Karbala International Journal of Modern Science

Manuscript 3360

Heterogeneous Resources in Infrastructures of the Edge Network Paradigm: A Comprehensive Review

Qusay S. Alsaffar

Leila Ben Ayed

Heterogeneous Resources in Infrastructures of the Edge Network Paradigm: A Comprehensive Review

Abstract

The late 1990s saw the rise of the edge computing network paradigm, as well as an increase in the number of IoT de-vices. This concept is viewed as a link between cloud servers and end-devices, bringing processing and storage re-sources closer to clients. As a result of its low latency and high performance, researchers and developers have expressed interest in it. However, this paradigm confronts a number of obstacles and restrictions, including restricted and hetero-geneous resources at network edges. In this paper, we provide a detailed review of heterogeneous resources in edge network infrastructures using a three-dimensional method. These three dimensions in this concept correspond to the edge computer layers, hardware layers, and software layers of the edge network paradigm infrastructure ecosystem. This review considers Artificial Intelligence (AI), which classifies cutting-edge works into two categories: AI-based and non-AI-based solutions based on research issues such as fault tolerance, power consumption, resource utilization, re-source allocation, latency, device ID, clustering, heuristic-based, and meta-heuristic-based. Because reviews in this field rarely address full research concerns linked to this research topic. This review provides a sufficient overview to address the majority of open research questions. We examine and compare outstanding issues in AI-based and non-AI-based systems, focusing on evaluation metrics for meeting Quality of Services (QoS) and Quality of Experience (QoE) stand-ards. We expect that this evaluation will assist developers and researchers in determining the appropriate solution from research issues to achieve their objectives in building IoT technology and edge computing networks.

Keywords

Edge Computing; Fog Computing; Cloudlet; Heterogeneous resources, Federated Learning, Data Center

REVIEW ARTICLE

Heterogeneous Resources in Infrastructures of the Edge Network Paradigm: A Comprehensive Review

Qusay S. Alsaffar [a,*], Leila B. Ayed [b]

[a] National School of Electronics and Telecommunications of Sfax, University of Sfax, Tunisia
[b] National School of Computer Science Tunisia, Tunisia

Abstract

The late 1990s saw the rise of the edge computing network paradigm, as well as an increase in the number of IoT devices. This concept is viewed as a link between cloud servers and end-devices, bringing processing and storage resources closer to clients. As a result of its low latency and high performance, researchers and developers have expressed interest in it. However, this paradigm confronts a number of obstacles and restrictions, including restricted and heterogeneous resources at network edges. In this paper, we provide a detailed review of heterogeneous resources in edge network infrastructures using a three-dimensional method. These three dimensions in this concept correspond to the edge computer layers, hardware layers, and software layers of the edge network paradigm infrastructure ecosystem. This review considers Artificial Intelligence (AI), which classifies cutting-edge works into two categories: AI-based and non-AI-based solutions based on research issues such as fault tolerance, power consumption, resource utilization, resource allocation, latency, device ID, clustering, heuristic-based, and meta-heuristic-based. Because reviews in this field rarely address full research concerns linked to this research topic. This review provides a sufficient overview to address the majority of open research questions. We examine and compare outstanding issues in AI-based and non-AI-based systems, focusing on evaluation metrics for meeting Quality of Services (QoS) and Quality of Experience (QoE) standards. We expect that this evaluation will assist developers and researchers in determining the appropriate solution from research issues to achieve their objectives in building IoT technology and edge computing networks.

Keywords: Edge computing, Fog computing, Cloudlet, Heterogeneous resources, Federated learning, Data center

1. Introduction

Most Internet of Things (IoT) gadgets and applications now access remote resources through cloud platforms provided by service providers. Typically, data generated by end machines such as laptops, mobile phones, sensors, and automobiles is transferred to the resources of a remote cloud for computation. This approach is deemed wasteful and unsatisfactory due to increased latencies when IoT units and apps are spread on a wide scale [1]. Edge computing is an emerging geo-distributed computational model that has currently piqued the interest of the research community by bringing cloud computing capabilities closer to users and data sources to meet the services and requirements of end users [2,3].

Edge computing (EC) encompasses a variety of paradigms, including fog computing (FC), mobile edge computing (MEC), and local cloud computing (cloudlet). These paradigms are implemented to reduce latency while meeting massive processing demands at the network's edge. To increase application capacity, as is always assumed in theoretical research, the edge's resources should be combined into a consistent resource grouping and diverse edge devices should collaborate. However, this cooperative is difficult to implement since heterogeneous organizations lead to siloed facilities [4].

Received 27 December 2023; revised 9 May 2024; accepted 14 May 2024.
Available online 13 June 2024

* Corresponding author.
E-mail address: qusayssaffar@gmail.com (Q.S. Alsaffar).
Peer review under responsibility of University of Kerbala.

https://doi.org/10.33640/2405-609X.3360

Heterogeneity refers to the existence of various types of hardware and software, most likely with vastly different storage spaces, computing power, and fundamental architecture; the substitution of a data source with a variety of structures, controls, and applications; and the interaction paradigm is limited to hardware and software itself [5]. Heterogeneous machines are another important feature of any Internet of Things environment [6]. This environment is designed to deal with varying levels of machine heterogeneity, which implies differences in not only protocols or machine operations, but also computational resources [7].

This study aims to conduct a complete assessment and categorization of cutting-edge research on heterogeneous infrastructure resources in the edge computing paradigm. We begin by providing an overview of the methodologies and tactics used to address the consequences caused by heterogeneity in edge computing networks. The three-dimensional scheme is then presented: the first dimension represents the ecosystem's edge layer components, the second dimension represents the hardware ecosystem's components, and the third dimension represents the software ecosystem's components. Furthermore, we classified the most recent research in the field of heterogeneous resources into two categories: AI-based and non-AI-based solutions, based on the most important research issues such as fault tolerance, power consumption, resource utilization, resource allocation, latency, device ID, clustering, heuristic-based, and meta-heuristic-based. The main contributions of this study are as follows:

1 Offering an outline of the important issues of heterogeneous resources of infrastructure in edge computing.
2 Offering a review of related surveys.
3 Classifying the components of the infrastructure in the ecosystem for edge computing according to the heterogeneous facilities into a three-dimensional scheme.
4 Conducting a taxonomy of the research issues with AI and without AI.
5 Offering the research issues and future works that methods and techniques can be applied in edge computing in order to improve performance.

The remaining parts of the research are structured as follows: In Section 2, we investigate diverse methods and techniques that study heterogeneous resource issues according to their goals. Section 3 addresses the scheme for layers and components of infrastructure in three dimensions. Section 4 provides methods and techniques for heterogeneous resources. Section 5 gives a discussion and analysis of the heterogeneous resource issues. Finally, Section 6 concludes the major outcomes and proposes future work.

2. Related works

Edge computing is defined as the investment in devices and applications located at the network edge [8,9]. While fog computing is defined as an investment in both edge resources and the cloud [10,11], local cloud (cloudlet) is a notion that expresses a small geographic area as a set of servers. This local cloud is used to provide services that are placed near the network edge. The local cloud aims to reduce response time and traffic in the global cloud by bringing computing capabilities closer to clients [4]. Hong et al. [12] conducted a review to distinguish and categorize the sectors that address the implementation fog/edge paradigm's resource management requirements. There are numerous challenges in different levels of heterogeneous settings, with several of them focusing on features such as latency, resource investment, fault tolerance, acceleration, and real-time communications [13]. The network edge architecture is introduced in several cooperative resource management methods, providing a unifying paradigm for two types of operations: centralized and decentralized [14].

The resource management strategies were investigated using an infrastructure of edge computing and cloud computing. This infrastructure is taken in the style of a classic category to determine the most recent approaches to the issue of infrastructure. The most essential research subjects covered include shorter reaction times and faster communication between end users, as well as some metrics and assessment tools [15]. The resource management approaches are based on classification and subsequent deconstruction in order to identify difficulties in the current state of the art. The study focuses on non-functional characteristics [16].

Liu et al. [17] conducted a survey of several architectures for dynamic edge devices and settings. Mach and Becvar [18] presented a study of approaches for computing in the vicinity of users in the field of dynamic edge computing. Mouradian et al. [19] created a taxonomy of architectures based on whether they are application-accessible or not, and they also considered numerous criteria, the most important of which was heterogeneity.

Fahimullah et al. [20] focused their study on machine learning's contribution to many fog computing research concerns. Goudarzi et al. [21]

used reinforcement learning and deep neural networks based on their experience with application placement. The Directed Acyclic Graphs approach was then used to describe applications with changing job numbers and paradigms. However, they built the nonrealistic Directed Acyclic Graphs dataset with varying priorities to simulate the scenarios of the Internet of Things causing heterogeneity of Directed Acyclic Graphs. The resources of the end user, fog, and cloud are a collection of decentralized heterogeneous things in fog settings. Given the fog environment's inherent heterogeneity and dynamic nature, resource assignment is difficult and susceptible to NP difficulties [22]. When assigning resources to new users, some characteristics of the fog computing network must be considered, including as heterogeneous applications, a random workload environment, and dynamic [23]. Shi and his colleagues [24] used deep reinforcement learning techniques to maximize utility while minimizing response time in a heterogeneous vehicular fog network.

