following points: QoS factors, technique/algorithm, tool/language, application area, mobility, security, heterogeneous network, distributed network, and some others. The QoS-based analysis is included for all these resource management approaches. The most common QoS factors considered in the reviewed articles are latency/delay/response time, monetary cost, computational complexity/convergence, resource utilization (memory/CPU/bandwidth), deadline awareness, energy consumption, reliability/fault tolerance, throughput, scalability, security, and mobility. Note, many of the articles used multiple techniques/algorithms in their proposed work. Similarly, almost every reviewed article worked on improving multiple QoS factors.

Research on resource management in fog/edge computing is still in its early stage. Most of the literature relies on mathematical proofs and simulation rather than using practical experimental test bed validation of their work.

Commonly used acronyms used in this paper are listed in Table 1. Some other resource management approach-specific acronyms are mentioned below in comparative analysis (Table 3).

### 1.1 Brief description of QoS factors in fog computing [10, 11]

This subsection provided a brief description of the various QoS factors: monetary cost [12, 13], computational complexity, latency/delay [14, 15], resource utilization [16, 17], mobility [18, 19], energy consumption [20, 21], reliability [22, 23], throughput, scalability [24], deadline [20, 25], security, and privacy [26].

#### 1.1.1 Resource utilization

There are many types of resources in the fog paradigm which need to be utilized efficiently. These resources include network bandwidth, edge server resources: CPU, memory, and secondary storage [14, 27]. Although fog devices are resource-rich as compared to end devices, they cannot fulfill a huge number of growing IoT devices demands [5].

#### 1.1.2 Scalability

In the distributed fog paradigm, scaling in/out the resources as per user application demand is one of the most prominent areas of research [28]. Existing work focused

**Table 1** List of acronyms

| Acronym | Description | Acronym | Description |
|---|---|---|---|
| AI | Artificial intelligence | LaO | Lagrangian optimization |
| AIDL | Android interface definition language | LO | Lyapunov optimization |
| AMPL | Algebraic modeling language | MEC | Multi-access edge computing |
| AR | Augmented reality | ML | Machine learning |
| CO | Convex optimization | MM | Mathematical model |
| CPS | Cyber physical system | PL | Programming language |
| DL | Deep learning | QoE | Quality of experience |
| DRL | Deep reinforcement learning | QoS | Quality of service |
| EC | Edge computing | RL | Reinforcement learning |
| ED | End devices | SDN | Software-defined networking |
| EN | Edge node | SLA | Service-level agreements |
| FC | Fog computing | SUMO | Simulation of urban mobility |
| GTA | Graph-theoretic approach | UAV | Un-manned aerial vehicle |
| IIoT | Industrial internet of things | VM | Virtual machine |
| IoT | Internet of things | VR | Virtual reality |
| IoV | Internet of Vehicles | | |

**Table 2** Area-wise contribution and limitations of literature work

| Area of survey | Citation | Contribution | Limitation |
|---|---|---|---|
| Fog | [367] | Review architectures, infrastructure, and algorithms | No comparative analysis |
| | [56] | Review architecture and algorithms and cover research challenges | Very briefly described algorithmic dimensions, reviewed fewer articles |
| | [368] | Review many resource management approaches in fog/edge computing | Detailed classification is not provided for various resource management techniques |
| | [369] | Explore fog paradigm in the context of IoT devices and their applications. Further, they included an overview of IoT hardware and software platforms | IoT-specific fog computing survey. No technique and tool-based comparative analysis on is done |
| Edge | [336] | Surveys on mechanisms for exploiting resource elasticity features and stream processing engines. Also, outlines future directions | Limited to resource elasticity and dynamic resource provisioning in the context of resource management |
| | [370] | Classify relevant aspects as resource location, resource type, resource use, and resource management objective. Further, identify several research gaps | Do not review from the perspective of techniques/algorithms and tools used |
| | [57] | Review articles in fog computing, cloudlets, and MEC | Review very few articles related to resource management |
| Resource scheduling | [54] | Analyzed and reviewed architecture, algorithms, and techniques. Classification based on static/dynamic approach. Identified open issues and challenges | Most of the reviewed articles considered belong only to heuristic and meta-heuristic techniques |
| | [55] | Categorized into two major groups, including heuristic and meta-heuristic | Review limited to heuristic and meta-heuristic approaches |
| | [371] | Reviewed resource scheduling articles in the context of techniques, performance indicators, applications, and mode of operation such as centralized and decentralized | Tools used in literature are not discussed |

**Table 2** (continued)

| Area of survey | Citation | Contribution | Limitation |
|---|---|---|---|
| Application placement | [348] | They provide Taxonomy, Review, and Future Directions | Specific to application placement, architecture and maintenance |
| | [372] | A classification of large-scale, heterogeneous, and geo-distributed systems; Also, discuss open challenges and provide future research prospects | Specific to service placement |
| | [373] | Reviewed the AI-based resource placement articles. They categorize the literature on the basis of evolutionary algorithms, machine learning, and combinatorial algorithms. Further, they reviewed the application placement articles based on security | Specific to AI-based application placement in edge computing. Tools used in literature work are not discussed |
| Load balancing | [374] | Discussed year-wise static and dynamic load balancing techniques. Further, they discuss load balancing specific performance measurement factors. They also proposed a load balancing architecture | Specific to load balancing and do not include the year 2021 articles |
| | [375] | Review based on SDN for load balancing approach; Provide future research directions | No comparative analysis of reviewed articles was provided |
| Task offloading | [376] | Classification based on the technique used; Detailed discussion on different classification techniques | Specific to computation offloading |
| | [377] | Discuss offloading framework and characteristics of offloading algorithms | No detailed comparison of offloading techniques |
| | [378] | Discuss middleware technologies, Analysis of offloading approaches, and future research challenges | Do not discuss proposed algorithms and used tools |
| | [379] | Detailed discussion on computation offloading and case studies | No classification-based comparison of task offloading techniques |
| | [380] | Review and classification of stochastic offloading mechanisms and future challenges | Specific survey to stochastic-based offloading |
| Resource Provisioning | [381] | Taxonomy study of machine learning-based resource provisioning, provide Summary of techniques, proposals, and tools/methods | Do not discuss performance parameters |

**Table 2** (continued)

| Area of survey | Citation | Contribution | Limitation |
|---|---|---|---|
| Orchestration Architecture | [43] | Orchestration architecture, challenges, and tools | No classification of research articles on the basis of algorithms used |
| | [382] | Fog orchestrator architectures, agent, comparative analysis of fog orchestration architecture | Do not cover the latest research articles |
| Container Orchestration | [383] | Taxonomy of container technology, container tools, architecture | Generalize discussion on container technology, not specific to edge computing |
| MEC/Fog/Cloudlet-based Orchestration | [384] | A Survey on end–edge–cloud orchestration paradigms: fog computing, MEC, Cloudlet, and transparent computing | No detailed review and comparison of articles |

on static and dynamic scaling. Efficient resource scaling includes finding a minimal number of fog nodes to satisfy the user demand [29].

#### 1.1.3 Delay/latency

Delay or latency time includes "time taken to send the data to fog node" + "data processing time at fog node" + "getting the response back to the user" [30, 31]. However, many existing papers do not consider data processing time as total delay time [32–37]. Many papers use terms delay, latency, response time, execution time, completion time, and processing time interchangeably. In this paper, we combined these terms altogether in one category.

#### 1.1.4 Throughput

This is the number of tasks processed by the fog paradigm per unit time. The throughput of the fog paradigm usually depends on the nearest fog node selection process.

#### 1.1.5 Monetary cost

This is the price need to be paid by the end-user to get the services from the service provider. The cost of the system depends on the urgency of the data processing need. Real-time processing applications users need to pay more as service providers need to contact third-party fog device owners.

#### 1.1.6 Computational complexity

This is defined as the time taken by the proposed technique to execute. This means we deal with the convergence speed of the proposed algorithm. Reinforcement learning is one of the most popular techniques used to solve resource management problems but leads to higher computational complexity. In this text, reviewed articles that provided fast convergence were also added in this category.

#### 1.1.7 Energy consumption

Most of the IoT devices are energy constraints; that is why they upload high-computational tasks on the fog layer. Resources such as bandwidth, CPU, and memory consume the energy [38].

At the fog layer, not all fog nodes are connected to the main power line. Usually, fog devices up to the first hop of the network get a direct power supply, upper layers still are battery dependent. Therefore energy consumption deals with the efficient utilization of energy at the fog layer. Green energy consumption-based research is also getting momentum these days [39, 40].

#### 1.1.8 Reliability

In case of failure of a dependent fog node, how the proposed system recovers from it without impacting QoE, defines the reliability of the system [41, 42]. In this text, we also consider fault tolerance, in the context of reliability.

#### 1.1.9 Deadline

Many applications need deadline-aware service, in which a particular task must be completed within a given deadline [20, 25]. Many papers focused on deadline-aware applications by using efficient task scheduling [43, 44].

#### 1.1.10 Mobility support

In edge computing, both edge nodes, as well as end devices, can be mobile. Mobility deals with keeping the session alive even though mobile users change their point of attachment. Edge servers communicate with each other to share mobile device information. In edge computing,

#### 1.1.11 Security and privacy

Fog devices are nearer to the end devices as compared to the cloud which makes them more trustworthy than the cloud [45]. Although fog devices are widely spread over the fog network and their surface of contact with IoT devices is vast, this makes them vulnerable to security threats. This may not possible to deploy full suit security solutions at resource constraint edge devices. As compared to the cloud, there are fewer security standard certifications and insufficient security designs for edge devices [46]. Many of the edge computing-based applications such as healthcare and smart grids required efficient ways of security and privacy [47–49].

Further, blockchain technology getting popular to handle security issues in the fog paradigm [50, 51]. Fog computing is distributed and decentralized network of resources which can be an ideal place for setting a blockchain-based secure decentralized data management platform [52, 53].

### 1.2 The main contributions of this review paper are

- We reviewed resource management approaches based on a data-driven approach and provides their brief description with subclassification.
- A systematic literature review of existing survey articles is included in this work.
- Comparative analysis of existing articles based on QoS factors, technique/algorithm, tool/language, and application area is done for every resource management approach.

- Finally, we discussed future research prospects in the field of resource management for fog/edge computing.

The remaining parts of this review work are structured as follows: Sect. 2 discusses relevant literature review in resource management of fog/edge approaches. Section 3 explains the article selection procedure for this review. Section 4 discusses the subclassification for all resource management approaches individually. Section 5 discusses the future research prospects in the context of QoS factors, technique/algorithm, tools, applications, mobility support, heterogeneity, AI-based, distributed network, hierarchical network, and security. In Sect. 6, this paper is concluded.

## 2 Existing survey reviewing and classification

An area-wise literature survey papers review is provided in Table 2. The classification of literature articles is done based on the survey area. A summary of contributions and limitations is included.

### 2.1 Limitations of existing work

- No existing work jointly provides a detailed classification of resource management approaches and comparative analysis [367, 368, 375, 377, 379] as per the best of our knowledge.
- Some works limited their review up to heuristic and meta-heuristic techniques [54, 55].
- There is no state-of-the-art survey paper related to resource estimation, resource discovery and resource provisioning approaches.
- Some of the works reviewed very limited research articles [56, 57].
- We try to cover an in-depth classification of resource management approaches. This survey does not bound to a few techniques and reasonable research articles are included for review.

## 3 Article selection process

Initially, the literature articles were selected based upon the defined keywords. Then articles are shortlisted based upon the title. Then a final list of research articles was selected after the double screening of abstracts and full articles. The article selection process is provided in Fig. 2. Research articles from highly reputed journals are included in this review which includes IEEE, ACM, Elsevier, Springer, Emerald, Wiley, Hindawi, and MDPI. Total 298 articles [2, 12–15, 18–23, 27, 29–36, 39–41, 44, 58–331] from the last 6 years [2016–2021] were included in this paper.

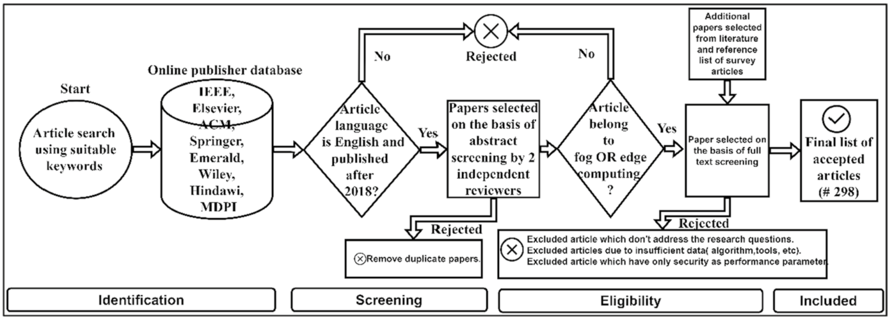


**Fig. 2** Research article selection process

## 4 Categories of resource management approaches

As compared to the cloud, a fog network is more resource constraint. The efficient utilization of these limited resources is a big challenge in such a heterogeneous and dynamic fog environment. Selecting the optimal fog node for application placement, scheduling the resources in-between running jobs, load balancing in-between the available resources, estimating the user resource requirement in advance, allocating the required resources to a specific task, resource monitoring, resource provisioning, resource orchestration, and appropriate task offloading, are main issues in fog/edge resource management. Based on these issues, we divided the resource management approaches into 9 different categories as shown in Fig. 3. A systematic comparative analysis of reviewed articles based on technique, tools, and application area is provided in Table 3. Further, QoS-based categorization of reviewed articles is provided in Table 4.

Further, a broad subclassification is provided for all these resource management approaches.

### 4.1 Resource scheduling

Resource scheduling helps in minimizing the execution time of end-user tasks by allocating optimal fog nodes for specific types of jobs such that QoS can be achieved while maintaining SLA (service-level agreement). A good scheduler should be general-purpose, efficient, fair, transparent, and dynamic.

Resource scheduling approaches used in cloud architecture cannot be applied directly to the fog network due to the highly heterogeneous nature of fog nodes [332]. This heterogeneous network makes resource scheduling an NP-hard optimization problem [333, 334]. This is because heuristic and meta-heuristic algorithms are popularly used for solving resource scheduling problems in fog computing.

Scheduling plays a very important role in the successful deployment of tasks on available resources as it exhibits communication patterns over distributed nodes. It is a process of trading communication costs against resource utilization while making ensure that the host machine has enough resources for tasks placed on it. This is

**Table 3** Comparative analysis of fog resource management articles

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [58] | Heuristic | CloudSim | Healthcare | [194] | AI-based | N/A | Non-specific |
| [59] | | C# | IoT | [195] | Mathematical model | MATLAB | Non-specific |
| [60] | | Python | Non-specific | [196] | | Cloudsim + java | IoT |
| [61] | | iFogSim | Non-specific | [41] | Heuristic | N/A | Healthcare |
| [62] | Meta-heuristic | C + + | Non-specific | [197] | AI-based (DL) | N/A | VR |
| [63] | | VB 6.0 | IoT | [198] | | Python (keras) | Non-specific |
| [64] | | C# | Industrial | [199] | | N/A | Non-specific |
| | | | | [200] | AI-based (DL) + B&B | CPLEX | Smart home |
| [65] | | CloudSim + MATLAB | Non-specific | [201] | AI-based (DRL + ML) | Python (TensorFlow) | Non-specific |
| [66] | | C# | Industrial | [202] | AI-based (DRL) | N/A | Healthcare |
| [67] | Fuzzy Logic approach + Meta-heuristic | MATLAB | Non-specific | [203] | | N/A | Vehicular |
| [68] | Meta-heuristic | MATLAB | Industrial | [204] | | Python (PyTorch + CUDA) | Non-specific |
| [69] | Mathematical model | N/A | Non-specific | | | | |
| [70] | | iFogSim + AMPL | Non-specific | [205] | | N/A | vehicular |
| [71] | Approximation | MATLAB | Non-specific | [206] | | N/A | Non-specific |
| [72] | Mathematical model | N/A | Non-specific | [207] | | Python (TensorFlow) | Non-specific |
| [44] | | N/A | Non-specific | [208] | | Python (PyTorch) | Non-specific |
| [73] | | MATLAB | Smart city | [20] | | Python (TensorFlow) | Video analytics |
| [74] | | MATLAB + YALMIP + CPLEX | IoT | [40] | | MATLAB | IoV |
| [75] | | iFogSim | IoT | [209] | AI-based (ML) | Linux perf | Healthcare |
| [76] | | N/A | Vehicular | [210] | | Contiki Cooja | Non-specific |
| [77] | | N/A | IoV | [211] | AI-based (RL) | N/A | UAV |
| [78] | | Java + CPLEX | Video surveillance | [212] | | N/A | Non-specific |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [79] | Heuristic | Gurobi | Image processing | [213] | | N/A | Solar and wind energy harvesting |
| [80] | Mathematical model | PEPA | Vehicular | [214] | | N/A | Non-specific |
| [81] | Heuristic | iFogSim | Video surveillance | [19] | | MATLAB | Industrial |
| [82] | Heuristic | N/A | Healthcare | [215] | | N/A | UAV |
| [83] | Heuristic | N/A | Non-specific | [216] | | Python (TensorFlow) | VR + Healthcare |
| [84] | | N/A | Non-specific | [217] | | MATLAB | Non-specific |
| [85] | Meta-heuristic | MATLAB | Word Count | [23] | | Python + C + + | Industrial |
| [86] | | iFogSim | IoT | [218] | Fuzzy Logic approach | EdgeCloudSim | AR + Healthcare |
| [87] | | Cloudsim | Non-specific | [219] | Game theory | N/A | Face recognition |
| [88] | | java | IoT | [220] | | N/A | Non-specific |
| [89] | | java | IIoT | [221] | | N/A | Non-specific |
| [90] | | iFogSim | Video surveillance | [222] | | N/A | Non-specific |
| [91] | | C + + | IoT | [223] | | N/A | Non-specific |
| [92] | | N/A | Vehicular | [224] | | N/A | IoT |
| [93] | | iFogSim | IoT | [225] | | Java + SUMO | Industrial + UAV + Vehicular |
| [94] | | N/A | Non-specific | [226] | | N/A | Smart city + UAV |
| [95] | | N/A | Industrial | [227] | | N/A | IoT |
| [96] | | iFogSim | Non-specific | [228] | | N/A | IoT |
| [97] | | Python | Vehicular | [229] | | N/A | UAV |
| [98] | | C# | Non-specific | [230] | Game theory + Mathematical model (CO) | N/A | UAV |
| [99] | Fuzzy Logic approach + Meta-heuristic | iFogSim | Smart city | [231] | Game theory + Mathematical model (LaO) | N/A | Vehicular |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [100] | AI-based (RL) | N/A | Non-specific | [232] | Graph-theoretic approach | MATLAB | Face recognition |
| [101] | AI-based (DRL) | SimPy | IoT | [233] | | N/A | Non-specific |
| [102] | AI-based (DL) | iFogSim | IIoT | [234] | | FogWorkflowSim | UAV |
| [103] | AI-based (ML) | Cloudsim | Non-specific | [235] | Heuristic | QualNet | Non-specific |
| [104] | AI-based (DRL) | N/A | Non-specific | [236] | | N/A | Non-specific |
| [18] | AI-based (DL) | MATLAB + Python | IoT | [21] | | MATLAB | Non-specific |
| [105] | AI-based (DRL) | Python (Pytorch) | IoT | [237] | | N/A | Non-specific |
| [106] | Heuristic | N/A | Non-specific | [238] | | C++ | Non-specific |
| [107] | Meta-heuristic | N/A | IoT | [239] | | C++ | Non-specific |
| [108] | Framework | N/A | Non-specific | [240] | | N/A | Non-specific |
| [109] | Framework | Java | Healthcare | [241] | | MATLAB + Python | UAV |
| [2] | Framework | D-ITG traffic generator | IoT | [242] | | SUMO + OMNeT++ | Vehicular + Image recognition + Video retrieval |
| [110] | Architecture | VB.NET | Robotics | [243] | | N/A | IIoT |
| [27] | AI-based (DRL) | N/A | 5G | [244] | | MATLAB | Smart city |
| [14] | AI-based (ML) | N/A | Telecom | [245] | | N/A | Non-specific |
| [111] | AI-based (RL) | N/A | Non-specific | [246] | | MATLAB | Vehicular |
| [112] | Architecture | iFogSim | Smart city | [247] | Mathematical model | N/A | Non-specific |
| [113] | Framework | iFogSim | Transportation | [248] | | JAY | Image recognition |
| [114] | Fuzzy logic approach | iFogSim | Smart phone | [249] | | Gurobi + AMPL | Non-specific |
| [115] | Heuristic | Gurobi | Non-specific | [250] | | N/A | Vehicular |
| [30] | | FogTorch | Fire detection | [251] | | N/A | IoT |
| [39] | | iFogSim | Healthcare | [252] | | CPLEX | Vehicular |
| [116] | | Tsung | IoT | [253] | | EdgeCloudSim | IoT |
| [117] | | N/A | Smart vehicles | [254] | | Java | Vehicular |
| [118] | | N/A | Video streaming | [255] | Mathematical model (LaO + CO) | N/A | Non-specific |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [119] | | iFogSim | Smart home | [256] | Mathematical model (LaO) | N/A | Vehicular |
| [120] | | Gurobi | CPS | [257] | | N/A | IoT + face recognition |
| [121] | | MATLAB | Transportation | [258] | | MATLAB | Smart home |
| [12] | | N/A | Non-specific | [259] | Mathematical model (LO) | SimPy | Non-specific |
| [122] | | iFogSim | Smart Home | [260] | | N/A | Non-specific |
| [123] | | N/A | Non-specific | [261] | | N/A | Non-specific |
| [124] | Mathematical model | N/A | Smart city | [262] | | Gurobi + MATLAB | Non-specific |
| [125] | | iFogSim | Non-specific | [263] | | N/A | IoT |
| [126] | | MATLAB | IoT | [264] | | N/A | Non-specific |
| [127] | | MATLAB + Gurobi | Non-specific | [265] | | N/A | Non-specific |
| [128] | | iFogSim | IoT | [266] | | N/A | Non-specific |
| [129] | | Kotlin | Non-specific | [267] | | Java | Non-specific |
| [130] | Meta-heuristic | N/A | Industrial | [32] | | N/A | Non-specific |
| [131] | | iFogSim | Industrial | [268] | | N/A | IoT |
| [132] | | iFogSim | IoT | [269] | Mathematical model (LO) + AI-based (DRL) | N/A | Non-specific |
| [133] | Heuristic | N/A | Non-specific | [33] | Mathematical model (CO) | MATLAB | UAV |
| [29] | AI-based (DRL) | Python (Tensorflow) | Non-specific | [270] | | N/A | Non-specific |
| [134] | AI-based (RL) | MATLAB | Healthcare | [271] | | N/A | IoT |
| [135] | Graph-theoretic approach | Hadoop | IoT | [272] | | N/A | Vehicular |
| [136] | Graph-theoretic approach | N/A | Vehicular | [273] | | N/A | Non-specific |
| [137] | Heuristic | Gurobi | Healthcare | [274] | | N/A | IoV |
| [138] | | CloudSim | Non-specific | [275] | | N/A | Non-specific |
| [22] | | N/A | Smart city | [276] | | N/A | Non-specific |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [139] | | CloudSim | IoT | [277] | | N/A | Non-specific |
| [140] | | NS2 | Vehicular | [278] | | N/A | Non-specific |
| [141] | | iFogSim | IoT | [279] | | N/A | Non-specific |
| [142] | | N/A | Face recognition | [280] | | N/A | Non-specific |
| [143] | Mathematical model | MATLAB | UAV | [281] | | N/A | UAV |
| [144] | Meta-heuristic | CloudAnalyst | IoT | [282] | | N/A | Non-specific |
| [145] | | N/A | IoV | [283] | Mathematical model (CO) + Heuristic | N/A | Healthcare |
| [31] | | MATLAB | Healthcare | [284] | Mathematical model (CO) + Approximation | N/A | Non-specific |
| [146] | | N/A | AR + Healthcare | [34] | Mathematical model (CO) | Python (PyOpt) | Non-specific |
| [147] | Heuristic | Cloudsim | IoT | [285] | Mathematical model (SGD) | N/A | UAV |
| [148] | Game theory | MATLAB | Non-specific | [286] | Meta-heuristic | CPLEX | vehicular |
| [149] | AI-based (DL) | iFogSim | Non-specific | [35] | | N/A | Non-specific |
| [150] | | N/A | Non-specific | [287] | | Python (TensorFlow) | Non-specific |
| [151] | | MATLAB | Non-specific | [288] | | Python | Non-specific |
| [152] | AI-based (DRL) | N/A | IoT | [289] | | N/A | Non-specific |
| [153] | | Python | Non-specific | [290] | | N/A | IoT |
| [154] | | N/A | AR | [291] | | AIDL | Non-specific |
| [155] | | Python (TensorFlow) | UAV | [292] | | iFogSim | Non-specific |
| [156] | | Python (TensorFlow + Keras) | vehicular | [36] | | N/A | IoT |
| [157] | AI-based (RL) | N/A | IoT | [293] | | N/A | Non-specific |
| [158] | | N/A | IIoT | [294] | Meta-heuristic + B&B | N/A | Non-specific |
| [159] | Approximation | N/A | Non-specific | [295] | AI-based (RL) | N/A | Non-specific |
| [160] | | C + + | IoT | [296] | Heuristic | N/A | Industrial |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [161] | Architecture | CloudSim | IoT | [297] | AI-based (RL) | N/A | IoT |
| [162] | Game theory | Python | Voice recognition + Face recognition + Video game | [298] | AI-based (RL) | iFogSim | Non-specific |
| [163] | | MATLAB | IoT | [299] | Framework | N/A | Online game + face detection |
| [164] | | N/A | Non-specific | [300] | Fuzzy Logic Approach | Apache JMeter | Industrial |
| [165] | | N/A | Non-specific | [301] | Fuzzy logic Approach + Meta-heuristic | MATLAB | Non-specific |
| [166] | | CPLEX | Non-specific | [302] | Heuristic | Apache JMeter | Online game |
| [167] | Game theory + Mathematical model (CO) | N/A | IoT | [303] | | Gurobi | Transportation |
| [168] | Graph-theoretic approach (Petri nets) | N/A | Non-specific | [304] | | iFogSim | Non-specific |
| [169] | Heuristic | N/A | Non-specific | [305] | | CloudSim | Non-specific |
| [13] | | CloudSim | Voice and video processing | [306] | | CloudSimSDN | Wikipedia web application |
| [170] | | iFogSim | Non-specific | [307] | Mathematical model | Java | IoT |
| [171] | | N/A | Vehicular | [308] | | N/A | Smart city |
| [172] | | N/A | IoV | [309] | | CPLEX + Java | Smart City |
| [173] | | Python | Smart City | [310] | Mathematical model (CO) | N/A | Traffic models |
| [174] | | CloudSim | Non-specific | [311] | Mathematical model (LO) | N/A | IoT |
| [175] | | MATLAB | Healthcare | [312] | | N/A | Non-specific |

**Table 3** (continued)

| Citation | Technique/algorithm | Tool/Lang | Application area | Citation | Technique/algorithm | Tool/Lang | Application area |
|---|---|---|---|---|---|---|---|
| [176] | Mathematical model | Java | Vehicular | [313] | Framework | AnyLogic | Vehicular |
| [177] | | MATLAB | IoT | [314] | | iFogSim | IoT |
| [178] | | N/A | IoV | [315] | Heuristic | MATLAB | Non-specific |
| [179] | | N/A | Non-specific | [316] | Framework | N/A | Non-specific |
| [180] | | N/A | UAV | [317] | Heuristic | WorkFlowSim | Image processing |
| [181] | Mathematical model (CO) | N/A | Vehicular | [318] | Architecture | N/A | Non-specific |
| [182] | | MATLAB | IoT | [319] | 802.11 beacon-based protocol | Python | Non-specific |
| [183] | | N/A | Non-specific | [320] | Architecture | N/A | IoT |
| [184] | Mathematical model (LO) | N/A | IoT | [321] | Framework | SUMO + C + + | Vehicular |
| [15] | | N/A | Non-specific | [322] | Container orchestration | N/A | IoT |
| [185] | | MATLAB | Video streaming | [323] | | MATLAB | IIoT |
| [186] | Mathematical model (LO) + AI-based (DRL) | MATLAB | UAV | [324] | | N/A | Vehicular |
| [187] | Meta-heuristic | N/A | Non-specific | [325] | AI-based (DRL) | N/A | Transportation |
| [188] | | MATLAB | IoT | [326] | Approximation | N/A | industrial |
| [189] | | MATLAB | IoT | [327] | Fuzzy Logic approach | EdgeCloudSim | Healthcare + AR |
| [190] | | MATLAB | IoT | [328] | Graph-theoretic approach | N/A | Vehicular |
| [191] | Game theory | N/A | Non-specific | [329] | Mathematical model | N/A | AR + Video Analytics |
| [192] | Mathematical model | N/A | Non-specific | [330] | | EdgeCloudSim | Non-specific |
| [193] | Meta-heuristic | N/A | Industrial | [331] | Mathematical model (ADMM + CO) | N/A | AR + Video Analytics |

PEPA: Performance Evaluation Process Algebra; AMPL: A Mathematical Programming Language

**Table 4** QoS-based analysis of resource management articles

| | Resource scheduling | Application placement | load balancing | Resource allocation | Task offloading | Resource orchestration /provisioning/discovery /estimation |
|---|---|---|---|---|---|---|
| Monetary cost | [2,18,60,74,76,86,89,93,109] | [12,14,113,115,125,129,130] | [137,144] | [160,162,163,164,165, 166,167,168,169, 170,174,176,185, 187,189] | [200,205,214,220,237,225, 227,252,259,265,267,272, 273, 291,294] | [326,331] / [295,297,300,301, 303,305,308,310,311,314, 315] / NA / [194,196] |
| Computational complexity | [58,61,62,63,65,67,69,44,74, 79,92,93,94,96,106,153,156] | [169,174,177,178,184,185,189, 191] | [31,146,148,192,193,196,198, 199] | [152,153,156,159,166, 173,175,176,181, 182,183,184,185, 186,190] | [34,35,36,200,201,202,204, 208,209,211,212,215,218, 219,221,223,224,227,228 229,230,240,241,243,255, 261,262,264,269,270, 273,275,277,278,280, 281,282,283,284,293] | [327,328,329,331] / [295,299, 301,303,304,308,310,314, 315] / NIL / [194] |
| Latency/ delay | [2,18,44,58,59,62,65,70,71,72, 73,75,77,79,80,81,85,86,90, 91,93,94,95,99,100,101,102, 103,105,106,107,108] | [12,14,27,30,39,111,112,113,114, 116,117,118,119,121,122, 124,125,126,127,128,129,130, 131,132] | [22,29,31,133,134,136,137,138, 140,141,142,144,145,148] | [13,15,149,152,153, 154,155,156,157,162, 163,164,168,169,172, 173,174,175,178,180, 181,183,184,185,186, 188,190] | [19,20,21,23,32,40,197,198, 199,202,203,205,206,207, 208,210,211,212,213,214, 215,217,218,221,222,223, 225,228,229,230,232,235, 236,237,239,240,241,242, 243,244,247,248,249,250, 251,252,253,254,255,256, 257,258,259,261,262,267, 269,270,271,273,274,275, 276,278,279,282,283,284, 285,286,287,289,290,291, 292,293,294,315] | [322,323,324,325,326,327,331] / [295,296,297,298,299,300, 301,302,303,304,305,306, 307,308,309,310,311,312,313, 314] / [317,318,319,320,321] / [191,192,193,194] |
| Resource utilization | [2,58,60,62,64,67,68,70,44,73, 75,76,77,79,81,82,83,93,98, 99,101,103,109] | [30,39,113,114,115,116,117,118, 119,121,122,123,126,128,131] | [134,135,139,140,141,146,147] | [13,153,157,159,161, 165,167,169,170,182, 185,190] | [34, 36, 197,215,218,226,231, 233,237,242,246,247, 253,254,260,263,268,272, 289,291] | [322,323,324,327,328,329,330 331] / [298,307,311,312,314] / [318] / [192,193,194,196] |
| Deadline | [64,65,66,70,87,91,92,101, 107,109] | [113,119,125] | NA | [151] | [33,201,234,246,247,248, 252,272] | NIL / [305,311] / NIL / NIL |

**Table 4** (continued)

| | Resource scheduling | Application placement | load balancing | Resource allocation | Task offloading | Resource orchestration /provisioning/discovery /estimation |
|---|---|---|---|---|---|---|
| Energy consumption | [2,59,68,69,70,71,72,75,77,82,84,87,88,89,95,96,98,101,102,104,106] | [39,112,113,120,122,127,128,132] | [140,143] | [13,150,151,152,154,155,156,159,171,172,175,176,177,178,179,180,182,184,185,186,188] | [19,20,21,23,32,33,34,35,197,198,200,201,202,204,206,207,208,209,211,212,213,216,221,222,225,228,229,230,232,233,234,235,236,237,238,239,240,241,244,247,248,251,255,256,258,261,262,263,264,265,266,268,270,271,273,275,276,277,278,279,281,282,283,286,288,289,290,291,292,293,294,315] | [323,326] / [298,303,304,308,313] / [319] / [41] |
| Reliability | [2,18,63,68,77,82,93,101,104,106] | [118,132] | [22,139,143] | [172,178] | [36, 201,202,230,242,252,261315] | [326,327,330] / [296,300,313] / NIL / [193] |
| Scalability | [60,44,96,104] | [112] | [136,146] | [166,169,171,172,177] | [23,199,219,228,245,264] | [323,328,329,331] / [298,299,300,302,304,307,310,314,315] / [320] / NIL |
| Throughput | [2,60,61,70,78,97,105] | NA | [29] | [151,158] | [199,217,218,224,245,246,249,255,263] | [329]/[307]/NIL/[194] |
| Mobility | [62,73,74,76,81,82,83,85,92,93,96,97,99,106] | [12,14,27,39,112,115,123,124,127,128] | [133,134,138,143,145,146,148] | [149,153,156,158,159,160,162,166,171,172,174,179,184184,185,187,188,189] | [19,20,21,34,197,199,202,203,204,206,211,213,214,219,220,221,222,223,234,235,238,239,240,246,247,248,249,254,256,258,261,262,264,265,266,267,272,273,274,275,276,278,281,282,284,285,286,287,289,291,293] | [324,327,329,331]/[297,298,299,302,304,305,308,310,311,313]/[317,318,319,321]/[195,196] |
| Security and privacy | [18,77] | [116,123,128] | NA | [13,159,172,175,188] | [199,210,216,220,234,235,249,272,289] | NIL/[300,304]/[321]/[196] |

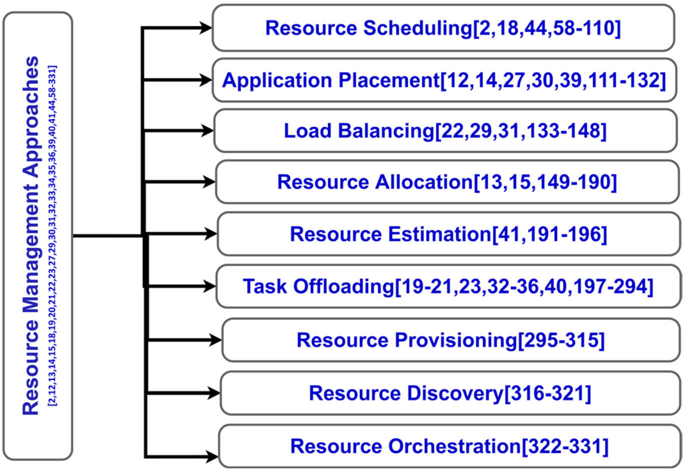


**Fig. 3** Categories of resource management approaches

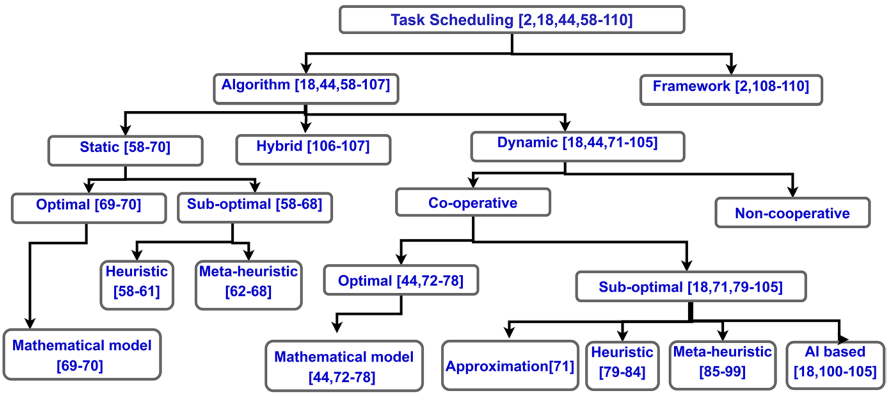


**Fig. 4** Subclassification of the resource scheduling approach

a must in scheduling to minimize resource consumption and to ensure that the workload assignment does not exceed the node's capacity.

Broadly scheduling algorithms are classified into three categories that are static, dynamic, and hybrid as shown in Fig. 4. Static scheduling [58–70] is also known as offline or deterministic scheduling. In static scheduling, the scheduler priorly knows all the scheduling parameters such as the number of tasks to be executed and available resources. However, this is not feasible all the time to have prior knowledge about tasks and resources in such a heterogeneous network like fog. That is why dynamic scheduling is a more appropriate approach for scheduling tasks in a fog environment.

Dynamic scheduling [18, 44, 71–105] is also known as online scheduling or non-deterministic scheduling. In this scheduling approach, there remains no prior knowledge about the user request, and tasks are submitted for execution on runtime. Even though dynamic scheduling looks like a more accurate approach in a fog environment, it is an expensive process to get an update about the available resources.

Hybrid scheduling [106, 107] combines the benefits of both static and dynamic scheduling. Hybrid scheduling is successful because there is not any single technique that can satisfy fog computing.

#### 4.1.1 Optimal versus sub-optimal

Optimal solutions can be obtained by "mathematical formulations." Although mathematical models provide the most accurate solution, they consume lots of time. So optimal solutions are feasible for small-scale problems. Even a very small fog network with few fog nodes, servers, and links can have a large number of optimization variables. On the other hand, sub-optimal algorithms provide approximate solutions to a problem in a reasonable amount of time. As the resource scheduling in a fog network is an NP-hard problem, sub-optimal algorithms are more realistic. Sub-optimal techniques include approximation, heuristic, meta-heuristic, AI-based, and hyper-heuristic algorithms.

#### 4.1.2 Cooperative and noncooperative

In cooperative scheduling, different fog nodes share the status of their resources with other nodes. Based on this information, the scheduler assigns the task to the available resource node. All the schedulers in the fog network work toward a common system optimization goal. Although cooperative scheduling gives stable results, it also increases the overhead of sharing data between nodes. In noncooperative scheduling, the individual scheduler works like an individual entity. Noncooperative schedulers are more effective in a static environment. Overhead is very low in noncooperative scheduling, but these algorithms are not stable.

Lots of recent work focus on dynamic scheduling. Heuristic, meta-heuristic, and AI-based techniques are used most popularly to solve resource scheduling problems.

### 4.2 Application placement

Optimal application placement (task placement or service placement) in a fog network not only enhances user quality of experience (QoE) but also helps in the best utilization of available resources. Placing end-user applications at edge nodes is not an easy task due to the heterogeneous, hierarchical, and distributed nature of the edge network [335]. User's dynamic expectations and end device's diversity also increase the complexity. Therefore application placement approaches developed for cloud architecture cannot be applied directly in fog networks.

Practically end devices themselves decide on choosing a particular node for service placement. Usually, nearby edge nodes are chosen for service placement. In case nearby nodes are found busy, nodes are selected in a distributed or hierarchal way.

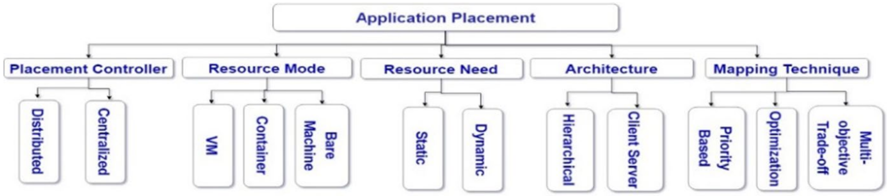


**Fig. 5** Subclassification of application placement approach

The placement controller works as a manager to control application placement activities in fog architecture. This also helps in estimating the waiting time for an application which ultimately drives the overall performance of the system. The placement controller either works in centralized or distributed mode. In centralized mode, the placement controller receives all the resource information from fog devices. Then based on this information appropriate node is assigned to a particular user application as per its resource requirement. In distributed mode, the placement controller works locally at various nodes which helps in obtaining the local optimal solution. The distributed controller is more efficient in the context of bandwidth usage and scaling while the centralized controller provides the best optimal solution but at higher bandwidth cost and low scaling [336]. A centralized controller is mostly used in those networks which have fewer fog nodes. Single point failure is another problem with any centralized system. The broad classification of application placement articles is shown in Fig. 5.

Resources are usually allocated in a virtual mode such as a virtual machine (VM) or container. Transferring the containers and VM is easy from one machine to another. Containers are popularly used these days at the fog layer due to their lightweight and ease of migration. End-user micro-services are broken down at the fog layer to get processed at various machines. Different parts of micro-services can be easy to migrate on distributed machines using containers. Maybe due to high cost (storage + processing), VM technology was least explored in the resource type category for edge applications.

Due to the dynamic nature of applications, this is very difficult to place applications on a fog network. Sharing the resource information at runtime is costly in terms of bandwidth. Simulation tools also lack in implementing dynamic models. Therefore lots of static service placement approaches have been developed in past.