The proposed approach was based on the method of horizontal collaboration in fog layers, fractional computational offloading, and a centralized learning technique in the data sink. Sarkar and Kumar [25] used a deep reinforcement learning technique in a heterogeneous fog environment that incorporates automated judgments via multiple parallel deep neural networks for offloading computing. These decisions are then applied to training and testing operations.

The authors of [26] gave a recap of the primary technologies used to construct a network edge platform, as well as a framework for a network edge platform based on decentralized storage. The edge computing environment uses virtualization technology that is based on a representation method. This method works by assembling heterogeneous machines such as switches, routers, gateways, storage, and servers into an on-premises network. The next phase is virtualization, which involves converting to a virtual machine and then deploying the specific application on it.

This technology has radically altered the concept of network architecture devices, allowing for the installation of various service applications on a virtualized system. Indeed, edge computing can be defined as a scene in which a large number of heterogeneous nodes interface and collaborate to handle user requests for computation and storage. As a result, in this research, we focused on developing a new scheme in three-dimensional architecture for edge computing that provides more insight and knowledge of the edge network ecosystem. In addition, whether or not AI is used, the most essential research challenges associated to homogeneous resources will be classified. “Table 1" lists the acronyms that will be used often throughout this review.

3. Infrastructure

The infrastructure for the network edge paradigm provides partitions that include layers of edge computing, hardware, and software to initialize all capabilities for applications that use this paradigm [27]. “Fig. 1” depicts the scheme of infrastructures in the edge computing ecosystem, illustrating various infrastructure entities, which are divided into three layers: edge computing, hardware, and software. The infrastructure layers for edge computing are divided into three tiers: end-user, edge, and cloud.

Hardware infrastructure consists of three tiers: device, node, and server. Software infrastructure consists of three tiers: data, control, and application. The infrastructure taxonomy in the ecosystem, which includes hardware and software. The hardware side's device layer includes Raspberry Pi, desktops, laptops, smartphones, and IoT devices. The node tier includes switches, gateways, routers, and WiFi access points, whereas the server tier

Table 1. Outline of acronyms frequently applied in this review.

Acronym	Definition	Acronym	Definition
AI	Artificial Intelligence	FC	Fog Computing
APs	Access Points	IoT	Internet of Thing
CDN	Content Delivery Networks	GENI	Global Environment for Network Innovations
CNN	Convolutional Neural Network	MEC	Mobile Edge Computing
CPU	Central Processing Unit	QoS	Quality of Services
DC	Data Center	QoE	Quality of Experience
DL	Deep Learning	PC	Personal Computer
ECDs	Edge Computing Devices	RAMs	Random Access Memories
EC	Edge Computing	SBC	Single-board Computer
FiWi	Fiber-Wireless	OS	Operating System
FL	Federated Learning	VM	Virtual Machine

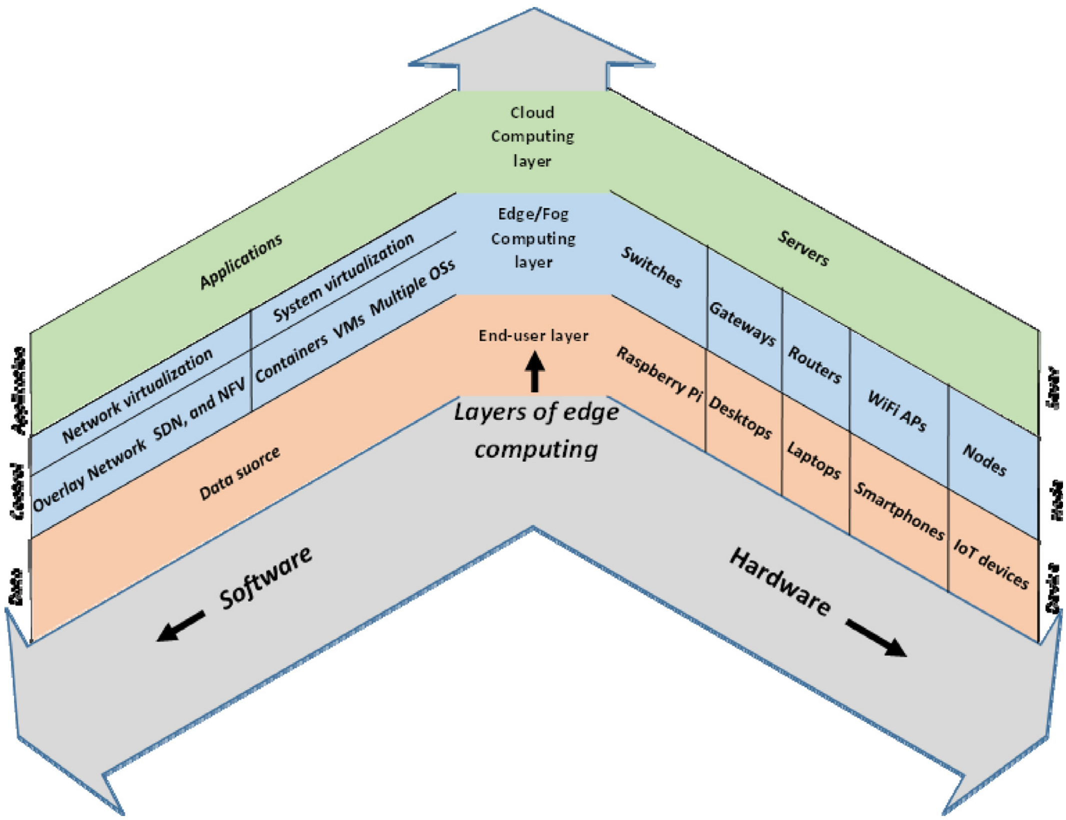

Fig. 1. Represents the scheme of infrastructures at the edge computing ecosystem in a three-dimensional.

includes servers. The data layer on the software side contains data sources, whereas the control tier encompasses system virtualization (VMs and containers) and network virtualization. The application tier contains application-related services. The previous layers correspond to end-devices, fog/edge computing, and cloud computing, respectively.

3.1. Layers of edge computing

Edge computing originated in the late 1990s, when Akamai introduced content delivery networks (CDN) to improve online efficiency. Edge computing is a more advanced version of CDN that makes use of global cloud infrastructure [9]. The main purpose of edge computing is to have a layer that sits between the cloud computing layer and the end device layer to perform computing. As a result, the edge computing ecosystem is divided into three layers: the cloud, the end device, and the edge layer.

3.1.1. Cloud layer

Cloud computing is a type of computer resource that is accessible to end devices over a network. A cloud is a data center that provides processing and caching services to end devices via the network. Cloud computing frequently distributes services across numerous positions, each of which is a DC. This technique decreases traffic, lowering application response times [28].

3.1.2. Device layer

A significant collection of devices that create large amounts of data can be found near the network's edges.

These devices swap. They transmit data via a communication network and monitor and operate the infrastructure. End devices at the edge typically use Internet of Things machines to connect to the network [29].

3.1.3. Edge layer

Edge computing devices (ECDs) are distributed computing resources that can be found between the cloud and end devices. These devices are scattered and can be placed in a variety of settings. ECDs characterize device heterogeneity, allowing communication between different protocol tires and non-IP-based technologies [30].

3.2. Hardware

The edge of the network consists of reachable IoT devices that activate cloud computing components such as servers, routers, switches, gateways, base stations, and other edge nodes [2,31,32]. Currently, these resources are embedded with Raspberry Pi, which has tremendous computational power. Edge computing makes use of items such as PC computers, portable devices, and mobile phones [12]. Hardware used at the network edge can be divided into three types: computing resources, network resources, and traditional data centers.

3.2.1. Computation resources

Computation resources for the local cloud include single-board computers and commodity items designed to handle local cloud data [12].

1. Single-board computers (SBCs), such as the Raspberry Pi, are frequently utilized as local cloud devices [33−35]. A single-board computer is a complete computer built on a single circuit board that includes a central processing unit, cache, and networking. The microdevice lacks slots for terminal equipment. FocusStack [34] creates a cloud environment by connecting many Raspberry Pi boards to linked cars and drones. FocusStack created a video sharing service in which cameras in vehicles collect moving views and address and deploy them using Raspberry Pi boards. Bel-lavista et al. [35] introduced a single board computer for Internet of Things gateways that is located near sensors, allowing for accurate data collecting. Hong et al. [36] employed single-board computers to provide crowd-sourced local cloud computing and flexible Internet of Things analytics.
2. Commodity Products: PC computers, portables, and mobile phones have all been employed as local cloud devices. For example, a recent study aimed to build a global cloud computing layer employing the aforementioned devices at educational institutions as well as public places. Because the owners of these resources may not always fully utilize their processing capabilities, local cloud computing companies may invest in resources to rent idle equipment to other consumers [12].

3.2.2. Network resources

Local cloud computing network resources include end-of-network switches, routers, gateways, WiFi APs, and edge racks [12].