Some of the existing research work prioritizes tasks as per user expectations which helps in improving QoE. Others focused more on resource optimization. A trade-off between QoS and resource optimization is another important aspect of service placement. Some of the articles highlight the compensation-based monetary support in edge computing [125]. Compensation is provided by the fog service provider in case of any type of system failure. Some of the other reviewed articles used the auction-based approach for function placement in fog computing [129]. Auction-based approaches provide SLA-based economic efficiency

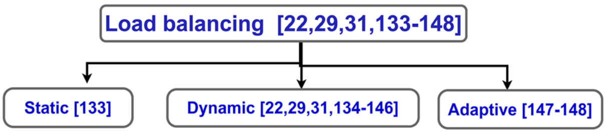


**Fig. 6** Subclassification of the load balancing approach

which provides a practical trade-off between end-user requests and actual services provided.

Many of the reviewed articles used bare machine technology to gain low latency. Bare machine implementation consumes less energy and less memory, and runs faster because OS is not required on the bare machine. Input and output operations can be executed very fast on the bare machine as compared to OS-based systems. Even containers running on a bare machine consume less CPU as compared to VM [337]. However, deploying containers on the bare machine can make migration tasks difficult.

Many of the reviewed articles used container technology for application placement in fog nodes. Container technology has been matured at the cloud level, but it is at the initial stage in fog/edge computing. Containers are lightweighte, best suited for resource constraint edge devices. This is very challenging and opportunistic to use containers for application placement in heterogeneous edge devices.

Container orchestration allows you to deploy a hundred–thousand of the instances of your application with a single command. Lightweight containers can be deployed easily on resource constraint edge nodes. [338] used raspberry Pi as an edge node and deploy Docker-based containers on it. Implementing container-based orchestration in a dynamic edge environment is a challenging task [339]. Although many solutions for deploying containers on fog nodes exist, these solutions do not consider geographical distribution, heterogeneity, and mobility of fog resources [24, 340]. Further, edge node overloading needs to be taken care of while migrating containers/VM from cloud to edge [341]. Finding optimal edge nodes for mobile user container/VMs migration is another challenge in MEC [342]. In addition, optimal accelerator management in containers is not explored sufficiently as compared to VMs [343]. The container migration time in the cloud is a secondary consideration while in fog computing container migration time may lead to performance degradation of the system [344].

### 4.3 Load balancing

Load balancing is one of the most popular areas of research among fog /cloud researchers. Although lots of work have already been done in cloud computing regarding load balancing, these developed techniques cannot be directly applied in the fog paradigm due to the different capabilities of fog nodes [135]. Another main challenge of implementing load balancing in a fog environment is dealing with heterogeneity [22]. Load balancing makes sure that no node is overloaded or under-loaded. This helps in improving overall performance, by reducing the response time and increasing throughput. Broadly load balancing techniques are categories into three modes that are static, dynamic, and adaptive as shown in Fig. 6.

Static load balancers do not differentiate between available fog nodes based on their capabilities (storage and processing) [133]. They equally distribute the load between all the available fog nodes, irrespective of their capabilities. A static load balancer further can use two different techniques to migrate the load in-between the fog nodes. These techniques are "deterministic" and "probabilistic." The deterministic approach uses the already available information about the fog nodes to take the decision. The probabilistic approach is better than the deterministic approach in terms of decision making. The probabilistic approach makes the decision based on better response time probability [345]. As you know, fog devices are highly heterogeneous, so static load balancing is not meant for fog computing. There is very little research work done related to static load balancing in fog computing. Round-robin is one example of a static load balancing algorithm. Dynamic load balancer remains updated about the resources available at various fog nodes [22, 29, 31, 134–146]. So the load is assigned as per the capabilities of fog nodes. Even though dynamic load balancer provides better response time and throughput, it consumes lots of bandwidth to share resource information. Static load balancers do not consume bandwidth for sharing resource information in-between the fog nodes. Dynamic load balancing can be further categorized as centralized, decentralized, and semi-centralized. In a centralized technique, we have a global load balancer that keeps the update of all the fog nodes in a network. In the decentralized technique, we have a local load balancer of each cluster of nodes. All the network nodes are divided into various clusters, which have their local load balancer. Semi-centralized techniques combine the features of both centralized and decentralized approaches. This local load balancer interacts with a global load balancer to make optimal load balancing with low bandwidth usage.

Load balancing in mobile networks is a challenging area, in which identifying and removing the ideal connections is a tedious job. Some of the articles assumed data centers were connected with high-speed transmission lines which result in efficient load balancing [142].

### 4.4 Resource allocation

The fog layer helps in executing low-latency-based applications by providing resources near to the end-user. But unlike cloud computing, fog nodes are resource constraints. So optimally allocating the fog resources is one of the most difficult tasks. Due to the dynamic and heterogeneous nature of fog nodes, resource allocation in the fog paradigm become an NP-hard problem. Heuristic, approximation, and RL-based algorithms are most popularly used to resolve resource allocation problems.

Resource allocation is a systematic approach to allocate fog nodes to IoT services such that SLA does not violate as well as resources can be utilized efficiently (energy, bandwidth, price, and so on). Note that resource allocation is different from

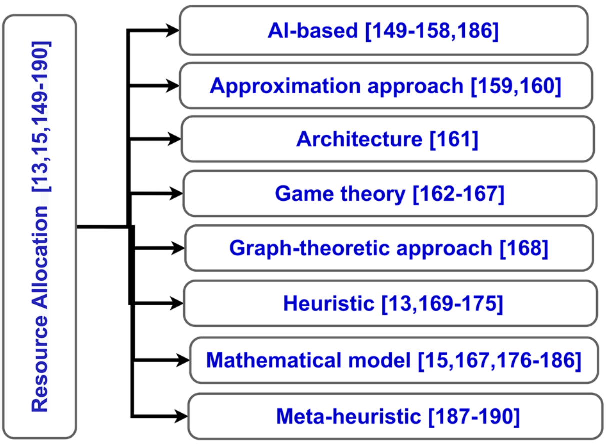


**Fig. 7** Subclassification of resource allocation approach

resource scheduling. In resource scheduling, we decide the sequence of execution of tasks while in resource allocation we assign the resources to end-users before scheduling them for a particular task. Otherwise many same types of algorithms are used in both scheduling and allocation of resources such as round-robin, first come first serve, and so on. Based upon the research papers published so far in this field, we can categorize resource allocation methods as auction, application, SLA, AI, and energy-based. The broad classification of resource management techniques is shown in Fig. 7.

Both user and resource owner want to get maximize their profit. So auction-based approach is used to assign available resources to users. In auction-based-approach user make their bid for resources, the user with the highest bid get the resource [159, 160, 162, 167, 168]. This approach gives benefits to both the service provider as well as the user. Nowadays both auction and non-auction (priority, computing power, deadline-based, so on)-based approaches are used very popularly for resource allocation.

Application-based research work can be further divided into three categories, i.e., data-intensive [13], real-time [161, 163, 170], and mobile [164, 171]. Data-intensive applications need to process lots of data, which is not usually cannot be processed at the edge level. So resource allocators must transfer these types of applications to the cloud for processing. Real-time applications need to be processed at the edge only. Resource allocation for mobile applications is a challenging task. SLA (service-level agreement)-based resource allocation takes care of deadline, delay, and priority-based agreements between user and service provider. In the deadline-based approach, user provides the fixed time slot for task completion, while the delay-sensitive approach makes sure about the low response time [173, 181, 184]. In a priority-based approach, high-priority tasks are allocated resources by preempting them with low-priority tasks [165]. Allocating resources to users according to their resource usage pattern is one of the emerging ways of resource allocation. AI approaches such as reinforcement learning are used to predict the pattern of resource

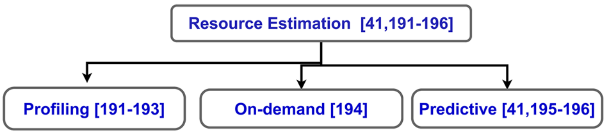


**Fig. 8** Subclassification of resource estimation approach

usage by end-user. This helps in improving QoS as well as resource utilization in an optimal way. This approach is still emerging as this is very difficult to apply AI-based heavy data-intensive approaches at resource constraint fog nodes. Energy-efficient approaches are also popular due to the increasing need for energy for data processing. Green energy-based methods are getting momentum nowadays [188].

### 4.5 Resource estimation

The resource estimation technique helps in determining how much resources are required by a particular application. Based on application characteristics crucial placement decisions are made for capacity constraint fog nodes. In this technique, we try to make a suitable match between application requirements and fog nodes. Such as applications with computing requirements are matched with high-processing-capacity fog nodes while applications that require more storage are directed to high-storage-capacity fog nodes.

Resource estimation helps in service-level agreement (SLA) management and achieving QoS. The changing requirement of end-user also makes it necessary to estimate application resource requirements, for efficient use of available resources.

Resource estimation techniques can be categories into three types, i.e., profiling, predictive, and on-demand as shown in Fig. 8.

In the profiling approach, resource requests from applications will remain static, which means if initially, an application requires computing resources that will not demand storage resources later. In this technique, the resource estimator initially will execute an application on all available resources. This will help in finding out which resource is better suitable for which application. Later an application will execute on its matching fog node only. Note that here we are assuming that the application resource requirement will not change, which will remain static. In the predictive approach, resources are allocated based on application experience. This technique is very effective in a dynamic environment, where fog nodes, as well as application characteristics, change frequently. This is a mathematical probability-based approach as compared to the profiling approach which is dependent on physical deployment. In this approach, resources can be increased and decreased as per application requirement, unlike the profiling technique in which scaling is not possible later. But profiling gives must better solutions as it checks the suitability of resources with application practically. In the on-demand approach, resources are estimated based on customer instant demand. Here AI (neural network, reinforcement learning)-based algorithms are more popularly used for resource estimation as compared to mathematical models used in the predictive approach [194]. In

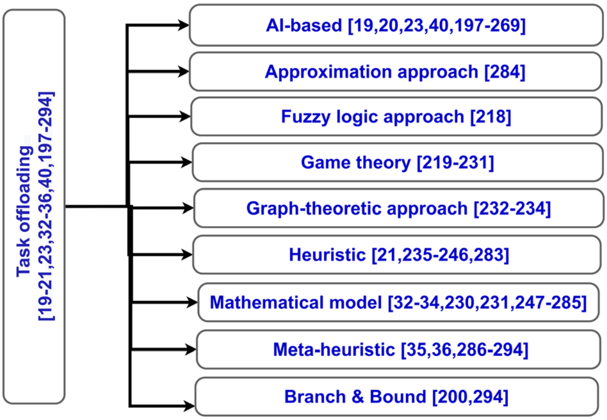


**Fig. 9** Subclassification of task offloading approach

this technique, application characteristics are not used for estimating the resource requirement but past on-demand statistics help in making the decision.

### 4.6 Task offloading

IoT devices usually have low battery, limited computation, and storage capacity, especially to support real-time applications such as online gaming, virtual reality, and augmented reality-based applications. 5G is another growing technology that is based on edge-based computing. So efficiently offloading end-user tasks to the appropriate fog node is a must, to realize the working of these applications.

In this survey, we have categories the offloading technique based on different optimization technologies used to solve this problem. These categories are (non-) convex optimization, game theory, Lyapunov optimization, Markov decision process, heuristic approach, meta-heuristic approach, and AI-based as shown in Fig. 9

In resource offloading, end-user offloads their resource-intensive task to nearby edge devices. Selecting the appropriate task and edge node for task offloading is one of the most important areas of research. Resource offloading does not only work between IoT devices to fog devices but plays an important role between fog devices and the cloud. Effective offloading of tasks can make a big difference in overall system performance. Many surveys have been done based on static- and dynamic-based offloading as well as communication mode-based offloading. In static offloading, fog nodes are already fixed for specific task offloading, while in the dynamic approach decision of offloading is taken as per the resource requirement by the end-user task in run time. Communication-based approaches can be one to one, one to many, many to one, and many to many. In a one-to-one approach, one user application can offload workload to only one fog node, while in a one-to-many approach, one task can be divided into many sub-modules which further can be assigned to different fog nodes. The one-to-many approach helps execute the applications in parallel, but more coordination is required between fog nodes. In many-to-one approach, many

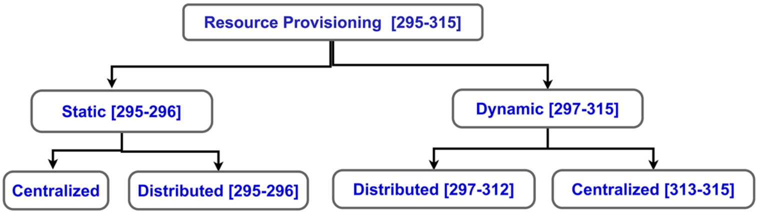


**Fig. 10** Subclassification of resource provisioning approach

tasks can be offloaded onto a single machine. Here VM and container technology play an important role to run many applications on the same hardware. The many-to-many approach supports both one-to-many and many-to-one approaches.

### 4.7 Resource provisioning

IoT devices are heterogeneous, which have a very dynamic requirement of resources. The resource provisioning approach helps in scaling in and scaling out of resources as per user requirements. Allocating the highest possible required resources may result in SLA fulfillment but will cause more financial burden on end-user, while allocating fewer resources may result in SLA violations. Resource provisioning also helps in deciding the target resource location for a particular application.

We have categorized the resource provisioning techniques as static and dynamic, which are further classified as centralized and distributed as shown in Fig. 10. In static resource provisioning, resources are allocated based on the current status of the system. The configuration of the resource scaling system remains fixed and does not change at run time. The threshold-based approach is very commonly used in static resource provisioning, in which a threshold value is set for allocating resources to a particular type of user. Setting up the threshold value always remains a challenging issue which may result in under-provisioning or over-provisioning of resources. In dynamic resource provisioning, prediction or learning algorithms are used to estimate the future demand of users. The resource monitoring system is used by this approach to keep itself up to date regarding the availability of resources. The dynamic approach adjusts itself automatically as per the user requirement at run time. There may be some delay in provisioning the new resource, but 100% resource utilization remains the goal of this system.

### 4.8 Resource discovery

Resource discovery can be performed in many ways such as with the help of service providers or dedicated node discovery leaders. Finding the available service providers can be costly (energy consumption). Many of the edge nodes operate on

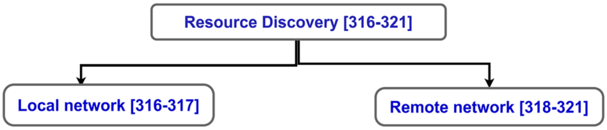


**Fig. 11** Subclassification of resource discovery approach

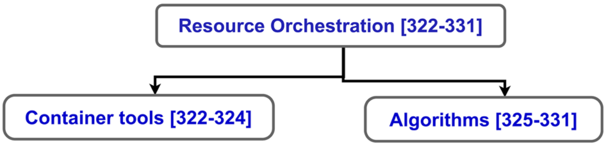


**Fig. 12** Subclassification of resource orchestration approach

batteries. These nodes need to utilize their energy efficiently in search of service providers.

In case the number of required fog nodes is more than the number of available nodes, new nodes can be discovered by the discovery leader. New nodes are discovered in an unregistered list of nodes. The unregistered node list consists of those nodes which do not update the leader about its resource status in the specified time interval. These nodes are considered dead and discovered again only in the case registered or live nodes do not satisfy the user resource requirement.

One another way of resource discovery is finding the best available surrogate. Broker nodes help in finding the best suitable surrogate node based on the requirement.

The available free resources can be searched in local networks or remote networks. Discovering local resources can be energy-efficient, while remote resource discovery improves scalability. We have categorized the resource discovery articles based on local and remote network coverage as shown in Fig. 11. Most of the literature assumes that edge resources are already discovered, although resource discovery is not an easy task.

### 4.9 Resource orchestration

As the number of Internet of Things (IoT) applications is growing at a very fast pace, this is not possible to handle huge data generated by IoT devices manually. Orchestration helps in automated coordination and management of middleware and services. Applications at the edge have a very dynamic nature because their resource requirement changes frequently. Resource orchestration maps service requests to resources. In resource orchestration, execution locations for end-user applications are mapped within a fog environment. These resources can be physical or virtual in the form of containers. Resource orchestration also helps in resource scaling, making advanced networking between fog nodes, sharing storage between different application hosts, as well as configuration management. Orchestrator keeps a close watch on the load and performance of different applications. Orchestrator also helps in handling failures. Typically multiple orchestrators remain in the network, so if one of them fails then others are still allowed to access the applications. Lots of work have already been done on the cloud platform for automatic resource orchestration. However, traditional cloud orchestration approaches such as OpenStack [346] cannot be applied directly to such a large-scale, heterogeneous, and distributed fog environment. We have categorized the resource orchestration articles based on container tools and algorithm used as shown in Fig. 12.

After the success of containers in the cloud paradigm, container-based applications are gaining pace in edge computing.

There are many container-based orchestration technologies in the market in which Docker Swarm and Kubernetes are the most popular. Mesos is another popular orchestrator. After the success of Google's Kubernetes in the cloud platform [347], many researchers also check the feasibility of Kubernetes-based orchestration in the edge environment.

Many orchestration frameworks have been designed to improve resource management at the edge network. Some orchestration framework also supports multi-domain resource management. Cross-domain resource orchestration helps in implementing a variety of user services. Although many fog orchestration architectures have already been proposed, these are not practically implemented in a real fog environment.

## 5 Future research prospects

Due to the heterogeneous and dynamic nature of fog nodes, this is not easy to allocate and share the resources in the fog paradigm as compared to the cloud. Resource management in the fog environment is more challenging than cloud computing [348].

In this section, we have discussed the research gap based on QoS factors, technique/algorithm, tools, applications, mobility support, heterogeneity, AI-based, distributed network, hierarchical network, and security. The reviewed articles in the range [2, 12–15, 18–23, 27, 29–36, 39–41, 44, 58–331] are taken into consideration. Note, many of the articles used multiple techniques/algorithms to work on their problems. Similarly, almost every reviewed article worked to improve multiple QoS factors. Articles with multiple QoS factors and techniques are considered multiple times. Note that, some of the articles used more than one tool while others do not use any tool at all.

### 5.1 QoS-based resource management

Fog computing helps in reducing the latency by processing the data at the edge of the network. The largest chunk of reviewed articles focused on improving latency (approx. 25%), which shows the advantage of fog/edge computing over cloud computing. Fog/edge architecture provides the data processing services near to the end devices which reduce the latency as compared to the cloud paradigm. Some of the researchers improved latency by reserving nearby resources for upcoming user demands [191, 193].

Usually, fog nodes that are close to end devices have a direct power supply, but fog nodes at higher layers have limited battery backup, which makes it essential to use energy efficiently. The use of optimal techniques and algorithms can help in reducing energy consumption by making fog nodes turn off and turn on whenever required. However, energy consumption remained one of the important areas of improvement still very few reviewed articles (approx. 2%) focused on renewable energy [84, 120, 188, 213, 264, 265].

Some of the articles focused on the energy-delay trade-off [71, 262, 292]. Some other existing articles stressed resource scheduling for the heterogeneous network [44, 108] while energy-aware real-time schedulers can be explored for heterogeneous fog environment [37, 349]. Security-based energy-aware scheduling of tasks for heterogeneous fog networks can be explored further [350].

Fog/edge computing architecture is dynamic and heterogenous, which leads to an increase in the computational complexity of proposed algorithms. Many articles focus on improving computational complexity (approx. 32%). Although AI-based approaches support distributed processing, learning, and storage capabilities, they required computationally intensive machines. The lightweight distributed AI-based models can be explored extensively.

Resources at the edge are limited as compared to the cloud. Using these resources efficiently for the growing number of IoT devices is one of the difficult tasks. The fourth-largest chunk of reviewed articles (approx. 30%) worked on efficient resource utilization. One of the main reasons for supporting efficient resource utilization was the use of dynamic techniques by a large number of articles. Some of the existing articles implemented dynamic accurate resource estimation in edge computing [41]. Resource estimation can help in minimizing resource underutilization by knowing the need of end-user in advance. Accurate and detailed estimation of required resources by end-user applications still needs to be explored more. Further, allocating the right amount of resources to variable workload-based mobile applications is another big challenge.

Due to the limited number of resources at the edge, renting edge devices is more costly as compared to the cloud. Monetary cost at edge remains different from cloud-based pricing models due to localized demand of resources and distributed fog nodes. Unlike the cloud market, there are very few service providers in the fog paradigm. So providing resources in a cost-efficient manner is a big challenge at the fog layer. Further, most of the existing works offloaded the task to the cloud in case the edge node was not available. This could be improved by making a consortium of fog service providers and directly communicating with them in case of edge resource scarcity. This could improve the performance as well as result in monetary benefits.

Sharing of sub-tasks among various applications can reduce the computational cost of fog nodes. In this, shared caching techniques can play important role in fog computing [42, 351]. The shared caching technique helps in the reusability of data among local users which can result in cost reduction and performance

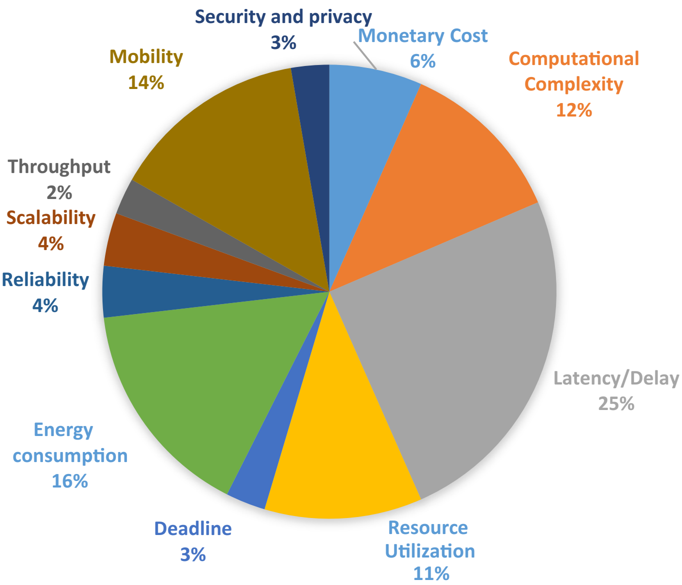


**Fig. 13** Number of articles based on QoS factor support

improvement. Although some of the reviewed articles used caching for resource management in fog computing [59, 127, 151, 267], there is a lot of scope to explore this area.

The problem of limited resources at the fog layer can be solved by adding more fog nodes. However this further increases the problem of energy consumption and deployment cost [352]. An optimal trade-off between resource scaling and QoS factors such as deployment cost and energy consumption needs to be balanced. Scaling resources at the edge increases monetary cost but improves latency. The right trade-off between cost and latency is one of the important areas of research. Around 6% of reviewed articles, minimized the monetary cost, in which very few discussed trade-offs between monetary cost and delay [101].

Fault tolerance/reliability is the ability of a system to continue its service despite some fault that occurs in the system. Estimating the resources needed before its actual request is made by the user application, improves the reliability of the system. [106] implemented the fault tolerance-based resource scheduling model which could handle unstable communication conditions between edge and end devices. Some of the articles also focused on priority-based resource scheduling [83, 109]. [109] proposed priority-based resource scheduling approach for critical heartbeat monitoring system, still do not consider any fault tolerance and security measures. Further, security threats can lead to various faults in the system. Since fault tolerance/reliability is weakly covered in the literature, it is presented as an open issue in this subsection.

In this review work, we find out that most of the existing articles worked upon a few QoS factors. Therefore, resource management techniques must focus on the

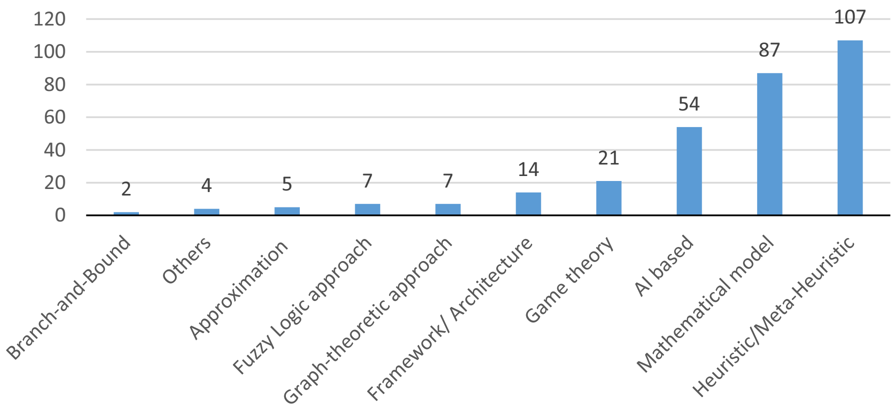


**Fig. 14** Number of articles based on the technique used

improvement of as many QoS factors as possible. Comparative analysis of various techniques is shown in Fig. 13.

### 5.2 Technique-based resource management

The fog paradigm is highly heterogeneous and dynamic. Heuristic and meta-heuristic techniques were most widely used for dynamic resource management in reviewed articles. Many of these articles worked upon NP-complete problems. Heuristic/meta-heuristic techniques helped in solving hard problems in a reasonable time. Although heuristic algorithms are efficient and quickly obtain optimal results, they can easily fall into local optimum. Some reviewed articles used hybrid and hyper-heuristic techniques which still have a lot of scopes to explore further.

The second most popular technique was mathematical optimization with 27% articles, in which convex optimization has approx. 8% and Lyapunov optimization have approx. 6% share. The Lyapunov optimization is a centralized approach that may cause the problem of high delay in large-scale edge networks, especially for real-time applications. In the reviewed articles, model-based techniques [71, 118, 222, 236] mostly used mathematics-based techniques for identifying optimal fitness value for a fitness function. Mathematics-based techniques are mature and provide near-optimal results. However, mathematical models have high complexity which results in poor practicality in the dynamic fog resource constraint environment.

AI-based techniques were the third most used technique to solve resource management problems in fog/edge computing. AI-based techniques are very efficient in handling dynamic resource scheduling. Around 18% of reviewed articles used AI-based technique, in which DRL was most popularly used with approx. 7% share. DRL remains one of the most popular techniques in the last three years (2019–21) for handling dynamic task offloading problems. The DRL-based technique is getting popularity due to its ability to handle dynamic complex problems. Surprisingly very

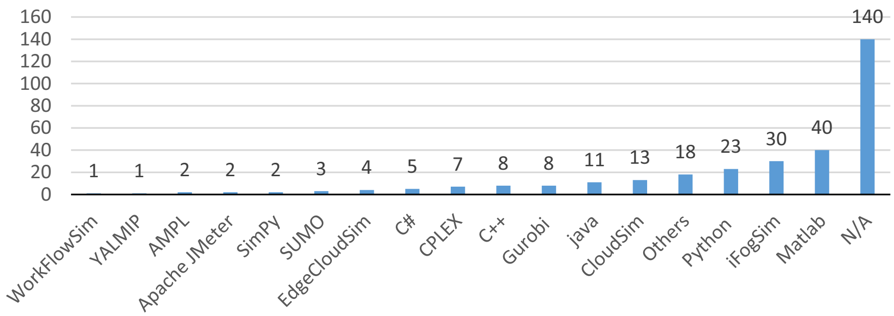


**Fig. 15** Number of articles based on the tool used

few of the AI-based reviewed articles worked on high computational complexity. AI-based resource management with minimized computation complexity has great scope for future research. Further AI-based algorithms can be used for measuring security threats, context awareness, and future resource requirements.

The next most explored technique in reviewed articles was the game theory with approx. 5% articles share. The game theory approach is suitable for task offloading in distributed mobile edge computing environments rather than centralized methods [219]. Game theory is simple and easy to implement which helps in the implementation of practical edge computing-based applications. Although game theory is a flexible technique, continuous iterations may be required to achieve Nash equilibrium.

The least explored techniques include the fuzzy logic approach, approximation, graph-theoretic approach, branch-and-bound, and others (containerization + protocol). Although approximation-based algorithms are simple and easy to implement, they can easily fall into local optimum. The fuzzy logic approach can help in partitioning the end-user task to place them into various fog nodes [114]. Further, these partitioned tasks can be prioritized to place on resource constraint fog nodes. Containerization is getting popularity at edge layer. However, unlike the cloud, the standard containerization tools are not available at the edge. Comparative analysis of various techniques is shown in Fig. 14.

### 5.3 Tools used in resource management

MATLAB was the most widely used tool for simulation in reviewed articles. MATLAB was successfully used to implement various types of techniques such as fuzzy logic [67, 301], AI-based [18, 40, 134, 151, 217], mathematical models [33, 73, 74, 126, 127, 143, 177, 185, 185, 186, 195, 258], heuristic [21, 121, 175, 241, 244, 246, 315], meta-heuristic [31, 67, 68, 85, 188–190, 301], game theory [148, 163], and approximation [71]. The second most popular tool used in reviewed articles, iFogSim, getting popularity due to its additional features of supporting dynamic and mobile-based networks. iFogSim remained popular among heuristic/meta-heuristic-based articles. Many of the authors preferred iFogSim for small and medium-scale experiments. The growing use of reinforcement learning to handle dynamic edge computing problems leads to the use of python. Due to the rich library, python has

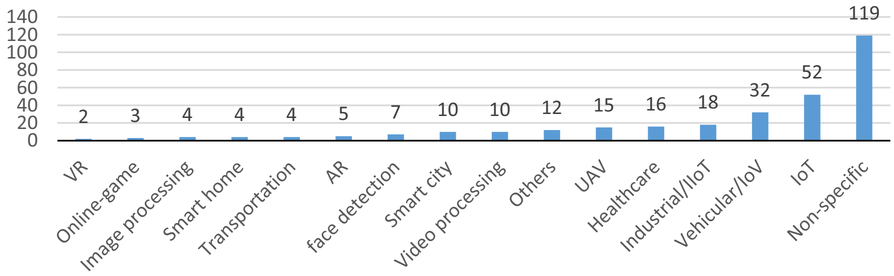


**Fig. 16** Number of articles based on the application area FFI

become one of the most popular tools used for AI-based experimentation. Python support various advanced libraries such as TensorFlow, NumPy, and Keras, which attracted the researchers. CloudSim [25] remained more popular among heuristic-based techniques [60, 149, 151, 160, 183, 188, 326]. Although lots of tools are available to perform experiments for resource management approaches, no one exactly provides the real fog computing environment for experimentation. This leads to an open challenge to provide an effective tool for fog paradigm-based experiments. Further, there is no RL-based standard environment is available to perform experiments related to edge computing. In the future, a standard RL-based environment can be designed to make fog computing-based experiments. These standard RL-based environments will also help in performing and comparing experiment results on standard common ground.

Most of the fog computing study is based on simulation tools such as iFogSim [353], EdgeCloudSim [354], and many more. However, many of the experimental test beds were set up by many authors. These test beds were specific to a particular problem. The generalization of large-scale test beds can be explored. Different tools used in the literature are shown in Fig. 15.

### 5.4 Application-oriented resource management

IoT devices generate huge amounts of data, which need to be processed in real-time. Fog computing is playing the most important role in the success of IoT-based applications. There are many IoT-based applications that require different QoS, SLA, and different working environments. IoT-based applications are the highest explored area in reviewed articles. Vehicular/IoV-based applications require real-time fast computation, which is not possible without edge computing. Autonomous driving requires below 50 ms response time that cannot be achieved with cloud computing alone [355]. Smart Industrial applications required real-time actions, which cannot be possible with cloud usage. Edge computing is playing important role in the implementation of IIoT. Healthcare 4.0, growing at a very fast pace, in which real-time monitoring of the patient is done with the help of edge technologies. Some of the reviewed articles performed UAV-based case studies. UAV playing important role in disaster management, product delivery, and defense applications. Video-based applications required a large amount of data processing. Some of the existing articles performed case studies related to video surveillance, video streaming, video games, and video analytics. Sending video data to the cloud server for processing can chock the backbone network. Edge computing helps in processing high-volume data near its source itself. Although many of the reviewed articles worked upon real-time applications [12, 70, 122, 170, 246, 248, 257, 274, 296, 304], this is challenging to reduce the latency of real-time applications [119]. Further, application-specific resource management approaches can be explored in near future. Different application areas used for the case study in the literature are shown in Fig. 16.

### 5.5 Mobility-based resource management

Mobility support in fog-based networks is very challenging due to its heterogeneous nature. Almost every node has different protocols and capacities. Coordination between such diverse network devices is not easy. Lots of work is needed to set up new communication protocols. The trade-off between migration cost and low latency remained one popular area of research in reviewed articles. However, the migration of user task which is offloaded to multiple servers is itself a challenge in MEC. It is important to guarantee service continuity with QoS requirements. New fog environment specific migration techniques can be worked upon. Another possible improvement in mobility support can be performed by making SLA-based collaboration between fog service providers. For example, if a mobile device moves from one region to another then other fog service providers may provide service to that device in case of its own service provider's un-availability. The number of edge nodes may not be sufficient alone for congestion-prone areas in MEC [356]. Further, this limitation of MEC can be improved by using mobile crowdsensing. Reinforcement learning may also play important role in the predictions of mobility patterns of end devices and fog nodes.

### 5.6 AI-based distributed resource management

The resources at the edge are limited as compared to the cloud. In Fog computing, usually, lightweight data processing devices such as wireless access points, routers, switches, and CCTV cameras are used as fog nodes. These resource-limited devices are not suitable for deploying AI-based techniques on them. AI-based techniques are one of the most popular techniques to solve resource management problems in fog/edge computing. The imbalance between resource-limited devices and AI-based approaches needs to be addressed. The distribution of deep neural networks across edge layers can resolve this problem. Multiple layers of deep learning can be operated on separate fog nodes [357] which leads to improvement in QoS. This research further extended for using geographical distribution edge nodes for executing different layers of a neural network [358]. However, the development of a distribution-based algorithm is a challenging task due to real-time process communication management. Some of the researchers worked on it. [203] implemented distributed task offloading model using the DRL approach. They worked on a real-time case study of intelligent connected vehicles. The authors improved the task delay and computation time by using the vehicles as a resource pool. [221] implemented distributed multi-user task offloading method using ML approach. They reduce the computation and communication offloading cost on multiple cloudlets. [136] implemented load distribution mechanism in real-time autonomous driving vehicles. They improved the scalability on the basis of the number of vehicles by reducing the processing delay. Further, they remained successfully to distribute a large amount of data among edge servers. [155] implemented multi-UAV-assisted resource allocation model for distributed IoT network. They proposed a DRL scheme for improving efficiency in an emergency scenario. [123] implemented federated learning-based content placement approach for distributed fog network. Federated learning reduces learning overhead by distributing the learning process on multiple fog nodes. However, federated learning at edge computing for deploying heavyweight AI applications needs to explore [359].

### 5.7 Distributed edge resource management

The edge paradigm is highly distributed which makes it a necessity to keep exact information about the architecture of edge nodes and their location. Many of the existing articles used the distributed approach for edge resource management. [136] implemented a load balancing technique for the dynamic map system of autonomous vehicles by linking distributed multiple edge servers. They divided the travel lanes for autonomous driving vehicles with respect to the edge servers. Each edge server was assigned one lane to process data. The data for processing are distributed among edge servers by the lane section ID feature. [22] proposed lightweight distributed load balancing algorithm for variable workload and high-delay-based heterogeneous networks. They conducted the testing for the proposed algorithm on geographically distributed smart city infrastructure. The results show the improvement in response time and reduce the loss rate. [155] implemented multi-UAV-assisted resource allocation and task offloading-based DRL framework for distributed IoT networks. They worked upon minimization of energy consumption, delay, and efficient utilization of computational resources. [203] proposed a DQN-based distributed task offloading approach that divides the complex task into multiple smaller tasks and efficiently offloads sub-task to achieve low execution time. The proposed model ensures the delay requirement of big tasks in the resource constraint fog environment. They used the vehicles as edge nodes and distribute the sub-task among vehicles. The scheme was based on vehicle-to-vehicle communication and relative distance between them. [221] proposed fully distributed multiuser task offloading algorithm to reduce communication and computation cost. The proposed model tested on the cloudlet-based network which results in maximizing the number of beneficiary mobile devices. However, they assumed that all the mobile devices are homogeneous in nature. [231] designed a distributed task offloading optimization framework for the MEC network. They compare the state-of-the-art distributed algorithms and other competitive methods. The proposed distributed method well approximates the global optimum. Although many reviewed articles worked on distributed edge computing, they do not evaluate the decision-making time of distribution of tasks among fog resources [19].

Many of the existing resource orchestration articles worked upon distributed edge computing [306, 320, 321, 331]. However, the proposed resource orchestration architectures used a theoretical approach to deal with distributed fog computing problems. New approaches can be proposed to manage distributed edge nodes efficiently.

### 5.8 Heterogeneous edge resource management

Unlike cloud resources, fog nodes are highly heterogeneous—different edge devices have different processor architectures [7]. The different processor architecture of heterogeneous edge resources can pose many challenges in task deployment [360]. Although some of the reviewed articles worked on heterogeneous edge computing network [115, 247, 258, 311], the use of heterogeneous edge resources for building an ideal edge computing-based model is infancy [361]. [69] proposed homogeneous fog network-based task scheduling model. Although their model provides maximal energy efficiency, it could not be applied to a realistic heterogeneous fog environment. [362] implemented homogeneous fog resource-based handshaking resource discovery protocol on the EaaS platform. EaaS platform used a centralized master node to provide edge nodes availability information which further limits it. Although some of the researchers worked upon heterogeneous edge networks, a lot of scopes are there to improve. [22] implemented load balancing mechanism for heterogeneous fog architecture. The authors consider the heterogeneous fog nodes in the context of processing power, numbers, and configuration. The experimental result shows a reduction in loss rate and improved latency. However, they do not focus on heterogeneity in terms of end devices and end-user tasks. [189] designed an algorithm for resource allocation and optimizing mining decisions in the mobility-supported blockchain network. They helped in providing optimized mining decisions. However, the IoT devices used in experiments were homogeneous in nature. Load balancing between heterogeneous fog nodes consumes lots of bandwidth which needs to be taken care of. Further, heterogeneity in mobile devices can be another big challenge [363].

### 5.9 Hierarchical edge-cloud-based resource management

Usually, tasks needing real-time processing are executed on edge nodes while tasks needing big data analytics are sent over to the cloud [364]. Hierarchical load distribution between edge and cloud resources makes fog computing more successful. Many of the reviewed articles used hierarchical edge-cloud architecture for resource management. [65] implemented a deadline-based task scheduling approach for the edge-cloud paradigm. They successfully minimize the execution time, although their solution could not be applied to dynamic and real-time-based problems. [77] proposed space–air–ground architecture-based resource scheduling optimization model for IoV. They used SDN and NFV techniques for communication. Their model shows the improvement in service delay, energy consumption, resource utilization, and security. [78] proposed the two scheduler approach which schedules the task on edge or cloud resources. The schedulers consumed low CPU, RAM, and storage space while ensuring a reduced makespan of fog applications. [82] proposed mobility-aware resource scheduling approach for distribution of healthcare tasks among fog and cloud resources. The proposed model schedules the critical task on edge nodes while low-priority high-computational task on cloud. The simulation results show low makespan and energy consumption. [117] implemented a service placement approach for the hierarchical edge-cloud network. They split a large task into smaller ones and distribute the smaller task between edge and cloud. However, this is not easy to divide a big task into small tasks such that small parts can execute simultaneously on cloud as well as edge. An appropriate application placement technique can be explored with an efficient task partitioning method for an edge-cloud network.

Further, edge intelligence is another new emerging area that can be explored, in which the front layers of the deep neural network can be offloaded to the edge server and the rest of the layers can be deployed on the cloud [365]. Although this will make it easy to deploy heavy AI-based models on the edge with the collaboration of the cloud, finding out which neural layer output must go to the cloud is itself a challenge.

### 5.10 Security and privacy-based resource management

The limitation of resources at the edge and wireless communication increases the security and privacy issue at the edge [366]. Due to the limitation of resources at the edge, complex security solutions cannot be deployed on the edge devices. End devices have even more limited storage, energy backup, and processing power. Providing security solutions for lightweight end devices is another big challenge. However, the deployment of firewall and instruction detection at the edge can be beneficial. But customization of machine learning and deep learning security-based algorithms for resource constraint edge nodes is challenging. Training the AI models with limited data and resources remains a research challenge. Many existing articles use the compute-intensive algorithm for security implementation at the fog layer but ultimately that is impacting the performance. A balanced approach between security and other functionalities can be explored in the future. Many of the reviewed articles left the security and privacy issue for future research [23, 102, 109, 125, 138, 144, 145, 198, 201, 226, 290].

Many of the reviewed articles used blockchain technology for security and privacy issues. [199] proposed blockchain-based approach which secures transaction data storage by restricting any delete or update on the blockchain. A computational difficult puzzle is set to verify the valid block. [216] proposed blockchain integrated scheme in which edge access points serve as blockchain nodes. Blockchain nodes process the task offloading information to enhance the security of the system. [220] implemented uniform pricing scheme using blockchain on lightweight network devices. However, they assume that the link between the miners and cloud/fog computing units is sufficiently reliable and secured. [249] proposed a model which provides satisfactory security by dividing network fragments between selected fog nodes. This reduces the damage caused by the leak of sensitive data due to malicious fog nodes. As the task gets divided between various fog nodes, one service provider could not access the complete information. However, the proposed model increased the delay, due to which real-time tasks could not be served. [272] proposed blockchain-based secure task offloading architecture. They considered both task migration and network-based security features for heterogeneous fog nodes. However, they do not consider the fault-tolerant aspect for vehicular fog cloud networks. Blockchain technology can help in improving security and privacy issue for distributed resource management in fog computing. The use of blockchain technology is still in a very early stage in fog resource management.

## 6 Conclusion

Fog computing is a powerful emerging technology that is helpful in the realization of various new world applications such as 5G, online gaming, autonomous vehicles, smart grid, smart traffic control, IIoT, healthcare 4.0, IoV, and many more. These applications are based on the efficient management of edge resources. One of the main concerns in fog/edge computing is limited resources at the edge network. Achieving QoS with SLA satisfaction in such a heterogeneous, distributed, and dynamic environment is another big challenge. This paper provides a systematic review of the last 6 years (2016–2021) literature, by a comparative analysis of articles based on QoS factors, techniques, tools, and application areas. The rigorous analysis of different resource management approaches helps us to find out their research gaps. The future research prospects were discussed based upon QoS factors, technique/algorithm, tools, applications, mobility support, heterogeneity, AI-based, distributed network, hierarchical network, and security. Some of the key findings are mentioned below:

- Heuristic and meta-heuristic techniques were most popularly used in reviewed articles.
- DRL has a lot of potentials to handle heterogeneous and dynamic edge computing problems.
- Lack of standardization of technologies such as containerization tools and RL-based environment in the fog paradigm.
- The use of renewable energy resources is a big area that needs more attention.
- Scalability, reliability, deadline, and throughput were least focused in the literature.
- The trade-off between migration cost and low latency can be another area of research for mobility-based resource management.
- IoT, IIoT, healthcare, and IoV remained the largest application area explored in reviewed articles.