1. Switches, Routers, and Gateways: Local cloud computing capability resources include network switches, routers, and gateways, which prepare a data route between end users and service providers. Aazam et al. proposed a shared gateway to assess if data collected from Internet of Things resources should be sent to data center clouds [37].
2. WiFi APs: ParaDrop [38], a local cloud computing framework, assumes that WiFi APs or other wireless gateways are ubiquitous and always turned on.
3. Edge Racks: The Global Environment for Network Innovations (GENI) integrates network, compute, and storage components onto a single rack [39]. GENI provides a local cloud computing layer by publishing GENI racks across many networked locations.

3.2.3. Classical data center

Hardware is a physical reality that is critical for constructing data center networks. Recently, the rise of large-scale cloud services has been driven by the efficiency of hardware in data center networks. The gear used in data centers is characterized as follows: switch, router, gateway, server, storage, rack, and cable, all of which are coupled to construct network topologies. The traditional key topologies of data center networks can be classified as multi-stage or multi-rooted [40]. "Fig. 2" depicts the topology of a multi-stage data center network (16 servers).

3.3. Software

The software operates directly on network edge devices such as CPUs, storage, cache, and network components. It manages devices and routes them to edge computing applications. As an example of software, consider virtualization [12]. Middleware operates on an operating system and provides supplemental duties that are not enhanced by virtualization software. The middleware integrates deployed compute devices and works to distribute instances (VMs) or containers to each network edge device [12].

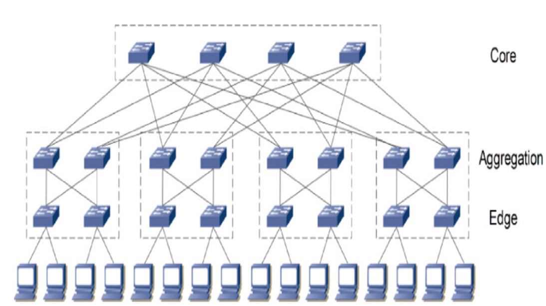

Fig. 2. A multi-stage data center network topology [40].

The network edge describes a computer layer that uses low-energy resources like routers and gateways. These resources now have computational power and are connected to a network. The combination of these modest resources creates a cloud computing layer that may be used by an existing set of Internet of Things (IoT) services [12].

Edge computing requires software to handle multitenancy since it includes several applications delivered by various tenants. System virtualization allows many operating systems to run simultaneously on the same server. It enables performance and fault isolation between groups of users in local cloud computing. System virtualization divides resources across users, preventing one user from exploiting the resources of another. As a result, a customer's mistake has no effect on other customers. System virtualization manages each user's resource use by evenly distributing resources across numerous users. A virtual machine (VM) is a collection of virtualized devices used to simulate a physical component. Virtualized devices include processors, RAMs, wireless NCs, Hard Disk (HDD) [12], and even computer graphics resources [41]. The user selects an operating system and does tasks in the Virtual Machine as if it were a physical component. The hypervisor (virtualization software) separates the enforcement region from local cloud computing users. Each user maintains its own Virtual Machines, and the user's Internet of Things sends messages to the Virtual Machines for processing.

The microCloud [42] avoids this issue by leveraging the capabilities of other edge resources. The microCloud provides the Cloudy software [43], a Docker-based container. Using the framework, a client can deploy tasks to a collection of containers running on various edge resources. The microCloud has the same elasticity as other public clouds. The microCloud focused on locally homogenous resources, but Khan et al. [44] coined the term “microCloud” to describe regionally distributed and heterogeneous resources.

4. Solutions of heterogeneous resources

Given the intrinsic variety of peripheral devices and disparately distributed devices and computing, developing an application in them can be complicated [45–47]. As a result, this evaluation focuses on a set of methods for analyzing and categorizing heterogeneous infrastructure resources. Qian and his colleagues [48] argued that artificial intelligence is an effective strategy for dealing with data heterogeneity. They provided a centralized model derived from a set of local models that had been tested with data relevant to categorical heterogeneity and edge sampling. This section addresses open concerns such as fault tolerance, power consumption, resource usage, latency, device identification, and clustering, as well as AI-based and non-AI-based solutions.

4.1. Research issues

This subsection discusses research issues that arise when dealing with heterogeneous resources at the network's edges. It identifies a number of unresolved concerns, including (fault tolerance, power consumption, resource utilization, resource allocation, latency, device identification, and clustering).

4.1.1. Fault tolerance

Fault tolerance refers to a system's ability (device, edge network, cloud) to continue operating without interruption when one or more of its components fail [14]. The goal of developing a fault-tolerant system is to prevent disruptions caused by a single point of failure, ensuring the high availability and continuity of task-critical applications.

4.1.2. Power consumption

Power consumption refers to the amount of electrical power used per unit time to run something, such as a device, edge network, or cloud. Watts or kilowatts are regularly used units for measuring power use. Because no gadget is 100% efficient, the power used by it is usually greater than what is actually required. Energy is lost as heat, vibrations, and/or electromagnetic radiation [40].

4.1.3. Resource utilization

Resource utilization is the process of determining how efficient resources are. By assigning a task, you structure it, but usage ensures its success [15].

4.1.4. Resource allocation

Resource allocation relates to selecting the appropriate devices for a task, managing them during task execution, and re-assigning or converting the burden as needed [23].

4.1.5. Latency

The term “latency” refers to the time it takes for data to get from one point to another across a network. A network with high latency will respond slowly, whereas a network with low latency will respond quickly [19].

4.1.6. Device ID

The device ID is a concealed string of letters and numbers that uniquely identifies mobile devices. It is mainly used to indicate the model of a mobile device. Mobile applications and programs typically retrieve the device ID while connecting with servers to identify themselves [39].

4.1.7. Clustering

A cluster network is a type of computer network in which all devices are connected with one another using a central device. The central device normally does not have a dedicated connection to the rest of the network, but it does have access to all devices in the cluster. Cluster networks are widely used for distributed computing because they can handle large volumes of traffic and provide fault tolerance if one or more units fail. Cluster networks are particularly renowned among research communities that demand data participation across several laboratories because they provide an appropriate pattern for researchers in diverse positions to collaborate on activities without the requirement for costly hardware connections across laboratories. Cluster-network systems could consist of a large number of individual devices connected into clusters that span continents [2]. Each of these devices has access to all of the information on every other device in the system, and that information is constantly updated. All cluster devices operate simultaneously, allowing clients to query any information they want and receive an instant response. A network like this would be extremely expensive to set up, requiring physical hardware connections between all machines. However, software methods that exist that allow you to create virtual networks over current commodity Internet connections without the need for costly gear or dedicated support staff [33].

4.2. AI-based

Artificial intelligence (AI) refers to the apps or technology that replicate human decision-making. This section addresses the AI-related research issues raised in Section 4. The artificial intelligence techniques in this study will be classified into four categories: machine learning (ML), deep learning (DL), convolutional neural network (CNN), edge learning (EL), federated learning (FL), and federated edge learning (FEEL).

4.2.1. Machine learning (ML)

Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on creating programs that learn and improve performance using data. For example, to lower the rate of offloading failure in a heterogeneous environment, address the job offloading issue in a device dynamically. The method utilized was based on modeling an edge network, which is fundamentally mobile and dynamic, and measuring the device cost, penalty of failure, and variety of quality of service needs. The task of loading is changed into an ongoing decision-making issue in a random process [49]. Hou et al. [50] developed an edge network paradigm in which a platform provides incentives for tools to orchestrate task and resource heterogeneity. This paradigm is an included optimization in which the platform sets scheduling and incentive schemes while edge nodes decide service capabilities. At the same time, each handling task minimizes its overall cost by self-redirecting through accessible edge nodes. Gupta et al. [51] proposed enhanced storage for information-centric networking with the Internet of Things by enabling AI-centric collaborative filtering at the cloud edge to cement IoT design heterogeneity. Tsung et al. [52] proposed a design for efficient processing and energy consumption using an ad hoc parallel artificial intelligence CPU. Immediate data access is used to transmit data between network terminals and CPUs, reducing DRAM traffic. Li et al. [53] presented the HLPN technique to model collection in edge environments, as edge computing consists of three diverse environments, and the HLPN technique addresses heterogeneity by validating acceptability using machine learning. Heterogeneity-Aware Approaches: The heterogeneous nature of edge computing systems, in contrast to typical cloud techniques that have supposed homogeneous devices, while at the same time is a requirement to consider heterogeneous challenges [54–56].

4.2.2. Deep learning (DL)

Deep learning (DL) is a machine learning approach that involves dealing with numerous levels in an artificial neural network. For example, a technique is developed that employs a cluster-centric heterogeneous edge network by integrating a collection of tools such as Docker, Kubernetes, Prometheus, Grafana, and Node Exporter to administer the network in a variety of ways, including assessment and monitoring [57]. Sun et al. [58] proposed deep learning as a new blueprint based on a mathematical paradigm for enabling the Internet of Things to adaptively cooperate in order to produce high-quality applications of heterogeneous devices. Han et al. [59] proposed a methodology designed specifically to accelerate the training of distribution and gossip-centric deep learning in heterogeneous edge computing systems. Carlier et al. [60] proposed multi-agent systems for resource management that blend Avatar and IoT-A architectures to share diverse resources. The authors of [61] presented an EdgeAgent-based solution for managing a set of entities. This paradigm incorporates some of the characteristics that define diverse resources. Shi et al. [62] discussed this study of problem edge network heterogeneity. They developed the mathematical framework for adaptive deep neural network paradigm breakdown and compute offloading. This modeling contains a large number of binary variables, which increases the solution area substantially, making it challenging to solve in multi-job scenes. Thus, they used mobility programming and greedy policy to reduce the solution area under the hypothesis of an acceptable solution. Cheng et al. [63] conducted a STELF-related investigation in which they looked for and examined the precision and processing efficiency of FF-DNN and R-DNN models.

4.2.3. Convolutional neural network (CNN)

The convolutional neural network (CNN) is a component of the deep learning network architecture that is used for classification and computer vision applications. Huang et al. [64] presented 'RoofSplit', a framework for combining bandwidth, storage, and CPU capabilities. Machine learning (CNN) can be used in this framework to obtain excellent performance in edge computing while avoiding the issue of characteristic heterogeneity. Chen and Qin [65] developed a distributed federated training technique by breaking down a training label convolutional neural network paradigm into individually trainable sub-paradigms that are compatible with a subset of learning jobs for each edge machine. When sub-paradigms are well-trained on edge servers, the paradigm variables for specific learning jobs can be collected from each edge machine and combined into the universal training paradigm on a single machine.

4.2.4. Federated learning (FL)

Federated learning (FL) is a machine learning method that trains a methodology over numerous separate sessions, each with its own dataset. A new collaborative learning scheme, known as the HeteroFL approach, is developed to deal with heterogeneous clients that have different CPU and communication capacities. This strategy focuses on training varied local paradigms with dynamically changing local capacities [66]. In this study [67], the authors of this study developed a way to categorize customers based on data and create a training network for each group. The method relies on determined weights to locate data matching among customers and adaptively pools them into the optimal number of groups after investing in the interconnection between weights and data of the federated learning network. Zhang et al. [68] proposed a HyFEM technique based on Hybrid collaborative learning that uses a characteristic-resembling formulation to balance consumers establishing accurate local paradigms and servers learning an accurate global paradigm. Zhang et al. [69]. This study developed the Federated Learning model (FedAda) for combining environment capacity and client data merits to adaptively assign appropriate jobs to each customer. This solution is based on an adaptive job allocation mechanism that addresses the runtime issue among customers while increasing communication in a heterogeneous edge network. Ryffel et al. [70] created the FedSGD algorithm, which simulates a collaborative setting by investing in a random portion of the devices and all of the data on this device. The mean of the gradients is calculated using the server in a way that is proportional to the amount of training samples on each device, and it is used to perform a gradient descent phase. McMahan et al. [71] created the FedAvg algorithm, which allows local devices to do several batch modifications on local data and transfers updated weights rather than gradients. Acar et al. [72] introduced the FedDyn approach to address dataset heterogeneity difficulties. This technique dynamically modifies each node's loss function, bringing the updated node losses near to the true global loss. Vahidian et al. [73] proposed the Sub-FedAvg technique to open a new collaborative learning technique model by introducing Hybrid Pruning. Sub-FedAvg aims to create a "Lottery Ticket Hypothesis" that compares centralized machine learning techniques to collaborative learning techniques developed neutrally. Yeganeh et al. [74] introduced Inverse Distance Aggregation, a variable weighing method for consumers based on metadata that analyzes dependent and non-identical centralized data. It uses the distance between paradigm weights and biases as a policy to reduce variation and improve convergence. Overman et al. [75] proposed a HyFDCA to address convex difficulties in the hybrid collaborative learning adjustment. Jaggi et al. [76] and Smith et al. [77] showed this technique in a scenario in which both samples and characteristics are split down by customers. Jiang et al. [78] proposed a customized federated learning technique with a gradient descent optimization to account for accuracy heterogeneity and accelerate training. Furthermore, they implemented an equitable universal aggregation policy for the edge node to limit the difference in precision gaps amongst machines' heterogeneity. Chen et al. [79] proposed an algorithm that differentiates service species and assigns various cloud-centric cooperative resources based on service providing species in order to meet the corresponding high performance computing requirements of services and grantee ideal assignment of edge devices. Deshmukh et al. [80] presented an approach known as Data Spine, which involves a unified platform capable of overcoming IoT difficulties by collaborating among end devices from various IoT platforms. Ahmed et al. [81] established a paradigm of federated transfer learning that takes users' heterogeneity into consideration and provides services for them. Wang et al. [82] proposed an algorithm named (CoCo) to accelerate decentralized federated learning by improving network design and paradigm compression to overcome system mobility and heterogeneous traffic constraints. Chen et al. [83] used a collaborative learning technique to improve training performance for diverse IoT terminal machines in crucial communication environments. They specifically configure a collaborative learning model and expand a lightweight server choice algorithm to efficiently execute learning jobs. Cai et al. [84] investigated ways to solve the issue of heterogeneity in edge resources for cooperation. By studying diverse edge resources, we can identify collaboration challenges and device sharing opportunities. However, Khan et al. [44] pioneered the concept of microCloud large-scale deployment with heterogeneous resources.

4.2.5. Edge learning (EL)

Edge learning is a subfield of AI that focuses on tackling problems on-machine, or at the end machine of the data source, using a pre-trained set of approaches. Compared to other approaches like deep learning and CNN, it is simple to set up, takes less time, and requires fewer photos for learning [85]. Zhan and Zhang [86] Provide the optimum pricing approach that can learn autonomously by mechanism edge learning. Liao et al. [87]. Introduced a distributed EL solution to improve fault tolerance among computing resource helpers and requesters in cognitive systems. Jia et al. [88]. Worked using Lyapunov optimization theory to construct and analyze a cost-effective optimization. Zhang et al. [89] employed a memristor device to improve the learning algorithm while reducing energy consumption. Li et al. [90] proposed an Edge-LaaS framework for knowledge-centric linked healthcare that locally processes health supervision data. Wen et al. [91] proposed a partitioned edge learning approach for routinely training an AI model across several end devices. Ding et al. [92] proposed a strategy to aid unmanned aerial vehicles with mobile edge computing by increasing secure computation efficiency. Liu et al. [93] suggested the importance aware automatic-repeat-request protocol to address the retransmission decision issue during each communication round.

4.2.6. Federate edge learning (FEEL)

Federated edge learning is the coordination of edge nodes through an edge server to build a collaborative machine learning model using locally distributed data samples [94]. Zhang et al. [95] attempted to formulate and solve the latency issue through a multidisciplinary effort that combined learning and communication. Taïk et al. [96] proposed data-aware scheduling for federated edge learning that considers both data and resource perspectives. Sun et al. [97] proposed an approach to improve training performance while considering machine energy limits, including communication and processing. Mo et al. [98] demonstrated successful answers to the outlined power reduction concerns using convex optimization techniques. Zeng et al. [99] proposed a power-efficient strategy that involves adjusting to machines' path cases and processing capabilities in order to minimize their total power usage. Zeng et al. [100] proposed heterogeneous computing to improve power efficiency and performance, contributing to the energy-efficient implementation of federated edge learning. Albaseer et al. [101] provided fundamental architectural ideas for enabling federated learning at edge networks while taking into account the issue of unlabeled data. They adopted the FedSem technique, which invests anonymous data in high performance. Lin et al. [102] established a framework known as the Social Federated Edge Learning Framework (SFEL) over the internet, which enlists dependable Who is qualified to participate in learning? Du et al. [103] proposed a dynamic machine scheduling method that can identify qualifying edge machines and deploy their local models with an appropriate energy control strategy to share model training over-the-air. Feng et al. [104] proposed a framework for heterogeneous computing and resource allocation based on a heterogeneous mobile architecture in order to achieve accurate federated learning processing. Luo et al. [105] introduced the Hierarchical Federated Edge Learning (HFEL) technique, in which paradigm aggregation is partially transferred from the cloud to edge servers. He et al. [106] developed a method for importance-aware common data selection and resource allocation to improve learning performance while including the federated edge learning (FEEL) technique. Wen et al. [107] developed a training process for the hierarchical federated edge learning framework that included ML model updating phases, local gradient computing, and weighted gradient uploading. Guo et al. [108] proposed a federated edge learning technique, LightFed where the edge devices deploy merely vital partial local models, and attain model aggregation and efficiency communication. Furthermore, they suggested a Training Filling Model (TFM) to determine the entire data distribution of edge devices and train a filling model to address uneven training data while protecting data privacy. In addition, a blockchain-powered confusion transmission approach was given to specify the assaults that safeguard model data. Zhu et al. [109] developed a digital version of broadband using AirComp aggregation, known as one-bit broadband digital aggregation (OBDA). This version addresses a difficult-to-implement FEEL technique in wireless networks using digital modulation. Kang et al. [110] proposed a many-to-one matching approach to address the mission assignment issue between mission publishers and servers that are trustworthy. The block chain methodology is used to maintain training records and handle reputation data in a decentralized and secure manner, eliminating the possibility of a single point of failure.