- The use of heterogeneous edge resources for building an ideal edge computing-based model is in infancy.
- Many reviewed articles worked on distributed edge computing, although they do not evaluate the decision-making time of distribution of tasks among fog resources.
- An appropriate application placement technique can be explored with an efficient task partitioning method for a hierarchical edge-cloud network.
- Deployment of heavy security solutions on resource constraint edge devices is another big challenge.
- Federated learning at edge computing for deploying heavyweight AI applications needs to explore more.

Fog technology's real-world implementation is at a very early stage. This is not possible to change all network hardware to comply with fog technology. Unlike cloud computing in which lots of big service providers are available, fog computing is also lacking the investment. However, fog/edge computing might be an opportunity for developing countries, where directly installing new fog-supported hardware is much cheaper than replacing it with old establish networks as in developed nations.

## References

1. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing (MCC '12). Association for Computing Machinery, New York, NY, USA, pp. 13–16. https://doi.org/10.1145/2342509.2342513
2. Samanta A, Tang J (2020) Dyme: dynamic microservice scheduling in edge computing enabled IoT. IEEE Internet Things J 7(7):6164–6174. https://doi.org/10.1109/JIOT.2020.2981958
3. Pengfei H, Dhelim S, Ning H, Qiu T (2017) Survey on fog computing: architecture, key technologies, applications and open issues. J Netw Comput Appl 98:27–42. https://doi.org/10.1016/j.jnca.2017.09.002
4. Kansal P, Kumar D, Kumar M (2020) Introduction to fog data analytics for IoT applications. A book chapter publish in "Springer Singapore" with ISBN 978–981–15–6044–6.
5. Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: a complete survey. J Syst Archit 98:289–330. https://doi.org/10.1016/j.sysarc.2019.02.009
6. Atlam HF, Walters RJ, Wills GB (2018) "Fog computing and the internet of things: a review" *Big Data Cogn*. Comput 2(2):10. https://doi.org/10.3390/bdcc2020010
7. Varghese B, Wang N, Nikolopoulos DS, Buyya R. 2017b. Feasibility of fog computing. (2017). arXiv:1701.05451arXiv:1701.05451
8. Yu W et al (2018) A survey on the edge computing for the internet of things. IEEE Access 6:6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504
9. Ouyang T, Zhou Z, Chen X (2018) Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing. IEEE J Sel Areas Commun 36(10):2333–2345. https://doi.org/10.1109/JSAC.2018.2869954
10. Haghi Kashani M, Rahmani AM, Jafari Navimipour N (2020) Quality of service-aware approaches in fog computing. Int J Commun Syst 33:e4340. https://doi.org/10.1002/dac.4340
11. Hong C-H, Lee K, Kang M, Yoo C (2018) qCon: QoS-aware network resource management for fog computing. Sensors 18:3444. https://doi.org/10.3390/s18103444

12. Kassir S, Veciana GD, Wang N, Wang X, Palacharla P (2020) Service placement for real-time applications: rate-adaptation and load-balancing at the network edge. In: 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), New York, NY, USA, 2020, pp. 207–215. https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00044
13. Sood SK, Singh KD (2019) SNA based resource optimization in optical network using fog and cloud computing. Opt Switch Netw 33:114–121. https://doi.org/10.1016/j.osn.2017.12.007
14. Zhao X, Shi Y, Chen S (2020) MAESP: mobility aware edge service placement in mobile edge networks. Comput Netw 182:107435. https://doi.org/10.1016/j.comnet.2020.107435
15. Li S, Wang Q, Wang Y, Xie J, Li C, Tan D, Kou W, Li W (2021) Joint congestion control and resource allocation for delay-aware tasks in mobile edge computing. Wireless Commun Mobile Comput 2021:1–16. https://doi.org/10.1155/2021/8897814
16. Tan Z, Yu FR, Li X, Ji H, Leung VC (2017) Virtual resource allocation for heterogeneous services in full duplex-enabled scns with mobile edge computing and caching. IEEE Trans Veh Technol 67(2):1794–1808
17. Wang P, Yao C, Zheng Z, Sun G, Song L (2018) Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems. IEEE Internet Things J 6(2):2872–2884
18. Shadroo S, Rahmani AM, Rezaee A (2020) The two-phase scheduling based on deep learning in the Internet of Things. Comput Netw 185:107684. https://doi.org/10.1016/j.comnet.2020.107684
19. Wang Y, Wang K, Huang H, Miyazaki T, Guo S (2019) Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications. IEEE Trans Industr Inf 15:976–986. https://doi.org/10.1109/TII.2018.2883991
20. Elgendy IA, Zhang WZ, He H et al (2021) Joint computation offloading and task caching for multi-user and multi-task MEC systems: reinforcement learning-based algorithms. Wireless Netw 27:2023–2038. https://doi.org/10.1007/s11276-021-02554-w
21. Zhu Q, Si B, Yang F, Ma Y (2017) Task offloading decision in fog computing system. Chin Commun 14(11):59–68. https://doi.org/10.1109/CC.2017.8233651
22. Beraldi R, Canali C, Lancellotti R, Mattia GP (2020) Distributed load balancing for heterogeneous fog computing infrastructures in smart cities. Perv Mobile Comput 67:101221. https://doi.org/10.1016/j.pmcj.2020.101221
23. Rafiq A, Ping W, Min W, Muthanna MSA (2021) Fog assisted 6TiSCH Tri-layer network architecture for adaptive scheduling and energy-efficient offloading using rank-based Q-learning in smart industries. IEEE Sensors J. https://doi.org/10.1109/JSEN.2021.3058976
24. Puliafito C, Vallati C, Mingozzi E, Merlino G, Longo F, Puliafito A (2019) Container migration in the fog: a performance evaluation. Sensors 19:1488. https://doi.org/10.3390/s19071488
25. Calheiros RN, Ranjan R, Beloglazov A, De Rose C'AF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw Practic Exp 41(1):23–50
26. Zhang J, Chen B, Zhao Y, Cheng X, Hu F (2018) Data security and privacy-preserving in edge computing paradigm: survey and open issues. IEEE Access 6:18209–18237
27. Zhai Y, Bao T, Zhu L, Shen M, Du X, Guizani M (2020) Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling. IEEE Wireless Commun 27(1):84–91. https://doi.org/10.1109/MWC.001.1900298
28. Babar M, Sohail KM (2021) ScalEdge: a framework for scalable edge computing in Internet of things–based smart systems. Int J Distrib Sens Netw. https://doi.org/10.1177/15501477211035332
29. Dong Y, Xu G, Zhang M, Meng X (2021) A high-efficient joint 'cloud-edge' aware strategy for task deployment and load balancing. IEEE Access 9:12791–12802. https://doi.org/10.1109/ACCESS.2021.3051672
30. Brogi A, Forti S (2017) QoS-Aware deployment of IoT applications through the Fog. IEEE Internet Things J 4(5):1185–1192. https://doi.org/10.1109/JIOT.2017.2701408
31. Yang J (2020) Low-latency cloud-fog network architecture and its load balancing strategy for medical big data. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-02245-y
32. Jiang C, Cao T, Guan J (2021) Intelligent task offloading and collaborative computation over D2D communication. Chin Commun 18(3):251–263. https://doi.org/10.23919/JCC.2021.03.020
33. Sun C, Ni W, Wang X (2021) Joint computation offloading and trajectory planning for UAV-assisted edge computing. IEEE Trans Wireless Commun. https://doi.org/10.1109/TWC.2021.3067163

34. Yu Z, Xu G, Li Y, Liu P, Li L (2021) Joint offloading and energy harvesting design in multiple time blocks for FDMA based wireless powered MEC. Future Internet 13:70. https://doi.org/10.3390/fi13030070
35. Guo F, Zhang H, Ji H, Li X, Leung VCM (2018) An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing. IEEE/ACM Trans Netw 26:2651–2664. https://doi.org/10.1109/TNET.2018.2873002
36. Fang J, Shi J, Lu S, Zhang M, Ye Z (2021) An efficient computation offloading strategy with mobile edge computing for IoT. Micromachines 12:204. https://doi.org/10.3390/mi12020204
37. Moulik S, Devaraj R, Sarkar A, "HEART: A heterogeneous energy-aware real-time scheduler, In: 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), 2019, pp. 476–481. https://doi.org/10.1109/VLSID.2019.00100
38. Baccarelli E, Naranjo PGV, Scarpiniti M, Shojafar M, Abawajy JH (2017) Fog of everything: Energy-efficient networked computing architectures, research challenges, and a case study. IEEE Acc 5:9882–9910. https://doi.org/10.1109/ACCESS.2017.2702013
39. Mahmoud MME, Rodrigues JJPC, Saleem K, Al-Muhtadi J, Kumar N, Korotaev V (2018) Towards energy-aware fog-enabled cloud of things for healthcare. Comput Electric Eng 67:58–69. https://doi.org/10.1016/j.compeleceng.2018.02.047
40. Wang K, Wang X, Liu X (2021) A high reliable computing offloading strategy using deep reinforcement learning for IoVs in edge computing. J Grid Comput 19:15. https://doi.org/10.1007/s10723-021-09542-6
41. Anzanpour A, Rashid H, Rahmani AM, Jantsch A, Dutt N, Liljeberg P (2019) Energy-efficient and Reliable wearable internet-of-things through fog-assisted dynamic goal management. Proc Comput Sci 151:493–500. https://doi.org/10.1016/j.procs.2019.04.067
42. Sharma S, Gupta N (2020) Federated learning based caching in fog computing for future smart cities. Internet Technol Lett. https://doi.org/10.1002/itl2.225
43. Svorobej S, Bendechache M, Griesinger F, Domaschka J (2020) Orchestration from the cloud to the edge. In: Lynn T, Mooney J, Lee B, Endo P (eds) The cloud-to-thing continuum. Palgrave studies in digital business & enabling technologies. Palgrave Macmillan, Cham. https://doi.org/10.1007/978-3-030-41110-7_4
44. Liu Z, Yang X, Yang Y, Wang K, Mao G (2019) DATS: dispersive stable task scheduling in heterogeneous fog networks. IEEE Internet Things J 6(2):3423–3436. https://doi.org/10.1109/JIOT.2018.2884720
45. Masip-Bruin X, Marin-Tordera E, Jukan A, Ren G-J (2018) Managing resources continuity from the edge to the cloud: architecture and performance. Future Gener Comput Syst 79(3):777–785. https://doi.org/10.1016/j.future.2017.09.036
46. Sha K et al (2018) On security challenges and open issues in internet of things, future gener. Comput Syst 83:326–337
47. Al-Khafajiy M, Otoum S, Baker T, Asim M, Maamar Z, Aloqaily M, Taylor M, Randles M (2021) Intelligent control and security of fog resources in healthcare systems via a cognitive fog model. ACM Trans Internet Technol 21(3):23. https://doi.org/10.1145/3382770
48. Ferrag MA, Babaghayou M, Yazici MA (2020) Cyber security for fog-based smart grid SCADA systems: solutions and challenges. J Info Secur Appl 52:102500. https://doi.org/10.1016/j.jisa.2020.102500
49. Sarma R, Kumar C, Barbhuiya FA (2021) PAC-FIT: an efficient privacy preserving access control scheme for fog-enabled IoT. Sustain Comput: Info Syst 30:100527. https://doi.org/10.1016/j.suscom.2021.100527
50. Stojmenovic I, Wen S (2014) The Fog computing paradigm: scenarios and security issues. Fed Conf Comput Sci Info Syst 2014:1–8. https://doi.org/10.15439/2014F503
51. Yang R, Yu FR, Si P, Yang Z, Zhang Y (2019) Integrated blockchain and edge computing systems: a survey, some research issues and challenges. IEEE Commun Surv Tutor 21(2):1508–1532. https://doi.org/10.1109/COMST.2019.2894727
52. Liu Y, Yu FR, Li X, Ji H, Leung VCM (2019) Decentralized resource allocation for video transcoding and delivery in lockchain-based system with mobile edge computing. IEEE Trans Veh Technol 68(11):11169–11185
53. Kang J, Yu R, Huang X, Wu M, Maharjan S, Xie S, Zhang Y (2018) Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE IoT J 6:4660–4670
54. Hosseinioun P, Kheirabadi M, Kamel Tabbakh SR, Ghaemi R (2020) Atask scheduling approaches in fog computing: a survey. Trans Emerg Tel Tech 2020:e3792. https://doi.org/10.1002/ett.3792

55. Yang X, Rahmani N (2020) Task scheduling mechanisms in fog computing: review, trends, and perspectives. Kybernetes. https://doi.org/10.1108/K-10-2019-0666
56. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA (2018) A Comprehensive survey on fog computing: state-of-the-art and research challenges. IEEE Commun Surv Tutor 20(1):416–464. https://doi.org/10.1109/COMST.2017.2771153
57. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. Future Gener Comput Syst 97:219–235. https://doi.org/10.1016/j.future.2019.02.050
58. Aladwani Tahani (2019) Scheduling IoT healthcare tasks in fog computing based on their importance. Proc Comput Sci 163:560–569. https://doi.org/10.1016/j.procs.2019.12.138
59. Khan OA, Malik SUR, Baig FM, Islam SU, Pervaiz H, Malik H, Ahmed SH (2020) A cache-based approach toward improved scheduling in fog computing. Softw Pract Exper 2020:1–13. https://doi.org/10.1002/spe.2824
60. Rausch Thomas, Rashed Alexander, Dustdar Schahram (2020) Optimized container scheduling for data-intensive serverless edge computing. Future Gener Comput Syst 114:259–271. https://doi.org/10.1016/j.future.2020.07.017
61. Madhura R, Elizabeth BL, Uthariaraj VR (2021) An improved list-based task scheduling algorithm for fog computing environment. Computing. https://doi.org/10.1007/s00607-021-00935-9
62. Bitam S, Zeadally S, Mellouk A (2018) Fog computing job scheduling optimization based on bees swarm. Enterprise Infor Syst 12(4):373–397. https://doi.org/10.1080/17517575.2017.1304579
63. Boveiri HR, Khayami R, Elhoseny M, Gunasekaran M (2019) An efficient Swarm-Intelligence approach for task scheduling in cloud based internet of things applications. J Ambient Intell Humaniz Comput 10:3469–3479
64. Barzegaran M, Karagiannis† V, Avasalcai C, Pop P, Schulte S, Dustdar S (2020) Towards extensibility-aware scheduling of industrial applications on fog nodes. In: IEEE International Conference on Edge Computing (EDGE 2020), pp. 1–9
65. Wang S, Li Y, Pang S, Qinghua L, Wang S, Zhao J (2020) A task scheduling strategy in edge-cloud collaborative scenario based on deadline. Sci Program 2020:9. https://doi.org/10.1155/2020/3967847
66. Barzegaran M, Cervin A, Pop P (2020) Performance optimization of control applications on fog computing platforms using scheduling and isolation. IEEE Access 8:104085–104098. https://doi.org/10.1109/ACCESS.2020.2999322
67. Li G, Liu Y, Junhua W, Lin D, Zhao S (2019) Methods of resource scheduling based on optimized fuzzy clustering in fog computing. Sensors. https://doi.org/10.3390/s19092122
68. Wang Juan, Li Di (2019) Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing. Sensors 19(5):1023. https://doi.org/10.3390/s19051023
69. Yang Y, Wang K, Zhang G, Chen X, Luo X, Zhou MT (2018) MEETS: maximal energy efficient task scheduling in homogeneous fog networks. IEEE Internet Things J 5:4076–4087
70. Auluck N, Azim A, Fizza K (2019) Improving the schedulability of real-time tasks using fog computing. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2019.2944360
71. Deng R, Lu R, Lai C, Luan TH, Liang H (2016) Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet Things J 3(6):1171–1181
72. Zhang G, Shen F, Chen N, Zhu P, Dai X, Yang Y (2018) DOTS: delay-optimal task scheduling among voluntary nodes in fog networks. IEEE Internet Things J 6:3533–3544
73. Zheng X, Li M, Guo J (2020) Task scheduling using edge computing system in smart city. Int J Commun Syst 2020:e4422. https://doi.org/10.1002/dac.4422
74. Huang J, Li S, Chen Y (2020) Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing. Peer-to-Peer Netw Appl 13(5):1776–1787. https://doi.org/10.1007/s12083-020-00880-y
75. Fellir F, El Attar A, Nafil K, Chung L (2020) A multi-Agent based model for task scheduling in cloud-fog computing platform. In: 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT), Doha, Qatar, pp. 377–382. https://doi.org/10.1109/ICIoT48696.2020.9089625
76. Huang X, Yu R, Ye D, Shu L, Xie S (2021) Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing. IEEE Trans Veh Technol 70(4):3773–3787. https://doi.org/10.1109/TVT.2021.3064426
77. Cao B et al (2021) Edge-cloud resource scheduling in space-air-ground integrated networks for internet of vehicles. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3065583

78. Guevara JC, da Fonseca NLS (2021) Task scheduling in cloud-fog computing systems. Peer-to-Peer Netw Appl 14:962–977. https://doi.org/10.1007/s12083-020-01051-9
79. Zeng D, Gu L, Guo S, Cheng Z, Yu S (2016) Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans Comput 65(12):3702–3712
80. Chen X, Wang L (2017) Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—PEPA. IEEE Commun Lett 21(4):745–748
81. Bittencourt LF, Diaz-Montes J, Buyya R, Rana OF, Parashar M (2017) Mobility-aware application scheduling in fog computing. IEEE Tech Comm Cloud Comput (TCCLD) 4(2):26–35
82. Abdelmoneem RM, Benslimane A, Shaaban E (2020) Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures. Comput Netw 179:107348. https://doi.org/10.1016/j.comnet.2020.107348
83. Liu B, Xiaolong X, Qi L, Ni Q, Dou W (2020) Task scheduling with precedence and placement constraints for resource utilization improvement in multi-user MEC environment. J Syst Arch 114:101970. https://doi.org/10.1016/j.sysarc.2020.101970
84. Hao Y, Cao J, Wang Q, Jinglin D (2020) Energy-aware scheduling in edge computing with a clustering method. Future Gener Comput Syst 117:259–272. https://doi.org/10.1016/j.future.2020.11.029
85. Sun Y, Lin F, Xu H (2018) Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. Wireless Pers Commun 102:1369–1385. https://doi.org/10.1007/s11277-017-5200-5
86. Nguyen BM, Thi Thanh Binh H, Do SB (2019) Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment. Appl Sci 9(9):1730
87. Xu J, Hao Z, Zhang R, Sun X (2019) A method based on the combination of laxity and ant colony system for cloud-fog task scheduling. IEEE Access 7:116218–116226
88. Abdel-Basset* M, Mohamed R, Elhoseny M, Bashir AK, Jolfaei A, Kumar N (2020) Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications. 2020.3001067, IEEE transactions on industrial informatics
89. Abdel-Basset M, El-shahat D, Elhoseny M, Song H (2020) Energy aware meta heuristic algorithm for industrial internet of things task scheduling problems in fog computing applications. IEEE Internet Things J 8:12638–12649
90. Ghobaei-Arani M, Souri A, Safara F, Norouzi M (2019) An efficient task scheduling approach using moth-flame optimization algorithm for cyber-physical system applications in fog computing. Trans Emerg Tel Tech. https://doi.org/10.1002/ett.3770
91. Wu C-G, Wang L (2020) An estimation of distribution algorithm to optimize the utility of task scheduling under fog computing systems. In: Fog computing (eds Zomaya A, Abbas A, Khan S). https://doi.org/10.1002/9781119551713.ch14
92. Liu Y, Wang S, Zhao Q, Shiyu Du, Zhou Ao, Ma X, Yang F (2020) Dependency-aware task scheduling in vehicular edge computing. IEEE Internet Things J 7(6):4961–4971. https://doi.org/10.1109/JIOT.2020.2972041
93. Sun H, Yu H, Fan G (2020) Contract-based resource sharing for time effective task scheduling in fog-cloud environment. IEEE Trans Netw Serv Manag 17(2):1040–1053. https://doi.org/10.1109/TNSM.2020.2977843
94. Wang S, Zhao T, Pang S (2020) Task scheduling algorithm based on improved firework algorithm in fog computing. In IEEE Access 8:32385–32394. https://doi.org/10.1109/ACCESS.2020.2973758
95. Wan J, Chen B, Wang S, Xia M, Li D, Liu C (2018) Fog computing for energy-aware load balancing and scheduling in smart factory. IEEE Trans Industr Inf 14:4548–4556
96. Hosseinzadeh M, Masdari M, Rahmani AM et al (2021) Improved butterfly optimization algorithm for data placement and scheduling in edge computing environments. J Grid Comput 19:14. https://doi.org/10.1007/s10723-021-09556-0
97. Hu F, Lv L, Zhang T, Shi Y (2021) Vehicular task scheduling strategy with resource matching computing in cloud-edge collaboration. IET Collab Intell Manuf. https://doi.org/10.1049/cim2.12023
98. Hosseinioun P, Kheirabadi M, Tabbakh SRK, Ghaemi R (2020) A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm. J Parallel Distrib Comput 143:88–96. https://doi.org/10.1016/j.jpdc.2020.04.008