4.2.7. Heuristic-based and meta-heuristic-based

The term heuristic-based refers to a set of problem-solving techniques that favor discovering answers quickly rather than ensuring ideal results. While heuristics have proven useful in many artificial intelligence applications, they also present several research challenges and limitations that must be carefully considered. On the other hand, meta-heuristics reflect common methodologies that can be used to address complex and challenging search problems. These problems are difficult for computer scientists to solve because they require the examination of a large number of combinations, which are frequently exponential. The developers of [111] concentrated on a solution module composed of several modules that launches clients' requested virtual machines on scale-down of available effective servers in order to reduce total data center power consumption. The approach is proposed, which is based on finding a solution for thermodynamic simulated annealing due to resource skewness, which may compel it to scale up VMs. Because many optimization difficulties are NP-Hard classification issues, heuristics and meta-heuristic techniques are used to tackle these types of problems. In unique objective issues, just one objective function needs to be optimized. Some works attempted to solve unique objective engineering problems using heuristic and exact methods [112]. Other articles have evolved meta-heuristics, such as genetic algorithms [113], particle swarm optimization [114,115], and skewness-aware-related [116], to tackle optimization challenges in the engineering field. Multi-aim optimization solutions, such as the Non-dominated Sorting Genetic Algorithm II [117], multiple objective particle swarm optimization [118], multiple objective genetic algorithm [119], multiple objective Bat algorithm for multi-objective [120], and multiple objective grey wolf optimizer [121], have been developed in the state-of-the-art to address multi-objective optimization issues. Beloglazov and Buyya [122] proposed a solution based on ideal online deterministic approaches and heuristics for power and performance efficient dynamic consolidation of virtual machines in cloud data centers.

4.3. Non AI-based

This section shall discuss edge computing research concerns without AI, such as fault tolerance, power consumption, resource utilization, latency, device identification, and clustering. Canali et al. [123] investigated the sequential forwarding strategy in terms of load and delay. Furthermore, they proposed an improved version of the technique that incorporates a delay-aware concept in the event of variable device connectivity in the network. Rodríguez et al. [124] proposed a framework focused on low-expensive and low-energy heterogeneous process units, with the goal of boosting programmers' capabilities and enabling runtime with full investment in computing resources.

Zhou et al. [125] proposed HetMECC, a mechanism for merging end-users, edge nodes, and cloud servers for data aggregation and deployment. Partitioning workloads allows peripheral applications to receive robust and effective cloud services. Beraldi et al. [126] introduced two decentralized load balancing strategies designed to work in a diverse setting. These strategies combine emulation with theoretical frameworks. Koo et al. [127] developed a system for translating and evaluating machines on the network, identifying devices and coordinating resource requests. As a result, clients can use a variety of Internet of Things applications. Li et al. [128] proposed a software-determined edge network heterogeneity to separate the control and data tires. Based on various needs, jobs in edge network heterogeneity are broken down into multiple sub-jobs at the control tire, and the edge server pact that responds to the jobs is built to execute decomposed sub-jobs. Cooke et al. [129] suggested a paradigm for inference based on decentralized processing in edge computing, utilizing hardware heterogeneity and substituting software and GPUs. This paradigm considers the processing cost of applications, uses a variety of device platforms, and takes into consideration environmental heterogeneity. Wang et al. [130] addressed the issue of compute-offloading heterogeneity in an OFDMA-centric CRAN using a mixed mobile edge computing node. An integrated subcarrier, energy allocation, and task breakdown issue is developed to lessen each client's delay. Furthermore, they presented an improved technique known as HFFE to address the challenging improvement issue by converting channel allocation pointers into undetached variables. Several recent works aim to address heterogeneity-related issues in dynamic contexts. For example, Honeybee has been proposed as a work sharing protocol that is used in cycles of heterogeneous dynamic devices to serve the application from a specific device [131]. Nishio et al. [132] proposed a method for managing heterogeneous devices based on service-oriented architecture that share functionality. Bellavista et al. [133] utilized Docker-based containers on a small single-board computer that serves as an edge device for gathering data from many sources. Thus, microCloud provides flexibility similar to traditional cloud computing. It focused on resources that were local and homogeneous. Yao et al. [134] proposed a technique for distributing cloudlet resources without exceeding QoS criteria by dividing the problem into two parts: heterogeneous cloudlet resource determination and distribution. Nguyen et al. [135] proposed a strategy for allocating resources based on an economic theory that includes geo-distributed heterogeneous devices and a set of services. Zhang et al. [136] discussed resource allocation in heterogeneous vehicle systems and edge computing. Kertész et al. [137] presented the Skippy model, which is an ongoing scheduler that enables the seamless placement of serverless edge functions across decentralized and heterogeneous networks. Abedpour et al. [138] tackled an optimization problem by articulating resource allocation in four levels of heterogeneity for IoT applications. Hosseini and Ramzanpoor [139] proposed a method called MOGA that takes into account resource usage and bandwidth waste rate, as well as dependability and application quality of service, in its constraints.

5. Discussion and analysis

In Section 3 of this review, we described the edge computing (EC) ecosystem's infrastructures using a taxonomy scheme. This technique classified EC into three dimensions, as shown in “Fig. 1". Despite the fact that this scheme divides EC infrastructures into three dimensions, EC management infrastructures can only be achieved if all three dimensions are combined into a single unit. For example, the hardware cannot be used without suitable software. As a result, EC infrastructures should be carefully studied, with all three dimensions working together in a federated manner to achieve the management infrastructures of the edge computing ecosystem.

In this section, we examine the state-of-the-art literature listed in the preceding section based on whether or not it uses artificial intelligence, as shown in “Table 2”. We observe that the majority of methodologies and strategies work with heterogeneous data. In contrast to research issues that employ artificial intelligence techniques, we discover that research issues that do not use these techniques must transfer their local data to the location of the models for testing and training, as indicated by the yellow-shaded rows in “Table 2”. Research projects that use artificial intelligence techniques do not need to send their data someplace else; instead, testing and training are performed locally.

Based on a decomposition of different methods and techniques for heterogeneous resources that are used for managing infrastructures in the edge computing (EC) ecosystem, we discovered that techniques of federated learning (FL) and deep learning (DL) received significant attention in fields of infrastructures in the edge computing (EC) ecosystem, as shown in “Fig. 3”. Meanwhile, machine learning (ML) and convolutional neural network (CNN) techniques have garnered little attention because the neural network is simple and the iterations are complex, making ML and CNN black boxes, respectively.

Federated learning techniques have proven useful in allowing clients to train and test their models while also deploying them via the network. This means that the client frequently updates her model based on her own data and the behavior of other customers. Therefore, the excellent performance of this technique can be shown in low-latency and other research concerns such as fault tolerance, power consumption, resource usage, device ID, and clustering, as demonstrated in “Fig. 4", as well as this technique's capacity to solve heterogeneous resource challenges.

Table 2. Research issues AI-based and non AI-based.

Refer.		AI-based								Research issues							Heterogeneous type	Method
No.	Year	ML	DL	CNN	FL	EL	FEEL	heuristic	Meta-heuristic	Fault tolerant	Power consumption	Resources utilization	latency	Device ID	cluster	Resources allocation		
[123]	**2020**									✓			✓				**Network**	**SFS**
[124]	**2021**										✓	✓	✓				**Multi-Processors**	**scheduler**
[66]	2021				✓							✓	✓				Clients	HeteroFL
[67]	2022				✓										✓		Data	AdaCFL
[68]	2020				✓								✓				Data	HyFEM
[69]	2022				✓						✓		✓				MEC environment	FedAda
[70]	2018				✓								✓				Data	FedSGD
[71]	2017				✓					✓			✓				Data	FedAvg
[72]	2021				✓					✓		✓	✓				Data & Device	FedDyn
[73]	2021				✓								✓				Statistical	Sub-FedAvg
[74]	2020				✓								✓				Statistical	IDA
[75]	2022				✓								✓				Data	HyFDCA
[76]	2014				✓								✓				Data	CoCoA
[77]	2018				✓								✓				Data	general-purpose framework
[49]	2019	✓									✓						Network	Decision-making
[50]	2023	✓										✓	✓				Resource & Request	LFID
[78]	2020				✓					✓		✓	✓				End-device	CuFL
[51]	2021	✓											✓				IoT architecture	ICN-IoT
[125]	**2020**									✓	✓	✓	✓				**Multilayer-devices**	**HetMECC**
[57]	2022		✓								✓		✓		✓		Devices	Integrating of platforms
[79]	2020				✓							✓	✓				Devices	distinguishes service types
[58]	2021		✓									✓	✓				Devices	ADC
[127]	**2019**													✓			**Devices & services**	**DNS**
[80]	2021				✓								✓				Platforms	Data Spine
[81]	2021				✓							✓	✓				Resource & architecture	FTL
[82]	2023				✓							✓	✓				Bandwidth	CoCo
[126]	**2020**									✓		✓					**Resources**	**distributed load balancing**
[64]	2023			✓									✓				Resources	RoofSplit
[83]	2021				✓								✓				Nodes	Asynchronous scheme
[59]	2020		✓										✓				Platforms	EdgeGossip
[60]	2020		✓									✓	✓	✓			Resources	New EdgeAgent
[61]	2020		✓									✓			✓		resources	EdgeAgent
[128]	**2022**									✓	✓						**networks**	**SDN**
[62]	2021		✓								✓		✓				Architecture	SDN
[63]	2022		✓								✓	✓	✓				Database	FF-DNN and R-DNN
[129]	**2020**												✓				**Hardware**	**model for reasoning**
[130]	**2020**										✓		✓				**Tasks offloading**	**HFFE**

[84]	2023			✓					✓	✓	✓		Paradigms SmallC, MobInt and FrontAH	SDN/NFV
[65]	2020		✓							✓			Devices	decentralized collaborative training
[52]	2019	✓							✓	✓			Computation	parallel AI processor
[53]	2017	✓							✓	✓			Devices	HLPN & RedEdge
[131]	**2016**							✓	✓		✓		**Nodes**	**Honeybee**
[132]	**2013**							✓	✓		✓		**Resources**	**service-oriented**
[133]	**2017**									✓			**Devices**	**Docker-based containers**
[44]	**2017**									✓			**Resources**	**Docker**
[134]	**2016**										✓		**Cloudlet servers**	**low-complexity heuristic**
[135]	**2021**									✓			**Devices**	**market-based framework**
[136]	**2017**								✓	✓			**Vehicular**	**mean-field approximation**
[137]	**2021**									✓	✓		**Device**	**Skippy**
[86]	2020				✓						✓		nodes	DRL-based
[87]	2020				✓			✓					nodes	CFCRB
[88]	2021				✓				✓		✓		IoT	Lyapunov
[89]	2023				✓				✓				memristor chip	crossbar array
[90]	2019				✓						✓		Sensors, devices	EdgeLaaS
[91]	2020				✓						✓		devices	PABA
[92]	2022				✓					✓			UVA	OELO
[93]	2020				✓						✓		dataset	Importance ARQ
[95]	2022					✓					✓		devices	3D objective
[96]	2021					✓					✓		Devices, data	DAS
[97]	2021					✓			✓				devices	Energy-aware
[98]	2021					✓			✓				devices	NOMA, TDMA
[99]	2020					✓			✓				Devices	scheduling priority
[100]	2021					✓				✓			devices	CCRM
[101]	2020					✓					✓		devices	FedSem
[102]	2021					✓					✓		plaforms	SFEL
[103]	2023					✓			✓				devices	scheduling
[104]	2022					✓			✓				Computing,	Lagrangian dual
[105]	2020					✓			✓		✓		Computing	HFEL
[106]	2020					✓					✓	✓	data	importance-aware
[107]	2022					✓		✓				✓	Resource	closed-form
[108]	2021					✓					✓		IoT	LightFed
[109]	2020					✓					✓		devices	OBDA
[110]	2021					✓							devices	blockchain
[111]	2021						✓		✓		✓		devices	SA-based
[112]	2022						✓			✓			components of applications	MOCSA
[113]	2021						✓				✓		devices	ID3 tree, SVM, GA

(*continued on next page*)

Table 2. (continued)

Refer. No.	Year	AI-based ML	DL	CNN	FL	EL	FEEL	heuristic	Meta-heuristic	Research issues Fault tolerant	Power consumption	Resources utilization	latency	Device ID	cluster	Resources allocation	Heterogeneous type	Method
[114]	2020								✓		✓						IoT	PSO-based
[115]	2020								✓		✓						devices	PSO-based
[116]	2022							✓					✓				devices	SA-based
[117]	2002								✓				✓				—	NSGA-II
[118]	2018								✓		✓						infrastructure	MOPSO
[119]	2018								✓				✓			✓	Platforms of providers	MOGA
[120]	2011							✓								✓	—	MOBA
[121]	2016							✓								✓	—	MOGWO
[122]	2012							✓								✓	Data centers	Optimal online deterministic
[138]	2023																**Components of IoT**	**(GA), k-means**
[139]	2023																**IoT**	**MOGA**

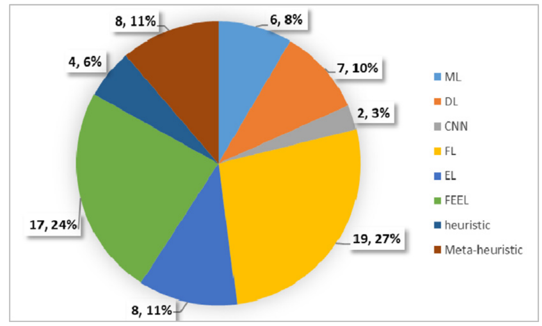

Fig. 3. Percentage of the presented evaluation AI techniques in the review.

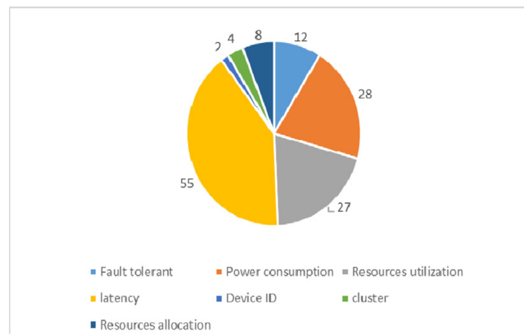

Fig. 4. Numbers of the presented evaluation research issues in the review with AI.

6. Conclusion

The edge network paradigm, along with the advancement of IoT technology, has become increasingly important given its role in improving human lives. As a result, several research have presented approaches and techniques for achieving higher performance while avoiding the problem of heterogeneous resources between the cloud computing layer and the end user layer. In this light, we conducted a thorough analysis of previous research on heterogeneous resources in edge network infrastructures and presented them in a three-dimensional system. Furthermore, the state-of-the-art literature is classified into two categories: research concerns including artificial intelligence (AI) "Intelligent edge" and without AI. A comparison and breakdown of research difficulties with and without AI in order to determine an evaluation metric for meeting Quality of Services (QoS) and Quality of Experience (QoE) standards. Furthermore, a discussion of the overall findings and limitations of the existing study is provided. In future work, for heterogeneous resources in edge network paradigm infrastructures to be thoroughly examined, we emphasize state-of-the-art research difficulties and look forward to greater research investment in blockchain technology to safeguard and maintain the privacy of client data.

Conflicts of interest

There is no conflict of interest.

Acknowledgements

The researchers would like to thank the editors and the anonymous reviewers for their valuable comments and suggestions, which have helped immensely in improving the quality of this review.