99. Javanmardi S, Shojafar M, Persico V, Pescapè A (2020) FPFTS: a joint fuzzy particle swarm optimization mobility-aware approach to fog task scheduling algorithm for Internet of Things devices. Softw Pract Exper. https://doi.org/10.1002/spe.2867
100. Farhat P, Sami H, Mourad A (2020) Reinforcement R-learning model for time scheduling of on-demand fog placement. J Supercomput 76(1):388–410
101. Gazori P, Rahbari D, Nickray M (2020) Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach. Future Gener Comput Syst 110:1098–1115
102. Bhatia M, Sood SK, Kaur S (2019) Quantum-based predictive fog scheduler for IoT applications. Comput Indus 111:51–67. https://doi.org/10.1016/j.compind.2019.06.002
103. Ullah Y (2020) Task classification and scheduling based on k-means clustering for edge computing. Wireless Pers Commun 113:2611–2624. https://doi.org/10.1007/s11277-020-07343-w
104. Tang Z, Jia W, Zhou X, Yang W, You Y (2020) Representation and reinforcement learning for task scheduling in edge computing. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2020.2990558
105. Sheng S, Chen P, Chen Z, Wu L, Yao Y (2021) Deep reinforcement learning-based task scheduling in iot edge computing. Sensors 21:1666. https://doi.org/10.3390/s21051666
106. Yan Li, Zhu Z, Chen B (2020) EASE: Energy-efficient task scheduling for edge computing under uncertain runtime and unstable communication conditions. Concurrency Computat Pract Exper 2019:e5465. https://doi.org/10.1002/cpe.5465
107. Aburukba RO, AliKarrar M, Landolsi T, El-Fakih K (2020) Scheduling Internet of Things requests to minimize latency in hybrid fog-cloud computing. Future Gener Comput Syst 111:539–551. https://doi.org/10.1016/j.future.2019.09.039
108. Zhu Z, Zhang J, Zhao J, Cao J, Zhao D, Jia G, Meng Q (2019) A hardware and software task-scheduling framework based on CPU+FPGA heterogeneous architecture in edge computing. IEEE Access 7:148975–148988. https://doi.org/10.1109/ACCESS.2019.2943179
109. Mastoi Q-u-a, Wah TY, Raj RG, Lakhan A (2020) A novel cost-efficient framework for critical heartbeat task scheduling using the internet of medical things in a fog cloud system. Sensors 20:441
110. De Benedetti M, Messina F, Pappalardo G, Santoro C (2017) JarvSis: a distributed scheduler for IoT applications. Cluster Comput 20:1775–1790. https://doi.org/10.1007/s10586-017-0836-1
111. Xiong G, Singh R, Li J (2021) Learning augmented index policy for optimal service placement at the network edge,arXiv:2101.03641
112. Vinueza PG, Naranjo ZP, Shojafar M, Conti M, Buyya R (2019) FOCAN: a Fog-supported smart city network architecture for management of applications in the Internet of Everything environments. J Parallel Distrib Comput 132:274–283. https://doi.org/10.1016/j.jpdc.2018.07.003
113. Tran MQ, Nguyen DT, Le VA, Nguyen DH, Pham TV (2019) Task placement on fog computing made efficient for iot application provision. Wireless Commun Mobile Comput 2019:17. https://doi.org/10.1155/2019/6215454
114. Mahmud R, Srirama SN, Ramamohanarao K, Buyya R (2019) Quality of Experience (QoE)-aware placement of applications in Fog computing environments. J Parallel Distrib Comput 132:190–203. https://doi.org/10.1016/j.jpdc.2018.03.004
115. Yao H, Bai C, Xiong M, Zeng D, Fu Z (2017) Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing. Concurrency Computat: Pract Exper 29:e3975. https://doi.org/10.1002/cpe.3975
116. Venticinque S, Amato A (2019) A methodology for deployment of IoT application in fog. J Ambient Intell Human Comput 10:1955–1976. https://doi.org/10.1007/s12652-018-0785-4
117. Souza VB, Masip-Bruin X, Marín-Tordera E, Sànchez-López S, Garcia J, Ren GJ, Jukan A, Juan Ferrer A (2018) Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. Future Gener Comput Syst 87:1–15. https://doi.org/10.1016/j.future.2018.04.042
118. Selimi M, Cerdà-Alabern L, Freitag F, Veiga L, Sathiaseelan A, Crowcroft J (2019) A lightweight service placement approach for community network micro-clouds. J Grid Comput 17:169–189. https://doi.org/10.1007/s10723-018-9437-3
119. Mahmud R, Ramamohanarao K, Buyya R (2018) Latency-aware application module management for fog computing environments. ACM Trans Internet Technol 19(1):2. https://doi.org/10.1145/3186592

120. Zeng D, Lin G, Yao H (2020) Towards energy efficient service composition in green energy powered cyber-physical fog systems. Future Gener Comput Syst 105:757–765. https://doi.org/10.1016/j.future.2018.01.060
121. Moubayed A, Shami A, Heidari P, Larabi A, Brunner R (2020) Edge-enabled V2X service placement for intelligent transportation systems. In IEEE Trans Mobile Comput. https://doi.org/10.1109/TMC.2020.2965929
122. Khosroabadi F, Fotouhi-Ghazvini F, Fotouhi H (2021) SCATTER: service placement in real-time fog-assisted IoT Networks. J Sens Actuator Netw 10:26. https://doi.org/10.3390/jsan10020026
123. Xiao T, Cui T, Islam SMR, Chen Q (2021) Joint Content placement and storage allocation based on federated learning in F-RANs. Sensors 21:215. https://doi.org/10.3390/s21010215
124. Velasquez K, Abreu DP, Curado M, Monteiro E (2017) Service placement for latency reduction in the internet of things. Ann Telecommun 72:105–115. https://doi.org/10.1007/s12243-016-0524-9
125. Mahmud R, Srirama SN, Ramamohanarao K, Buyya R (2020) Profit-aware application placement for integrated Fog-Cloud computing environments. J Parallel Distrib Comput 135:177–190. https://doi.org/10.1016/j.jpdc.2019.10.001
126. Baranwal G, Yadav R, Vidyarthi DP (2020) QoE Aware IoT application placement in fog computing using modified-TOPSIS. Mobile Netw Appl 25(5):1816–1832. https://doi.org/10.1007/s11036-020-01563-x
127. Bi S, Huang L, Zhang YA (2020) Joint optimization of service caching placement and computation offloading in mobile edge computing systems. IEEE Trans Wireless Commun 19(7):4947–4963. https://doi.org/10.1109/TWC.2020.2988386
128. Nashaat H, Ahmed E, Rizk R (2020) IoT application placement algorithm based on multi-dimensional QoE prioritization model in fog computing environment. IEEE Access 8:111253–111264. https://doi.org/10.1109/ACCESS.2020.3003249
129. Bermbach D, Maghsudi S, Hasenburg J, Pfandzelter T (2020) Towards auction-based function placement in serverless fog platforms. In: IEEE international conference on fog computing (ICFC)
130. Lin C, Yang J (2018) Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0. IEEE Trans Indus Info 14(10):4603–4611. https://doi.org/10.1109/TII.2018.2827920
131. Skarlat O, Nardelli M, Schulte S, Borkowski M, Leitner P (2017) Optimized IoT service placement in the fog. SOCA 11:427–443. https://doi.org/10.1007/s11761-017-0219-8
132. Goudarzi M, Wu H, Palaniswami MS, Buyya R (2020) An application placement technique for concurrent IoT applications in edge and fog computing environments. IEEE Trans Mobile Comput. https://doi.org/10.1109/TMC.2020.2967041
133. Tham C, Chattopadhyay R (2017). A load balancing scheme for sensing and analytics on a mobile edge computing network. In 2017 IEEE 18th international symposium on a world of wireless, mobile and multimedia networks (WoWMoM), Macau, pp. 1–9. https://doi.org/10.1109/WoWMoM.2017.7974307
134. Talaat FM, Saraya MS, Saleh AI, Ali HA, Ali SH (2020) A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01768-8
135. Ningning S, Chao G, Xingshuo A, Qiang Z (2016) Fog computing dynamic load balancing mechanism based on graph repartitioning. Chin Commun 13(3):156–164. https://doi.org/10.1109/CC.2016.7445510
136. Hosono K, Maki A, Watanabe Y, Takada H, Sato K (2021) Implementation and evaluation of load balancing mechanism with multiple edge server cooperation for dynamic map system. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2021.3067909
137. Gu L, Zeng D, Guo S, Barnawi A, Xiang Y (2017) Cost efficient resource management in fog computing supported medical cyber-physical system. IEEE Trans Emerg Topics Comput 5(1):108–119. https://doi.org/10.1109/TETC.2015.2508382
138. Kapsalis A, Kasnesis P, Venieris IS, Kaklamani DI, Patrikakis CZ (2017) A cooperative fog approach for effective workload balancing. IEEE Cloud Comput 4(2):36–45. https://doi.org/10.1109/MCC.2017.25
139. Alqahtani F, Amoon M, Nasr AA (2021) Reliable scheduling and load balancing for requests in cloud-fog computing. Peer-to-Peer Netw Appl. https://doi.org/10.1007/s12083-021-01125-2

140. Hameed AR, Islam SU, Ahmad I, Munir K (2020) Energy- and performance-aware load-balancing in vehicular fog computing. Sustain Comput: Info Syst 30: 100454. https://doi.org/10.1016/j.suscom.2020.100454
141. Li C, Zhuang H, Wang Q, Zhou X (2018) SSLB: self-similarity- based load balancing for large-scale fog computing. Arab J Sci Eng 43:7487–7498. https://doi.org/10.1007/s13369-018-3169-3
142. Beraldi R, Mtibaa A, Alnuweiri H (2017) Cooperative load balancing scheme for edge computing resources. In: 2017 second international conference on fog and mobile edge computing (FMEC), Valencia, pp. 94–100. https://doi.org/10.1109/FMEC.2017.7946414.
143. Ouyang W, Chen Z, Wu J, Yu G, Zhang H (2021) Dynamic task migration combining energy efficiency and load balancing optimization in three-tier UAV-enabled mobile edge computing system. Electronics 10:190. https://doi.org/10.3390/electronics10020190
144. Manasrah AM, Aldomi A, Gupta BB (2019) An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. Cluster Comput 22(1):1639–1653. https://doi.org/10.1007/s10586-017-1559-z
145. He X, Ren Z, Shi C, Fang J (2016) A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles. Chin Commun 13(Supplement 2):140–149. https://doi.org/10.1109/CC.2016.7833468
146. Lim J, Lee D (2020) A load balancing algorithm for mobile devices in edge cloud computing environments. Electronics 9(4):686
147. Xiaolong X, Shucun F, Cai Q, Tian W, Liu W, Dou W, Sun X, Liu AX (2018) Dynamic resource allocation for load balancing in fog environment. Wireless Commun Mobile Comput 2018:15. https://doi.org/10.1155/2018/6421607
148. Zhang F, Wang MM (2020) Stochastic congestion game for load balancing in mobile edge computing. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2020.3008009
149. Qin Z, Cheng Z, Lin C, Zhaoyi L, Wang L (2021) Optimal workload allocation for edge computing network using application prediction. Wireless Commun Mobile Comput 2021:13. https://doi.org/10.1155/2021/5520455
150. Hung Y-W, ChenY-C, Lo C, So AG, Chang SC (2021) Dynamic workload allocation for edge computing. IEEE Trans Very Large Scale Integ (VLSI) Syst 29(3): 519–529. https://doi.org/10.1109/TVLSI.2021.3049520
151. Chen J, Xing H, Lin X, Nallanathan A, Bi S (2021) Joint resource allocation and cache placement for location-aware multi-user mobile edge computing, arXiv:2103.11220
152. Liu X, Yu J, Wang J, Gao Y (2020) Resource allocation with edge computing in IoT networks via machine learning. IEEE Internet Things J 7(4):3415–3426. https://doi.org/10.1109/JIOT.2020.2970110
153. Wang J, Zhao L, Liu J, Kato N (2019) Smart resource allocation for mobile edge computing: a deep reinforcement learning approach. IEEE Trans Emerg Topics Comput. https://doi.org/10.1109/TETC.2019.2902661
154. Chen X, Liu G (2021) Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3050804
155. Seid AM, Boateng GO, Anokye S, Kwantwi T, Sun G, Liu G (2021) Collaborative computation offloading and resource allocation in multi-UAV assisted iot networks: a deep reinforcement learning approach. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3063188
156. Li D, Xu S, Li P (2021) Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular V2X networks. Sensors 21:372. https://doi.org/10.3390/s21020372
157. Deng S, Xiang Z, Zhao P, Taheri J, Gao H, Yin J, Zomaya AY (2020) Dynamical resource allocation in edge for trustable internet-of-things systems: a reinforcement learning method. IEEE Trans Indus Info 16(9):6103–6113. https://doi.org/10.1109/TII.2020.2974875
158. Vimal S, Khari M, Dey N, Crespo RG, Harold Robinson Y (2020) Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT. Comput Commun 151:355–364. https://doi.org/10.1016/j.comcom.2020.01.018
159. Jiao Y, Wang P, Niyato D, Suankaewmanee K (2019) Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. IEEE Trans Parallel Distrib Syst 30(9):1975–1989. https://doi.org/10.1109/TPDS.2019.2900238
160. Anglano C, Canonico M, Guazzone M (2018) Profit-aware resource management for edge computing systems. In: EdgeSys '18: International Workshop on Edge Systems, Analytics and

Networking, June10–15, Munich, Germany. ACM, New York, NY, USA, 7 pages.https://doi.org/10.1145/3213344.3213349
161. Alsaffar AA, Pham HP, Hong C-S, Huh E-N, Aazam M (2016) An architecture of IoT service delegation and resource allocation based on collaboration between fog and cloud computing. Mobile Info Syst 2016:15. https://doi.org/10.1155/2016/6123234
162. Baek B, Lee J, Peng Y, Park S (2020) Three dynamic pricing schemes for resource allocation of edge computing for IoT environment. IEEE Internet Things J 7(5):4292–4303. https://doi.org/10.1109/JIOT.2020.2966627
163. Zhang H, Xiao Y, Bu S, Niyato D, Yu FR, Han Z (2017) Computing resource allocation in three-tier IoT fog networks: a joint optimization approach combining stackelberg game and matching. IEEE Internet Things J 4(5):1204–1215. https://doi.org/10.1109/JIOT.2017.2688925
164. Zhang H, Zhang Y, Gu Y, Niyato D, Han Z (2017) A hierarchical game framework for resource management in fog computing. IEEE Commun Mag 55(8):52–57. https://doi.org/10.1109/MCOM.2017.1600896
165. Duo L, Li Q, Haitao X, Zhou Y (2020) Dynamic priority-based service resource allocation for context-aware conflict resolution in wisdom network with fog computing. Wireless Commun Mobile Comput 2020:7. https://doi.org/10.1155/2020/8812482
166. Cui G, He Q, Chen F, Zhang Y, Jin H, Yang Y (2021) Interference-aware game-theoretic device allocation for mobile edge computing. IEEE Trans Mobile Comput. https://doi.org/10.1109/TMC.2021.3064063
167. Nguyen DT, Le LB, Bhargava V (2018) Price-based resource allocation for edge computing: a market equilibrium approach. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2018.2844379
168. Ni L, Zhang J, Jiang C, Yan C, Yu K (2017) Resource allocation strategy in fog computing based on priced timed petri nets. IEEE Internet Things J 4(5):1216–1228. https://doi.org/10.1109/JIOT.2017.2709814
169. Xu J, Palanisamy B, Ludwig H, Wang Q (2017) Zenith: utility-aware resource allocation for edge computing. In: 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, pp. 47–54. https://doi.org/10.1109/IEEE.EDGE.2017.15
170. Kochar V, Sarkar A (2016) Real time resource allocation on a dynamic two level symbiotic fog architecture. In: 2016 Sixth International Symposium on Embedded Computing and System Design (ISED), Patna, pp. 49–55. https://doi.org/10.1109/ISED.2016.7977053.
171. Naranjo P, Pooranian Z, Shamshirband S, Abawajy J, Conti M (2017) Fog over virtualized IoT: new opportunity for context-aware networked applications and a case study. Appl Sci 7(12):1325
172. Zhang W, Zhang Z, Chao H (2017) Cooperative fog computing for dealing with big data in the internet of vehicles: architecture and hierarchical resource management. IEEE Commun Mag 55(12):60–67. https://doi.org/10.1109/MCOM.2017.1700208
173. Zhao L, Wang J, Liu J, Kato N (2019) Optimal edge resource allocation in IoT-based smart cities. IEEE Netw 33(2):30–35. https://doi.org/10.1109/MNET.2019.1800221
174. Naha RK, Garg S (2021) Multi-criteria–based dynamic user behaviour–aware resource allocation in fog computing. ACM Trans Internet Things 2(1):1–31. https://doi.org/10.1145/3423332
175. Wang J, Wang L (2021) A computing resource allocation optimization strategy for massive internet of health things devices considering privacy protection in cloud edge computing environment. J Grid Comput 19:17. https://doi.org/10.1007/s10723-021-09558-y
176. Peng X, Ota K, Dong M (2020) Multiattribute-based double auction toward resource allocation in vehicular fog computing. IEEE Internet Things J 7(4):3094–3103. https://doi.org/10.1109/JIOT.2020.2965009
177. Chen X, Zhou Y, Yang L (2021) Lu Lv, Hybrid fog/cloud computing resource allocation: joint consideration of limited communication resources and user credibility. Comput Commun 169:48–58. https://doi.org/10.1016/j.comcom.2021.01.026
178. Cao B, Sun Z, Zhang J, Gu Y (2020) Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2020.3048844
179. Bai T, Pan C, Ren H, Deng Y, Elkashlan M, Nallanathan A (2021) Resource allocation for intelligent reflecting surface aided wireless powered mobile edge computing in OFDM Systems. IEEE Trans Wireless Commun. https://doi.org/10.1109/TWC.2021.3067709
180. Li WT, Zhao M, Wu YH et al (2021) Collaborative offloading for UAV-enabled time-sensitive MEC networks. J Wireless Com Network. https://doi.org/10.1186/s13638-020-01861-8

181. Zhang K, Peng M, Sun Y (2020) Delay-optimized resource allocation in fog based vehicular networks. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2020.3010861
182. Fu Y, Yang X, Yang P et al (2021) Energy-efficient offloading and resource allocation for mobile edge computing enabled mission-critical internet-of-things systems. J Wireless Com Netw. https://doi.org/10.1186/s13638-021-01905-7
183. Baidas MW (2021) Resource allocation for offloading-efficiency maximization in clustered NOMA-enabled mobile edge computing networks. Comput Netw 189:107919. https://doi.org/10.1016/j.comnet.2021.107919
184. Chang L, Liu XG, Sheng Q (2020) Dynamic resource allocation and computation offloading for IoT fog computing system. IEEE Trans Indus Info. https://doi.org/10.1109/TII.2020.2978946
185. Feng J, Liu L, Pei Q, Hou F, Yang T, Wu J (2021) Service characteristics-oriented joint optimization of radio and computing resource allocation in mobile-edge computing. IEEE Internet Things J 8(11):9407–9421. https://doi.org/10.1109/JIOT.2021.3058363
186. Liao H, Zhou Z, Zhao X, Wang Y (2021) Learning-based queue-aware task offloading and resource allocation for space–air–ground-integrated power IoT. IEEE Internet Things J 8(7):5250–5263. https://doi.org/10.1109/JIOT.2021.3058236
187. Huang X, Zhang W, Yang J, Yang L, Yeo CK (2021) Market-based dynamic resource allocation in mobile edge computing systems with multi-server and multi-user. Comput Commun 165:43–52. https://doi.org/10.1016/j.comcom.2020.11.001
188. Abbasi M, Mohammadi-Pasand E, Khosravi MR (2021) Intelligent workload allocation in IoT–Fog–cloud architecture towards mobile edge computing. Comput Commun 169:71–80. https://doi.org/10.1016/j.comcom.2021.01.022
189. Wang Y, Chen C-R, Huang P-Q, Wang K (2021) A new differential evolution algorithm for joint mining decision and resource allocation in a MEC-enabled wireless blockchain network. Comput Indus Eng 155:107186. https://doi.org/10.1016/j.cie.2021.107186
190. Jiang C, Li Y, Su J et al (2021) Research on new edge computing network architecture and task offloading strategy for Internet of Things. Wireless Netw. https://doi.org/10.1007/s11276-020-02516-8
191. Elbamby B, Saad H (2018) Proactive edge computing in fog networks with latency and reliability guarantees. J Wireless Com Network 2018:209. https://doi.org/10.1186/s13638-018-1218-y
192. Nair B, Somasundaram MSB (2019) Overload prediction and avoidance for maintaining optimal working condition in a fog node. Comput Electr Eng 77:147–162. https://doi.org/10.1016/j.compeleceng.2019.05.011
193. Verba N, Chao K-M, Lewandowski J, Shah N, James A, Tian F (2019) Modeling industry 4.0 based fog computing environments for application analysis and deployment. Future Gener Comput Syst 91:48–60. https://doi.org/10.1016/j.future.2018.08.043
194. Guo J, Li C, Chen Y, Luo Y (2020) On-demand resource provision based on load estimation and service expenditure in edge cloud environment. J Netw Comput Appl 151:102506. https://doi.org/10.1016/j.jnca.2019.102506
195. Li S, Wu W (2018) Method of resource estimation based on QoS in edge computing. Wireless Commun Mobile Comput 2018:1–9. https://doi.org/10.1155/2018/7308913
196. Aazam M, St-Hilaire M, Lung C-H, Lambadaris I, Huh E-N (2018) IoT resource estimation challenges and modeling in fog. Fog Comput Internet Things. https://doi.org/10.1007/978-3-319-57639-8_2
197. Farahbakhsh F, Shahidinejad A, Ghobaei-Arani M (2020) Multi-user context-aware computation offloading in mobile edge computing based on Bayesian learning automata. Trans Emerg Telecommun Technol. https://doi.org/10.1002/ett.4127
198. Shakarami A, Shahidinejad A, Ghobaei-Arani M (2021) An autonomous computation offloading strategy in mobile edge Computing: a deep learning-based hybrid approach. J Netw Comput Appl 178:102974. https://doi.org/10.1016/j.jnca.2021.102974
199. Chu CH (2021) Task offloading based on deep learning for blockchain in mobile edge computing. Wireless Netw 27:117–127. https://doi.org/10.1007/s11276-020-02444-7
200. Yu B, Zhang X, You I, Khan US (2021) Efficient computation offloading in edge computing enabled smart home. IEEE Access 9:48631–48639. https://doi.org/10.1109/ACCESS.2021.3066789
201. Li Z, Chang V, Ge J et al (2021) Energy-aware task offloading with deadline constraint in mobile edge computing. J Wireless Com Netw. https://doi.org/10.1186/s13638-021-01941-3

202. Min M, Xiao L, Chen Y, Cheng P, Wu D, Zhuang W (2019) Learning-based computation offloading for iot devices with energy harvesting. IEEE Trans Veh Technol 68:1930–1941. https://doi.org/10.1109/TVT.2018.2890685
203. Chen C, Zhang Y, Wang Z, Wan S, Pei Q (2021) Distributed computation offloading method based on deep reinforcement learning in ICV. Appl Soft Comput 103:107108. https://doi.org/10.1016/j.asoc.2021.107108
204. Li H, Fang F, Ding Z (2021) DRL-assisted resource allocation for NOMA-MEC offloading with hybrid SIC. Entropy 23:613. https://doi.org/10.3390/e23050613
205. Weng Y, Chu H, Shi Z (2021) An intelligent offloading system based on multiagent reinforcement learning. Secur Commun Netw 2021:1–13. https://doi.org/10.1155/2021/8830879
206. Liu T, Zhang Y, Zhu Y, Tong W, Yang Y (2021) Online computation offloading and resource scheduling in mobile-edge computing. IEEE Internet Things J 8(8):6649–6664. https://doi.org/10.1109/JIOT.2021.3051427
207. Li C et al (2021) Dynamic offloading for multiuser Muti-CAP MEC networks: a deep reinforcement learning approach. IEEE Trans Veh Technol 70(3):2922–2927. https://doi.org/10.1109/TVT.2021.3058995
208. Tuong VD, Truong TP, Nguyen T-V, Noh W, Cho S (2021) Partial computation offloading in NOMA-assisted mobile edge computing systems using deep reinforcement learning. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3064995
209. Aazam M, Zeadally S, Flushing EF (2021) Task offloading in edge computing for machine learning-based smart healthcare. Comput Netw 191:108019. https://doi.org/10.1016/j.comnet.2021.108019
210. Manogaran G, Mumtaz S, Mavromoustakis CX, Pallis E, Mastorakis G (2021) Artificial intelligence and blockchain-assisted offloading approach for data availability maximization in edge nodes. IEEE Trans Veh Technol 70(3):2404–2412. https://doi.org/10.1109/TVT.2021.3058689
211. Cheng N, Lyu F, Quan W, Zhou C, He H, Shi W, Shen X (2019) Space/aerial-assisted computing offloading for Iot applications: a learning-based approach. IEEE J Sel Areas Commun 37:1117–1129. https://doi.org/10.1109/JSAC.2019.2906789
212. Wei Z, Zhao B, Su J, Lu X (2019) Dynamic edge computation offloading for internet of things with energy harvesting: a learning method. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2018.2882783
213. Xu J, Chen L, Ren S (2017) Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Trans Cognit Commun Netw 3:361–373. https://doi.org/10.1109/TCCN.2017.2725277
214. Zheng X, Li M, Tahir M, Chen Y, Alam M (2019) Stochastic computation offloading and scheduling based on mobile edge computing. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2919651
215. Zhu S, Gui L, Zhao D, Cheng N, Zhang Q, Lang X (2021) Learning-based computation offloading approaches in UAVs-assisted edge computing. IEEE Trans Veh Technol 70(1):928–944. https://doi.org/10.1109/TVT.2020.3048938
216. Lin P, Song Q, Yu FR, Wang D, Guo L (2021) Task offloading for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3051419
217. Aljanabi S, Chalechale A (2021) Improving IoT services using a hybrid fog-cloud offloading. IEEE Access 9:13775–13788. https://doi.org/10.1109/ACCESS.2021.3052458
218. Hossain MD, Sultana T, Hossain MA, Hossain MI, Huynh LNT, Park J, Huh E-N (2021) Fuzzy decision-based efficient task offloading management scheme in multi-tier MEC-enabled networks. Sensors 2021(21):1484. https://doi.org/10.3390/s21041484
219. Chen X, Jiao L, Li W, Xiaoming F (2016) Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans Netw 24(5):2795–2808. https://doi.org/10.1109/TNET.2015.2487344
220. Xiong Z, Feng S, Wang W, Niyato D, Wang P, Han Z (2018) Cloud/fog computing resource management and pricing for blockchain networks. IEEE Internet Things J 6(3):4585–4600. https://doi.org/10.1109/JIOT.2018.2871706
221. Cao H, Cai J (2018) Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: a game-theoretic machine learning approach. IEEE Trans Veh Technol 67:752–764. https://doi.org/10.1109/TVT.2017.2740724

222. Liu L, Chang Z, Guo X (2018) Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices. IEEE Internet Things J 5(3):1869–1879. https://doi.org/10.1109/JIOT.2018.2816682
223. Liu Y, Changqiao X, Zhan Y, Liu Z, Guan J, Zhang H (2017) Incentive mechanism for computation offloading using edge computing: a Stackelberg game approach. Comput Netw 129(2):399–409. https://doi.org/10.1016/j.comnet.2017.03.015
224. Zuo Y, Jin S, Zhang S, Zhang Y (2021) Blockchain storage and computation offloading for cooperative mobile-edge computing. IEEE Internet Things J 8(11):9084–9098. https://doi.org/10.1109/JIOT.2021.3056656
225. Zhao L et al (2021) Vehicular computation offloading for industrial mobile edge computing. IEEE Trans Indus Info. https://doi.org/10.1109/TII.2021.3059640
226. Dai M, Su Z, Xu Q, Zhang N (2021) Vehicle assisted computing offloading for unmanned aerial vehicles in smart city. IEEE Trans Intell Transp Syst 22(3):1932–1944. https://doi.org/10.1109/TITS.2021.3052979
227. Yang X, Luo H, Sun Y, Zou J, Guizani M (2021) Coalitional game based cooperative computation offloading in MEC for reusable tasks. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3064186
228. Yang Y, Long C, Wu J, Peng S, Li B (2021) D2D-enabled mobile-edge computation offloading for multi-user IoT Network. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3068722
229. Ren Y, Xie Z, Ding Z, Sun X, Xia J, Tian Y (2021) Computation offloading game in multiple unmanned aerial vehicle-enabled mobile edge computing networks. IET Commun 2021:1–10. https://doi.org/10.1049/cmu2.12102
230. Apostolopoulos PA, Fragkos G, Tsiropoulou EE, Papavassiliou S (2021) Data offloading in UAV-assisted multi-access edge computing systems under resource uncertainty. IEEE Trans Mobile Comput. https://doi.org/10.1109/TMC.2021.3069911
231. Zhou J, Tian D, Sheng Z, Duan X, Shen X (2021) Distributed task offloading optimization with queueing dynamics in multi-agent mobile-edge computing networks. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3063509
232. Abdenacer N, Wu H, Abdelkader NN, Dhelim S, Ning H (2021) A novel framework for mobile edge computing by optimizing task offloading. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3064225
233. Song M, Lee Y, Kim K (2021) Reward-oriented task offloading under limited edge server power for multi-access edge computing. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3065429
234. Li X, Chen T, Yuan D, Xu J, Liu X (2021) A novel graph-based computation offloading strategy for workflow applications in mobile edge computing, arXiv:2102.12236
235. Mukherjee A, Deb P, De D, Buyya R (2018) C2OF2N: a low power cooperative code offloading method for femtolet-based fog network. J Supercomput 74:2412–2448. https://doi.org/10.1007/s11227-018-2269-x
236. Ahn S, Gorlatova M, Chiang M (2017) Leveraging fog and cloud computing for efficient computational offloading. In: 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, pp. 1–4. https://doi.org/10.1109/URTC.2017.8284203
237. Liang K, Zhao L, Zhao X, Wang Y, Ou S (2016) Joint resource allocation and coordinated computation offloading for fog radio access networks. In Chin Commun 13(Supplement 2):131–139. https://doi.org/10.1109/CC.2016.7833467
238. Hmimz Y, Chanyour T, El Ghmary M, Malki MOC (2021) Joint radio and local resources optimization for tasks offloading with priority in a mobile edge computing network. Pervasive Mobile Comput 73:101368. https://doi.org/10.1016/j.pmcj.2021.101368
239. Hmimz Y, Chanyour T, El Ghmary M et al (2021) Bi-objective optimization for multi-task offloading in latency and radio resources constrained mobile edge computing networks. Multimed Tools Appl 80:17129–17166. https://doi.org/10.1007/s11042-020-09365-9
240. Li K (2021) Heuristic computation offloading algorithms for mobile users in fog computing. ACM Trans Embed Comput Syst 20(2):1–28. https://doi.org/10.1145/3426852
241. Wang Q, Gao A, Hu Y (2021) Joint power and QoE optimization scheme for multi-UAV assisted offloading in mobile computing. IEEE Access 9:21206–21217. https://doi.org/10.1109/ACCESS.2021.3055335

242. Cha N, Wu C, Yoshinaga T, Ji Y, Yau K-LA (2021) Virtual edge: exploring computation offloading in collaborative vehicular edge computing. IEEE Access 9:37739–37751. https://doi.org/10.1109/ACCESS.2021.3063246
243. Hao X, Zhao R, Yang T et al (2021) A risk-sensitive task offloading strategy for edge computing in industrial Internet of Things. J Wireless Com Network 2021:39. https://doi.org/10.1186/s13638-021-01923-5
244. Li SA (2021) task offloading optimization strategy in MEC-based smart cities. Internet Technol Lett 4:e158. https://doi.org/10.1002/itl2.158
245. Liang J, Li K, Liu C, Li K (2021) Joint offloading and scheduling decisions for DAG applications in mobile edge computing. Neurocomputing 424:160–171. https://doi.org/10.1016/j.neucom.2019.11.081
246. Liu C, Liu K, Ren H et al (2021) RtDS: real-time distributed strategy for multi-period task offloading in vehicular edge computing environment. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05766-5
247. Ko H, Lee J, Pack S (2018) Spatial and temporal computation offloading decision algorithm in edge cloud-enabled heterogeneous networks. IEEE Access 6:18920–18932. https://doi.org/10.1109/ACCESS.2018.2818111
248. Silva J, Marques ERB, Lopes LMB, Silva F (2021) Energy-aware adaptive offloading of soft real-time jobs in mobile edge clouds, arXiv:2102.05504
249. Razaq MM, Tak B, Peng L, Guizani M (2021) Privacy-aware collaborative task offloading in fog computing. IEEE Trans Comput Soc Syst. https://doi.org/10.1109/TCSS.2020.3047382
250. Liu Y et al (2021) Physical layer security assisted computation offloading in intelligently connected vehicle networks. IEEE Trans Wireless Commun. https://doi.org/10.1109/TWC.2021.3051772
251. Bai W et al (2021) Joint optimization of computation offloading, data compression, energy harvesting, and application scenarios in fog computing. IEEE Access 9:45462–45473. https://doi.org/10.1109/ACCESS.2021.3067702
252. Khadir AA, Seno SAH (2021) SDN-based offloading policy to reduce the delay in fog-vehicular networks. Peer-to-Peer Netw Appl 14:1261–1275. https://doi.org/10.1007/s12083-020-01066-2
253. Almutairi J, Aldossary M (2021) Modeling and analyzing offloading strategies of IoT applications over edge computing and joint clouds. Symmetry 13:402. https://doi.org/10.3390/sym13030402
254. Tang L, Tang B, Zhang L et al (2021) Joint optimization of network selection and task offloading for vehicular edge computing. J Cloud Comp 10:23. https://doi.org/10.1186/s13677-021-00240-y
255. Xue J, An Y (2021) Joint task offloading and resource allocation for multi-task multi-server NOMA-MEC networks. IEEE Access 9:16152–16163. https://doi.org/10.1109/ACCESS.2021.3049883
256. Xiaohui G, Zhang G (2021) Energy-efficient computation offloading for vehicular edge computing networks. Comput Commun 166:244–253. https://doi.org/10.1016/j.comcom.2020.12.010
257. Li R, Ma Q, Gong J, Zhou Z, Chen X (2021) Age of processing: age-driven status sampling and processing offloading for edge computing-enabled real-time IoT applications. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3064055
258. Li W, Jin S (2021) Performance evaluation and optimization of a task offloading strategy on the mobile edge computing with edge heterogeneity. J Supercomput. https://doi.org/10.1007/s11227-021-03781-w
259. Nan Y, Li W, Bao W, Delicato FC, Pires PF, Zomaya AY (2018) A dynamic tradeoff data processing framework for delay-sensitive applications in Cloud of Things systems. J Parallel Distrib Comput 112(1):53–66. https://doi.org/10.1016/j.jpdc.2017.09.009
260. Bao W, Li W, Delicato FC, Pires PF, Dong Yuan D, Zhou BB, Zomaya AY (2017) Cost-effective processing in fog-integrated internet of things ecosystems. In: Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '17). Association for Computing Machinery, New York, NY, USA, pp. 99–108. https://doi.org/10.1145/3127540.3127547
261. Mao Y, Zhang J, Letaief KB (2016) Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J Sel Areas Commun 34:3590–3605. https://doi.org/10.1109/JSAC.2016.2611964
262. Zhang G, Zhang W, Cao Y, Li D, Wang L (2018) Energy delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices. IEEE Trans Industr Inf 14:4642–4655. https://doi.org/10.1109/TII.2018.2843365

263. Lyu X, Ni W, Tian H, Liu RP, Wang X, Giannakis GB, Paulraj A (2017) Optimal schedule of mobile edge computing for internet of things using partial information. IEEE J Sel Areas Commun 35:2606–2615. https://doi.org/10.1109/JSAC.2017.2760186
264. Chen W, Wang D, Li K (2018) Multi-user multi-task computation offloading in green mobile edge cloud computing. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2018.2826544
265. Zhang D, Tan L, Ren J, Awad MK, Zhang S, Zhang Y, Wan P (2019) Near-optimal and truthful online auction for computation offloading in green edge-computing systems. IEEE Trans Mob Comput. https://doi.org/10.1109/TMC.2019.2901474
266. Pu L, Chen X, Xu J, Fu X (2016) D2d fogging: An energy efficient and incentive-aware task offloading framework via network assisted d2d collaboration. IEEE J Sel Areas Commun 34:3887–3901. https://doi.org/10.1109/JSAC.2016.2624118
267. Peng K, Nie J, Kumar N, Cai C, Kang J, Xiong Z, Zhang Y (2021) Joint optimization of service chain caching and task offloading in mobile edge computing. Appl Soft Comput 103:107142. https://doi.org/10.1016/j.asoc.2021.107142
268. Zhao F, Chen Y, Zhang Y, Liu Z, Chen X (2021) Dynamic offloading and resource scheduling for mobile edge computing with energy harvesting devices. IEEE Trans Netw and Service Manag. https://doi.org/10.1109/TNSM.2021.3069993
269. Suzhi Bi, Liang Huang, Hui Wang, and Ying-Jun Angela Zhang (2021) Stable online computation offloading via lyapunov-guided deep reinforcement learning, arXiv:2102.03286v1
270. Huynh LNT, Pham Q-V, Nguyen TDT, Hossain MD, Shin Y-R, Huh E-N (2021) Joint computational offloading and data-content caching in NOMA-MEC networks. IEEE Access 9:12943–12954. https://doi.org/10.1109/ACCESS.2021.3051278
271. Liu L, Sun B, Wu Y, Tsang DHK (2021) Latency optimization for computation offloading with hybrid NOMA–OMA transmission. IEEE Internet Things J 8(8):6677–6691. https://doi.org/10.1109/JIOT.2021.3055510
272. Lakhan A, Ahmad M, Bilal M, Jolfaei A, Mehmood RM (2021) Mobility aware blockchain enabled offloading and scheduling in vehicular fog cloud computing. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2021.3056461
273. Liu L, Chang Z, Guo X, Mao S, Ristaniemi T (2018) Multi objective optimization for computation offloading in fog computing. IEEE Internet Things J 5(1):283–294. https://doi.org/10.1109/JIOT.2017.2780236
274. Wang X, Ning Z, Wang L (2018) Offloading in internet of vehicles: a fog-enabled real-time traffic management system. IEEE Trans Industr Inf 14(10):4568–4578. https://doi.org/10.1109/TII.2018.2816590
275. Wang Y, Sheng M, Wang X, Wang L, Li J (2016) Mobile edge computing: partial computation offloading using dynamic voltage scaling. IEEE Trans Commun 64:4268–4282. https://doi.org/10.1109/TCOMM.2016.2599530
276. Chen M, Hao Y (2018) Task offloading for mobile edge computing in software defined ultra-dense network. IEEE J Sel Areas Commun 36:587–597. https://doi.org/10.1109/JSAC
277. Ji L, Guo S (2019) Energy-efficient cooperative resource allocation in wireless powered mobile edge computing. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2018.2880812
278. Hu X, Wong K, Yang K (2018) Wireless powered cooperation assisted mobile edge computing. IEEE Trans Wireless Commun 17:2375–2388. https://doi.org/10.1109/TWC.2018.2794345
279. Wang F, Xu J, Wang X, Cui S (2018) Joint offloading and computing optimization in wireless powered mobile-edge computing systems. IEEE Trans Wireless Commun 17:1784–1797. https://doi.org/10.1109/TWC.2017.2785305
280. Bi S, Zhang YJ (2018) Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading. IEEE Trans Wireless Commun 17:4177–4190. https://doi.org/10.1109/TWC.2018.2821664
281. Zhou F, Wu Y, Hu RQ, Qian Y (2018) Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems. IEEE J Sel Areas Commun 36:1927–1941. https://doi.org/10.1109/JSAC.2018.2864426
282. Dinh TQ, Tang J, La QD, Quek TQS (2017) Offloading in mobile edge computing: task allocation and computational frequency scaling. IEEE Trans Commun 65:3571–3584. https://doi.org/10.1109/TCOMM.2017.2699660
283. Qiu Y, Zhang H, Long K (2021) Computation offloading and wireless resource management for healthcare monitoring in fog-computing based internet of medical things. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3066604