References

[1] B. Varghese, R. Buyya, Next generation cloud computing: new trends and research directions, Future Generat Comput Syst 79 (2018) 849−861, https://doi.org/10.1016/j.future.2017.09.020.
[2] M.S. Miah, M. Schukat, E. Barrett, An enhanced sum rate in the cluster based cognitive radio relay network using the sequential approach for the future Internet of Things, Human-centric Comput Info Sci 8 (2018) 1−27, https://doi.org/10.1186/s13673-018-0139-4.
[3] Y. Deng, Z. Chen, D. Zhang, M. Zhao, Workload scheduling to ward worst-case delay and optimal utility for single-hop Fog-IoT architecture, IET Commun 12 (2018) 2164−2173, https://doi.org/10.1049/iet-com.2018.5077.
[4] F. Zeng, K. Zhang, L. Wu, J. Wu, Efficient caching in vehicular edge computing based on edge-cloud collaboration, IEEE Trans Veh Technol 72 (2023) 2468−2481, https://doi.org/10.1109/TVT.2022.3213130.
[5] L. Mottola, G.P. Picco, Programming wireless sensor networks: fundamental concepts and state of the art, ACM Comput Surv 43 (2011) 1−51, https://doi.org/10.1145/1922649.1922656.
[6] M. Balazinska, A. Deshpande, M.J. Franklin, P.B. Gibbons, J. Gray, M. Hansen, V. Tao, Data management in the worldwide sensor web, IEEE Perv Comput 6 (2007) 30−40, https://doi.org/10.1109/MPRV.2007.27.
[7] T. Teixeira, S. Hachem, V. Issarny, N. Georgantas, Service oriented middleware for the internet of things: a perspective, in: European conference on a service-based internet, Springer Berlin Heidelberg, Heidelberg. 2011, pp. 220−229, https://doi.org/10.1007/978-3-642-24755-2_21, 6994.
[8] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for VM-based cloudlets in mobile computing, IEEE Perv Comput 8 (2009) 14−23, https://doi.org/10.1109/MPRV.2009.82.
[9] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30−39, https://doi.org/10.1109/MC.2017.9.
[10] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, 2012, pp. 13−16, https://doi.org/10.1145/2342509.2342513.
[11] A.V. Dastjerdi, R. Buyya, Fog computing: helping the Internet of Things realize its potential, Computer 49 (2016) 112−116, https://doi.org/10.1109/MC.2016.245.
[12] C.H. Hong, B. Varghese, Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms, ACM Comput Surv (CSUR) 52 (2019) 1−37, https://doi.org/10.1145/3326066.
[13] D.G. Amalarethinam, A.M. Josphin, Dynamic task scheduling methods in heterogeneous systems: a survey, Int J Comput Appl 110 (2015) 12−18, https://doi.org/10.5120/19318-0859.
[14] Q. Luo, S. Hu, C. Li, G. Li, W. Shi, Resource scheduling in edge computing: a survey, IEEE Commun Surv Tutor 23 (2021) 2131−2165, https://doi.org/10.48550/arXiv.2108.08059.
[15] M. Ghobaei-Arani, A. Souri, A.A. Rahmanian, Resource management approaches in fog computing: a comprehensive review, J Grid Comput 18 (2020) 1−42. https://link.springer.com/article/10.1007/s10723-019-09491-1.
[16] K. Toczé, S. Nadjm-Tehrani, A taxonomy for management and optimization of multiple resources in edge computing, Wireless Commun Mob Comput 2018 (2018) 7476201, https://doi.org/10.1155/2018/7476201.
[17] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. De Foy, Y. Zhang, Mobile edge cloud system: architectures, challenges, and approaches, IEEE Syst J 12 (2017) 2495−2508, https://doi.org/10.1109/JSYST.2017.2654119.
[18] P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading, IEEE Commun Surv Tutor 19 (2017) 1628−1656, https://doi.org/10.48550/arXiv.1702.05309.
[19] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: state-of-the-art and research challenges, IEEE Commun Surv Tutor 20 (2017) 416−464, https://doi.org/10.1109/COMST.2017.2771153.
[20] D. Alsadie, Resource management strategies in fog computing environment-A comprehensive review, Int J Comp Sci Netw Secur 22 (2022) 310−328. https://koreascience.kr/article/JAKO202213341743831.pdf.
[21] M. Goudarzi, M.S. Palaniswami, R. Buyya, A distributed deep reinforcement learning technique for application placement in edge and fog computing environments, IEEE Trans Mobile Comput 22 (2021) 2491−2505, https://doi.org/10.1109/TMC.2021.3123165.
[22] P. Kansal, M. Kumar, O.P. Verma, Classification of resource management approaches in fog/edge paradigm and future research prospects: a systematic review, J Supercomput 78 (2022) 13145−13204, https://doi.org/10.1007/s11227-022-04338-1.
[23] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, R. Buyya, Resource allocation and task scheduling in fog computing and internet of everything environments: a taxonomy, review, and future directions, ACM Comput Surv 54 (2022) 1−38, https://doi.org/10.1145/3513002.
[24] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.C. Chen, H. Zhang, Reliable computation offloading for edge-computing-enabled software-defined IoV, IEEE Internet Things J 7 (2020) 7097−7111, https://doi.org/10.1109/JIOT.2020.2982292.
[25] I. Sarkar, S. Kumar, Deep learning-based energy-efficient computational offloading strategy in heterogeneous fog computing networks, J Supercomput 78 (2022) 15089−15106, https://doi.org/10.1007/s11227-022-04461-z.
[26] H. Ning, Y. Li, F. Shi, L.T. Yang, Heterogeneous edge computing open platforms and tools for internet of things, Future Generat Comput Syst 106 (2020) 67−76, https://doi.org/10.1016/j.future.2019.12.036.
[27] B. Confais, A. Lebre, B. Parrein, An object store service for a fog/edge computing infrastructure based on ipfs and a scale-out nas, in: 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), IEEE. 2017, pp. 41−50, https://doi.org/10.1109/ICFEC.2017.13.
[28] P.P. Ray, An introduction to dew computing: definition, concept and implications, IEEE Access 6 (2017) 723−737. https://api.semanticscholar.org/CorpusID:3324933.
[29] B. Varghese, R. Buyya, Next generation cloud computing: new trends and research directions, Future Generat Comput Syst 79 (2018) 849−861, https://doi.org/10.1016/j.future.2017.09.020.
[30] M. Noreikis, Y. Xiao, A. Ylä-Jaäiski, QoS-oriented capacity planning for edge computing, in: 2017 IEEE International

Conference on Communications (ICC), 2017, pp. 1–6, https://doi.org/10.1109/ICC.2017.7997387.
[31] D. Jo, G.J. Kim, IoT+ AR: pervasive and augmented environments for "Digi-log" shopping experience, Human-centric Comput Info Sci 9 (2019) 1–17, https://doi.org/10.1186/s13673-018-0162-5.
[32] Y. Deng, Z. Chen, D. Zhang, M. Zhao, Workload scheduling toward worst-case delay and optimal utility for single-hop Fog-IoT architecture, IET Commun 12 (2018) 2164–2173, https://doi.org/10.1049/iet-com.2018.5077.
[33] S.J. Johnston, P.J. Basford, C.S. Perkins, H. Herry, F.P. Tso, D. Pezaros, J. Singer, Commodity single board computer clusters and their applications, Future Generat Comput Syst 89 (2018) 201–212, https://doi.org/10.1016/j.future.2018.06.048.
[34] B. Amento, B. Balasubramanian, R.J. Hall, K. Joshi, G. Jung, K.H. Purdy, FocusStack: Orchestrating edge clouds using location-based focus of attention, in: 2016 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. 2016, pp. 179–191, https://doi.org/10.1109/SEC.2016.22.
[35] P. Bellavista, A. Zanni, Feasibility of fog computing deployment based on docker containerization over raspberrypi, in: Proceedings of the 18th International Conference on Distributed Computing and Networking, 2017, pp. 1–10, https://doi.org/10.1145/3007748.3007777.
[36] H.J. Hong, From cloud computing to fog computing: unleash the power of edge and end devices, in: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE. 2017, pp. 331–334, https://doi.org/10.1109/CloudCom.2017.53.
[37] M. Aazam, E.N. Huh, Fog computing and smart gateway based communication for cloud of things, in: 2014 International Conference on Future Internet of Things and Cloud, IEEE. 2014, pp. 464–470, https://doi.org/10.1109/FiCloud.2014.83.
[38] P. Liu, D. Willis, S. Banerjee, ParaDrop: enabling lightweight multi-tenancy at the network's extreme edge, in: 2016 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. 2016, pp. 1–13, https://doi.org/10.1109/SEC.2016.39.
[39] A. Gosain, M. Berman, M. Brinn, T. Mitchell, C. Li, Y. Wang, H. Zhang, Enabling campus edge computing using GENI racks and mobile resources, in: 2016 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. 2016, pp. 41–50, https://doi.org/10.1109/SEC.2016.24.
[40] E. Ahvar, A.C. Orgerie, A. Lebre, Estimating energy consumption of cloud, fog, and edge computing infrastructures, IEEE Transact Sustain Comput 7 (2019) 277–288, https://doi.org/10.1109/TSUSC.2019.2905900.
[41] C.H. Hong, I. Spence, D.S. Nikolopoulos, GPU virtualization and scheduling methods: a comprehensive survey, ACM Comput Surv 50 (2017) 1–37, https://doi.org/10.1145/3068281.
[42] R. Morabito, N. Beijar, Enabling data processing at the network edge through lightweight virtualization technologies, in: 2016 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops), IEEE. 2016, pp. 1–6, https://doi.org/10.1109/SECONW.2016.7746807.
[43] M. Selimi, A.M. Khan, E. Dimogerontakis, F. Freitag, R.P. Centelles, Cloud services in the Guifi.net community network, Comput Network 93 (2015) 373–388, https://doi.org/10.1016/j.comnet.2015.09.007.
[44] Y. Elkhatib, B. Porter, H.B. Ribeiro, M.F. Zhani, J. Qadir, E. Rivière, On using micro-clouds to deliver the fog, in: IEEE Internet Computing vol. 21, IEEE. 2017, pp. 8–15, https://doi.org/10.1109/MIC.2017.35.
[45] E. Serral, P. Valderas, V. Pelechano, Towards the model driven development of context-aware pervasive systems, Pervasive Mob Comput 6 (2010) 254–280, https://doi.org/10.1016/j.pmcj.2009.07.006.
[46] A. Awan, S. Jagannathan, A. Grama, Macroprogramming heterogeneous sensor networks using cosmos, ACM SIGOPS - Oper Syst Rev 41 (2007) 159–172, https://doi.org/10.1145/1272996.1273014.
[47] L. Mottola, G.P. Picco, Programming wireless sensor networks: fundamental concepts and state of the art, ACM Comput Surv 43 (2011) 1–51, https://doi.org/10.1145/1922649.1922656.
[48] J. Qian, L.K. Hansen, X. Fafoutis, P. Tiwari, H.M. Pandey, Robustness analytics to data heterogeneity in edge computing, Comput Commun 164 (2020) 229–239, https://doi.org/10.1016/j.comcom.2020.10.020.
[49] F. Zhang, J. Ge, C. Wong, C. Li, X. Chen, S. Zhang, V. Chang, Online learning offloading framework for heterogeneous mobile edge computing system, J Parallel Distr Comput 128 (2019) 167–183, https://doi.org/10.1016/j.jpdc.2019.02.003.
[50] Y. Liang, M. Lu, Z.J.M. Shen, R. Tang, Data center network design for internet-related services and cloud computing, Prod Oper Manag 30 (2021) 2077–2101, https://doi.org/10.1111/poms.13355.
[51] D. Gupta, S. Rani, S.H. Ahmed, S. Verma, M.F. Ijaz, J. Shafi, Edge caching based on collaborative filtering for heterogeneous ICN-IoT applications, Sensors 21 (2021) 5491, https://doi.org/10.3390/s21165491.
[52] P.K. Tsung, T.C. Chen, C.H. Lin, C.Y. Chang, J.M. Hsu, Heterogeneous computing for edge AI, in: 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), IEEE. 2019, pp. 1–2. https://api.semanticscholar.org/CorpusID:195222740.
[53] C.S. Li, F. Darema, V. Chang, Distributed behavior model orchestration in cognitive internet of things solution, Enterprise Inf Syst 12 (2018) 414–434, https://doi.org/10.1080/17517575.2017.1355984.
[54] C. Hillman, Y. Ahmad, M. Whitehorn, A. Cobley, Near real-time processing of proteomics data using Hadoop, Big Data 2 (2014) 44–49, https://doi.org/10.1089/big.2013.0036.
[55] R. Sugumaran, J. Burnett, A. Blinkmann, Big 3d spatial data processing using cloud computing environment, in: Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, 2012, pp. 20–22, https://doi.org/10.1145/2447481.2447484.
[56] J.A. Hartigan, M.A. Wong, Algorithm AS 136: a k-means clustering algorithm, J Royal Statist Soc Series C (Applied statistics) 28 (1979) 100–108, https://doi.org/10.2307/2346830.
[57] Y.W. Chan, H. Fathoni, H.Y. Yen, C.T. Yang, Implementation of a cluster-based heterogeneous edge computing system for resource monitoring and performance evaluation, IEEE Access 10 (2022) 38458–38471, https://doi.org/10.1109/ACCESS.2022.3166154.
[58] D. Sun, J. Wu, J. Yang, H. Wu, Intelligent data collaboration in heterogeneous-device IoT platforms, ACM Trans Sens Netw 17 (2021) 1–17, https://doi.org/10.1145/3427912.
[59] R. Han, S. Li, X. Wang, C.H. Liu, G. Xin, L.Y. Chen, Accelerating gossip-based deep learning in heterogeneous edge computing platforms, IEEE Trans Parallel Distr Syst 32 (2020) 1591–1602, https://doi.org/10.1109/TPDS.2020.3046440.
[60] F. Carlier, V. Fresse, J.P. Jamont, A. Rosay, L. Pallardy, A Multi-Edge-Agent system approach for sharing heterogeneous computing resources, in: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), IEEE. 2020, pp. 1–6, https://doi.org/10.1109/VTC2020-Fall49728.2020.9348722.
[61] F. Carlier, V. Fresse, J.P. Jamont, L. Pallardy, A. Rosay, Pooling of heterogeneous computing resources: a novel approach based on Multi-Edge-Agent concept, ICAART 1 (2020) 185–192, https://doi.org/10.5220/0008954301850192.
[62] L. Shi, Z. Xu, Y. Shi, Y. Fan, X. Ding, Y. Sun, A DNN inference acceleration algorithm in heterogeneous edge computing: joint task allocation and model partition, in: Collab Comput: Networking, Applications and Worksharing: 16th EAI International Conference, CollaborateCom 2020, Proceedings, Part I, Springer, Cham. 2021, pp. 237–254, https://doi.org/10.1007/978-3-030-67537-0_15, 349.