284. Zhang G, Zhang S, Zhang W, Shen Z, Wang L (2021) Joint service caching, computation offloading and resource allocation in mobile edge computing systems. IEEE Trans Wireless Commun. https://doi.org/10.1109/TWC.2021.3066650
285. Liao Z, Ma Y, Huang J, Wang J, Wang J (2021) HOTSPOT: a UAV-assisted dynamic mobility-aware offloading for mobile edge computing in 3D space. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3051214
286. Muzakkir Hussain Md, Sufyan Beg MM (2021) CODE-V: multi-hop computation offloading in vehicular fog computing. Future Gener Comput Syst 116:86–102. https://doi.org/10.1016/j.future.2020.09.039
287. Youwei Yuan L, Qian GJ, Longxuan Y, Zixuan Y, Zhao Q (2021) Efficient computation offloading for service workflow of mobile applications in mobile edge computing. Mobile Info Syst 2021:1–11. https://doi.org/10.1155/2021/5578465
288. Sun Y, Song C, Yu S, Liu Y, Pan H, Zeng P (2021) Energy-efficient task offloading based on differential evolution in edge computing system with energy harvesting. IEEE Access 9:16383–16391. https://doi.org/10.1109/ACCESS.2021.3052901
289. Guan S, Boukerche A (2021) A Novel mobility-aware offloading management scheme in sustainable multi-access edge computing. IEEE Trans Sustain Comput. https://doi.org/10.1109/TSUSC.2021.3065310
290. Peng G, Wu H, Wu H, Wolter K (2021) Constrained multi-objective optimization for IoT-enabled computation offloading in collaborative edge and cloud computing. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3067732
291. Tout H, Mourad A, Kara N, Talhi C (2021) Multi-persona mobility: joint cost-effective and resource-aware mobile-edge computation offloading. IEEE/ACM Trans Netw. https://doi.org/10.1109/TNET.2021.3066558
292. Keshavarznejad M, Rezvani MH, Adabi S (2021) Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms. Cluster Comput. https://doi.org/10.1007/s10586-020-03230-y
293. Liao Z, Peng J, Xiong B et al (2021) Adaptive offloading in mobile-edge computing for ultra-dense cellular networks based on genetic algorithm. J Cloud Comp 10:15. https://doi.org/10.1186/s13677-021-00232-y
294. Deng X, Sun Z, Li D, Luo J, Wan S (2021) User-centric computation offloading for edge computing. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3057694
295. Russo Russo G, Nardelli M, Cardellini V, Lo Presti F (2018) Multi-level elasticity for wide-area data streaming systems: a reinforcement learning approach. Algorithms 11:134. https://doi.org/10.3390/a11090134
296. Dehnavi S, Faragardi HR, Kargahi M, Fahringer T (2019) A reliability-aware resource provisioning scheme for real-time industrial applications in a Fog-integrated smart factory. Microproc Microsyst 70:1–14. https://doi.org/10.1016/j.micpro.2019.05.011
297. Mohammad FM, Jabbehdari S, Javadi HHS (2020) A dynamic fog service provisioning approach for IoT applications. Int J Commun Syst 33(14):e4541
298. Shahidinejad A, Ghobaei-Arani M (2020) Joint computation offloading and resource provisioning for edge-cloud computing environment: a machine learning-based approach. Softw: Pract Exper 50: 2212– 2230. https://doi.org/10.1002/spe.2888
299. Wang N, Matthaiou M, Nikolopoulos DS, Varghese B (2020) DYVERSE: dynamic vertical scaling in multi-tenant edge environments. Future Gener Comput Syst 108:598–612. https://doi.org/10.1016/j.future.2020.02.043
300. Tseng F, Tsai M, Tseng C, Yang Y, Liu C, Chou L (2018) A lightweight autoscaling mechanism for fog computing in industrial applications. IEEE Trans on Indus Info 14(10):4529–4537. https://doi.org/10.1109/TII.2018.2799230
301. Mulinti RB, Nagendra M (2021) An efficient latency aware resource provisioning in cloud assisted mobile edge framework. Peer-to-Peer Netw Appl 14:1044–1057. https://doi.org/10.1007/s12083-020-01070-6
302. Wang N, Varghese B, Matthaiou M, Nikolopoulos DS (2017) ENORM: a framework for edge node resource management. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2017.2753775
303. Baghban H, Huang C-Y, Hsu C-H (2020) Resource provisioning towards OPEX optimization in horizontal edge federation. Comput Commun 158:39–50. https://doi.org/10.1016/j.comcom.2020.04.009

304. Vinueza Naranjo PG, Baccarelli E, Scarpiniti M (2018) Design and energy-efficient resource management of virtualized networked Fog architectures for the real-time support of IoT applications. J Supercomput 74:2470–2507. https://doi.org/10.1007/s11227-018-2274-0
305. Naha RK, Garg S, Chan A, Battula SK (2020) Deadline-based dynamic resource allocation and provisioning algorithms in fog-cloud environment. Future Gener Comput Syst 104:131–141. https://doi.org/10.1016/j.future.2019.10.018
306. Son J, Buyya R (2019) Latency-aware virtualized network function provisioning for distributed edge clouds. J Syst Softw 152:24–31. https://doi.org/10.1016/j.jss.2019.02.030
307. El Kafhali S, Salah K (2017) Efficient and dynamic scaling of fog nodes for IoT devices. J Supercomput 73:5261–5284. https://doi.org/10.1007/s11227-017-2083-x
308. Arkian HR, Diyanat A, Pourkhalili A (2017) MIST: fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications. J Netw Comput Appl 82:152–165. https://doi.org/10.1016/j.jnca.2017.01.012
309. Santos J, Wauters T, Volckaert B, De Turck F (2021) Towards end-to-end resource provisioning in fog computing over low power wide area networks. J Netw Comput Appl 175:102915. https://doi.org/10.1016/j.jnca.2020.102915
310. Ma X, Wang S, Zhang S, Yang P, Lin C, Shen XS (2019) Cost-efficient resource provisioning for dynamic requests in cloud assisted mobile edge computing. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2019.2903240
311. Zhou Z, Yu S, Chen W, Chen X (2020) CE-IoT: cost-effective cloud-edge resource provisioning for heterogeneous IoT applications. IEEE Internet Things J 7(9):8600–8614. https://doi.org/10.1109/JIOT.2020.2994308
312. Abouaomar A, Cherkaoui S, Mlika Z, Kobbane A (2021) Resource provisioning in edge computing for latency sensitive applications. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2021.3052082
313. Madan N, Malik AW, Rahman AU, Ravana SD (2020) On-demand resource provisioning for vehicular networks using flying fog. Veh Commun 25:100252. https://doi.org/10.1016/j.vehcom.2020.100252
314. Etemadi M, Ghobaei-Arani M, Shahidinejad A (2020) Resource provisioning for IoT services in the fog computing environment. An autonomic approach. Comput Commun 161:109–131. https://doi.org/10.1016/j.comcom.2020.07.028
315. Li C, Bai J, Luo Y (2020) Efficient resource scaling based on load fluctuation in edge-cloud computing environment. J Supercomput 76:6994–7025. https://doi.org/10.1007/s11227-019-03134-8
316. Murturi I, Avasalcai C, Tsigkanos C, Dustdar S (2019) edge-to-edge resource discovery using metadata replication. In: 2019 IEEE 3rd International Conference on Fog and Edge Computing (ICFEC), Larnaca, Cyprus, pp.1–6. https://doi.org/10.1109/CFEC.2019.8733149
317. Babou CSM, Fall D, Kashihara S, Taenaka Y, Bhuyan MH, Niang I, Kadobayashi Y (2020) Hierarchical load balancing and clustering technique for home edge computing. IEEE Access 8:127593–127607. https://doi.org/10.1109/ACCESS.2020.3007944
318. Gedeon J, Meurisch C, Bhat D, Stein M, Wang L, Mühlhäuser M (2017) Router-based brokering for surrogate discovery in edge computing. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, pp. 145–150. https://doi.org/10.1109/ICDCSW.2017.61
319. Rejiba Z, Masip-Bruin X, Jurnet A, Marin-Tordera E, Ren G (2018) F2C-aware: enabling discovery in wi-fi-powered fog-to-cloud (F2C) systems. In: 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), Bamberg, pp. 113–116. https://doi.org/10.1109/MobileCloud.2018.00025
320. Tanganelli G, Vallati C, Mingozzi E (2018) Edge-centric distributed discovery and access in the internet of things. IEEE Internet Things J 5(1):425–438. https://doi.org/10.1109/JIOT.2017.2767381
321. Saurez E, Hong K, Lillethun D, Ramachandran U, Ottenwälder B (2016) Incremental deployment and migration of geo-distributed situation awareness applications in the fog. In: Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems (DEBS ’16). Association for Computing Machinery, New York, NY, USA, 258–269. https://doi.org/10.1145/2933267.2933317
322. Fayos-Jordan R, Felici-Castell S, Segura-Garcia J, Lopez-Ballester J, Cobos M (2020) Performance comparison of container orchestration platforms with low cost devices in the fog, assisting

Internet of Things applications. J Netw Comput Appl 169:102788. https://doi.org/10.1016/j.jnca.2020.102788
323. Okwuibe J, Haavisto J, Harjula E, Ahmad I, Ylianttila M (2020) SDN enhanced resource orchestration of containerized edge applications for industrial IoT. IEEE Access 8:229117–229131. https://doi.org/10.1109/ACCESS.2020.3045563
324. Slamnik-Kriještorac N, de Britto e Silva E, Municio E, Carvalho de Resende HC, Hadiwardoyo SA, Marquez-Barja JM (2020) Network service and resource orchestration: a feature and performance analysis within the MEC-enhanced vehicular network context. Sensors 20: 3852. https://doi.org/10.3390/s20143852
325. Yuan Q et al (2020) Cross-domain resource orchestration for the edge-computing-enabled smart road. IEEE Network 34(5):60–67. https://doi.org/10.1109/MNET.011.2000007
326. Petri I, Rana O, Bittencourt LF, Balouek-Thomert D, Parashar M (2020) Autonomics at the edge: resource orchestration for edge native applications. IEEE Internet Comput. https://doi.org/10.1109/MIC.2020.3039551
327. Sonmez C, Ozgovde A, Ersoy C (2019) Fuzzy workload orchestration for edge computing. IEEE Trans Netw Serv Manage 16(2):769–782. https://doi.org/10.1109/TNSM.2019.2901346
328. Vu D-N, Dao N-N, Na W, Cho S (2020) Dynamic resource orchestration for service capability maximization in fog-enabled connected vehicle networks. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2020.3001289
329. Liu Q, Han T (2019) VirtualEdge: multi-domain resource orchestration and virtualization in cellular edge computing. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, pp. 1051–1060. https://doi.org/10.1109/ICDCS.2019.00108
330. Babirye S, Serugunda J, Okello D, Mwanje S (2020) Resource-aware workload orchestration for edge computing. In: 2020 28th Telecommunications Forum (TELFOR), pp. 1–4. https://doi.org/10.1109/TELFOR51502.2020.9306551
331. Liu Q, Han T (2019) DIRECT: distributed cross-domain resource orchestration in cellular edge computing. In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '19). Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/3323679.3326516
332. AlEbrahim S, Ahmad I (2017) Task scheduling for heterogeneous computing systems. J Supercomput 73:2313–2338. https://doi.org/10.1007/s11227-016-1917-2
333. Sharma Y, Das Z, Moulik S (2021) SPORTS: a semi-partitioned real-time scheduler for heterogeneous multicore platforms. In: Ning L., Chau V., Lau F. (eds) Parallel Architectures, Algorithms and Programming. PAAP 2020. Communications in Computer and Information Science, vol 1362. Springer, Singapore. https://doi.org/10.1007/978-981-16-0010-4_35
334. Moulik S, Devaraj R, Sarkar A COST: a cluster-oriented scheduling technique for heterogeneous multi-cores. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2018, pp. 1951–1957. https://doi.org/10.1109/SMC.2018.00337
335. Dastjerdi AV, Gupta H, Calheiros RN, Ghosh SK, Buyya R. 2016. Fog computing: principles, architectures, and applications. In: Internet of Things. Elsevier, pp. 61–75
336. de Assunção MD, da Silva Veith A, Buyya R (2018) Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. J Netw Comput Appl 103: 1–17. https://doi.org/10.1016/j.jnca.2017.12.001
337. Chirivella-Perez E, Alcaraz JM, Calero QW, Gutiérrez-Aguado J (2018) Orchestration architecture for automatic deployment of 5G services from bare metal in mobile edge computing infrastructure. Wireless Commun Mobile Comput 2018:18. https://doi.org/10.1155/2018/5786936
338. Bellavista P, Zanni A. 2017 Feasibility of fog computing deployment based on docker containerization over raspberry Pi. In: Proceedings of the 18th International Conference on Distributed Computing and Networking. ACM, 16.
339. Liu P, Willis D, Banerjee S. 2016 Paradrop: enabling lightweight multi-tenancy at the network's extreme edge. In: 2016 IEEE/ACM Symposium on Edge Computing (SEC). IEEE, pp. 1–13
340. Cardellini V, Lo Presti F, Nardelli M, Rossi F (2020) Self-adaptive container deployment in the fog: a survey. In: Brandic I., Genez T., Pietri I., Sakellariou R. (eds) algorithmic aspects of cloud computing. ALGOCLOUD 2019. lecture notes in computer science, vol 12041. Springer, Cham. https://doi.org/10.1007/978-3-030-58628-7_6
341. Shekhar S, Chhokra AD, Bhattacharjee A, Aupy G, Gokhale A. 2017. INDICES: exploiting edge resources for performance-aware cloud hosted services. In: IEEE 1st International Conference on Fog and Edge Computing. IEEE, pp. 75–80

342. Bittencourt LF, Lopes MM, Petri I, Rana OF 2015. Towards virtual machine migration in fog computing. In: 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. IEEE, pp. 1–8
343. Hong C-H, Spence I, Nikolopoulos DS (2017) FairGV: fair and fast GPU virtualization. IEEE Trans Parallel Distrib Syst 28(12):3472–3485
344. Ha K, Abe Y, Eiszler T, Chen Z, Hu W, Amos B, Upadhyaya R, Pillai P, Satyanarayanan M you can teach elephants to dance: agile VM handoff for edge computing. In: Proceedings of the ACM/IEEE 2nd Symposium on Edge Computing (SEC), San Jose, CA, USA, 28 July 2017; pp. 1–14
345. He J, Ji S, Pan Y, Li Y (2014) Constructing load-balanced data aggregation trees in probabilistic wireless sensor networks. IEEE Trans Parallel Distrib Syst 25(7):1681–1690
346. Sefraoui O, Aissaoui M, Eleuldj M (2012) OpenStack: toward an open-source solution for cloud computing. Int J Comput Appl 55:3
347. Brewer EA. 2015. Kubernetes and the path to cloud native. In: Proceedings of the Sixth ACM Symposium on Cloud Computing (SoCC ’15). Association for Computing Machinery, New York, NY, USA, 167. https://doi.org/10.1145/2806777.2809955
348. Mahmud R, Ramamohanarao K, Buyya R (2020) Application management in fog computing environments: a taxonomy, review and future directions. ACM Comput Surv 1(1):36. https://doi.org/10.1145/nnnnnnn.nnnnnnn
349. Moulik S, Das Z, Saikia G CEAT: a cluster based energy aware scheduler for real-time heterogeneous systems. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 1815–1821. https://doi.org/10.1109/SMC42975.2020.9283084
350. Zhou J et al (2020) Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT. IEEE Trans Serv Comput 13(4):745–758. https://doi.org/10.1109/TSC.2019.2963301
351. Hua Y, Guan L, Kyriakopoulos KG (2020) A Fog caching scheme enabled by ICN for IoT environments. Future Gener Comput Syst 111:82–95. https://doi.org/10.1016/j.future.2020.04.040
352. Afrin M, Jin J, Rahman A, Tian Y-C, Kulkarni A (2019) Multi-objective resource allocation for edge cloud based robotic workflow in smart factory. Futur Gener Comput Syst 97(2019):119–130
353. Gupta H, Dastjerdi AV, Ghosh SK, Buyya R 2017. iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Softw: Pract Exp 47(9): 1275–1296
354. Sonmez C, Ozgovde A, Ersoy C (2018) EdgeCloudSim: an environment for performance evaluation of edge computing systems. Trans Emerg Telecommun Technol 29(11):e3493
355. Schulz P, Matthe M, Klessig H, Simsek M, Fettweis G, Ansari J, Ashraf SA, Almeroth B, Voigt J, Riedel I, Puschmann A, Mitschele-Thiel A, Muller M, Elste T, Windisch M (2017) Latency critical IoT applications in 5G: perspective on the design of radio interface and network architecture. IEEE Commun Mag 55(2):70–78
356. Bellavista P, Chessa S, Foschini L, Gioia L, Girolami M (2018) Human-enabled edge computing: exploiting the crowd as a dynamic extension of mobile edge computing. IEEE Commun Mag 56(1):145–155
357. Kang Y, Hauswald J, Gao C, Rovinski A, Mudge T, Mars J, Tang L (2017) Neurosurgeon: collaborative intelligence between the cloud and mobile edge. Acm Sigplan Notices 52(4):615–629
358. Teerapittayanon S, McDanel B, Kung HT 2017. Distributed deep neural networks over the cloud, the edge and end devices. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 328–339
359. Xia Q, Ye W, Tao Z, Jindi W, Li Q (2021) A survey of federated learning for edge computing: research problems and solutions. High-Confid Comput 1(1):100008. https://doi.org/10.1016/j.hcc.2021.100008
360. Sanaei Z, Abolfazli S, Gani A, Buyya R (2014) Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun Surv Tut 16(1):369–392
361. Xiao Y, Noreikis M, Yl¨a-Ja¨ aiski A 2017. Qos-oriented capacity planning for edge computing In: 2017 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6
362. Varghese B, Wang N, Li J, Nikolopoulos DS 2017a. Edge-as-a-Service: towards distributed cloud architectures. In International Conference on Parallel Computing (Advances in Parallel Computing). IOS Press, pp. 784–793
363. Fernando N, Loke SW, Rahayu W 2016. Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds. IEEE Transactions on Cloud Computing (2016)

364. Tang B, Chen Z, Hefferman G, Wei T, He H, Qing Yang Q. 2015. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In: Proceedings of the ASE BigData & Social Informatics. ACM, 28
365. Zhou Z, Chen X, Li E, Zeng L, Luo K, Zhang J (2019) Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc IEEE. https://doi.org/10.1109/JPROC.2019.2918951
366. Xiao Y, Jia Y, Liu C, Cheng X, Yu J, Lv W (2019) Edge computing security: state of the art and challenges. Proc IEEE 107:1608–1631
367. Hong C-H, Varghese B (2019) Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. ACM Comput Surv 52(5):1–37. https://doi.org/10.1145/332606
368. Ghobaei-Arani M, Souri A, Rahmanian AA (2020) Resource management approaches in fog computing: a comprehensive review. J Grid Comput 18:1–42. https://doi.org/10.1007/s10723-019-09491-1
369. Puliafito C, Mingozzi E, Longo F, Puliafito A, Rana O (2019) Fog computing for the internet of things: a Survey. ACM Trans Internet Technol (TOIT) 19(2):18
370. Toczé K, Nadjm-Tehrani S (2018) A taxonomy for management and optimization of multiple resources in edge computing. Wireless Commun Mobile Comput 2018:1–23. https://doi.org/10.1155/2018/7476201
371. Luo Q, Shihong H, Li C, Li G, Shi W (2021) Resource scheduling in edge computing: a survey. IEEE Commun Surv Tut 23(4):2131–2165
372. Salaht FA, Desprez F, Lebre A (2020) An overview of service placement problem in fog and edge computing. ACM Comput 53(3):1–35. https://doi.org/10.1145/3391196
373. Nayeri ZM, Ghafarian T, Javadi B (2021) Application placement in Fog computing with AI approach: taxonomy and a state of the art survey. J Netw Comput Appl 185:103078. https://doi.org/10.1016/j.jnca.2021
374. Kaur M, Aron R (2021) A systematic study of load balancing approaches in the fog computing environment. J Supercomput 77:9202–9247. https://doi.org/10.1007/s11227-020-03600-8
375. Neghabi N, Hosseinzadeh R (2018) Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. IEEE Access 6:14159–14178. https://doi.org/10.1109/ACCESS.2018.2805842
376. Lin H, Zeadally S, Chen Z, Labiod H, Wang L (2020) A survey on computation offloading modeling for edge computing. J Netw Comput Appl 169:102781. https://doi.org/10.1016/j.jnca.2020.102781
377. Wang J, Pan J, Esposito F, Calyam P, Yang Z, Mohapatra P (2019) Edge cloud offloading algorithms: issues, methods, and perspectives. ACM Comput Surv (CSUR) 52(1):1–23
378. Aazam M, Zeadally S, Harras KA (2018) Offloading in fog computing for IoT: review, enabling technologies, and research opportunities. Future Gener Comput Syst 87:278–289. https://doi.org/10.1016/j.future.2018.04.057
379. Jiang C, Cheng X, Gao H, Zhou X, Wan J (2019) Toward computation offloading in edge computing: a survey. IEEE Access 7:131543–131558. https://doi.org/10.1109/ACCESS.2019.2938660
380. Shakarami A, Ghobaei-Arani M, Masdari M et al (2020) A survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective. J Grid Comput. https://doi.org/10.1007/s10723-020-09530-2
381. Le Duc T, Leiva RG, Casari P, Östberg P-O (2019) Machine learning methods for reliable resource provisioning in edge-cloud computing: a survey. ACM Comput Surv 52(5):1–39. https://doi.org/10.1145/3341145
382. Velasquez K, Abreu DP, Assis M et al (2018) (2018) Fog orchestration for the Internet of Everything: state-of-the-art and research challenges. J Internet Serv Appl 9:14. https://doi.org/10.1186/s13174-018-0086-3
383. Casalicchio E (2019) Container orchestration: A survey. In: Puliafito A., Trivedi K. (eds) Systems Modeling: Methodologies and Tools. EAI/Springer innovations in communication and computing. Springer, Cham. https://doi.org/10.1007/978-3-319-92378-9_14
384. Ren J, Zhang D, He S, Zhang Y, Li T (2019) A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet. ACM Comput Surv 52(6):1–36. https://doi.org/10.1145/3362031

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.