[63] Y. Cheng, J. Du, Y. Yang, Z. Ma, N. Li, J. Zhao, D. Wu, Designing collaborative edge computing for electricity heterogeneous data based on social IoT systems, Int J Distr Syst Technol 13 (2022) 1−22, https://doi.org/10.4018/IJDST.307955.
[64] Y. Huang, H. Zhang, X. Shao, X. Li, H. Ji, RoofSplit: an edge computing framework with heterogeneous nodes collaboration considering optimal CNN model splitting, Future Generat Comput Syst 140 (2023) 79−90, https://doi.org/10.1016/j.future.2022.10.006.
[65] X. Chen, Z. Qin, Exploring decentralized collaboration in heterogeneous edge training, in: 2020 IEEE/ACM Symposium on Edge Computing (SEC), IEEE. 2020, pp. 450−453, https://doi.org/10.1109/SEC50012.2020.00069.
[66] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: challenges, methods, and future directions, IEEE Signal Process Mag 37 (2020) 3 50−60, https://doi.org/10.1109/MSP.2020.2975749.
[67] B. Gong, T. Xing, Z. Liu, J. Wang, X. Liu, Adaptive clustered federated learning for heterogeneous data in edge computing, Mobile Network Appl 27 (2022) 1520−1530, https://doi.org/10.1007/s11036-022-01978-8.
[68] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, Y. Khazaeni, Bayesian nonparametric federated learning of neural networks, in: International Conference on Machine Learning, 2019, pp. 7252−7261, https://doi.org/10.48550/arXiv.1905.12022.
[69] J. Zhang, X. Cheng, C. Wang, Y. Wang, Z. Shi, J. Jin, T. Zhang, FedAda: fast-convergent adaptive federated learning in heterogeneous mobile edge computing environment, World Wide Web 25 (2022) 1971−1998, https://doi.org/10.1007/s11280-021-00989-x.
[70] Z. Ghodsi, T. Gu, S. Garg, Safety nets: verifiable execution of deep neural networks on an untrusted cloud, in: Advances in Neural Information Processing Systems, 2017, pp. 4672−4681, https://doi.org/10.48550/arXiv.1706.10268.
[71] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR. 2017, pp. 1273−1282, https://doi.org/10.48550/arXiv.1602.05629.
[72] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc Mach Learn Syst 2 (2020) 429−450, https://doi.org/10.48550/arXiv.1812.06127.
[73] S. Vahidian, M. Morafah, B. Lin, Personalized federated learning by structured and unstructured pruning under data heterogeneity, in: 2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW), IEEE. 2021, pp. 27−34, https://doi.org/10.48550/arXiv.2105.00562.
[74] Y. Yeganeh, A. Farshad, N. Navab, S. Albarqouni, Inverse distance aggregation for federated learning with non-IID data, in domain adaptation and representation transfer, and distributed and collaborative learning: second MICCAI workshop, DART 2020, and First MICCAI Workshop, DCL 2020, Held in Conjunction with MICCAI 2020, Proceedings 2, Springer International Publishing. 2020, pp. 150−159, https://doi.org/10.48550/arXiv.2008.07665.
[75] Y. Liu, X. Zhang, Y. Kang, L. Li, T. Chen, M. Hong, Q. Yang, FedBCD: a communication-efficient collaborative learning framework for distributed features, IEEE Trans Signal Process 70 (2022) 4277−4290, https://doi.org/10.1109/TSP.2022.3198176.
[76] O. Shamir, N. Srebro, T. Zhang, Communication-efficient distributed optimization using an approximate Newton-type method, in: International Conference on Machine Learning, 2014, pp. 1000−1008, https://doi.org/10.48550/arXiv.1312.7853, 32.
[77] V. Smith, S. Forte, M. Chenxin, M. Takáč, M.I. Jordan, M. Jaggi, CoCoA: a general framework for communication-efficient distributed optimization, J Mach Learn Res 18 (2018) 1−49, https://doi.org/10.48550/arXiv.1611.02189.
[78] H. Jiang, M. Liu, B. Yang, Q. Liu, J. Li, X. Guo, Customized federated learning for accelerated edge computing with heterogeneous task targets, Comput Netw 183 (2020) 107569, https://doi.org/10.1016/j.comnet.2020.107569.
[79] B. Chen, G. Wu, Q. Zhang, X. Tao, Heterogeneous sharing resource allocation algorithm in power grid internet of things based on cloud-edge collaboration, in: The 10th International Conference on Computer Engineering and Networks, Springer Singapore. 2021, pp. 1403−1413, https://doi.org/10.1007/978-981-15-8462-6_160.
[80] R.A. Deshmukh, D. Jayakody, A. Schneider, V. Damjanovic-Behrendt, Data spine: a federated interoperability enabler for heterogeneous IoT platform ecosystems, Sensors 21 (2021) 4010, https://doi.org/10.3390/s21124010.
[81] K.M. Ahmed, A. Imteaj, M.H. Amini, Federated deep learning for heterogeneous edge computing, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE. 2021, pp. 1146−1152, https://doi.org/10.1109/ICMLA52953.2021.00187.
[82] L. Wang, Y. Xu, H. Xu, M. Chen, L. Huang, Accelerating decentralized federated learning in heterogeneous edge computing, IEEE Trans Mobile Comput 22 (2023) 5001−5016, https://doi.org/10.1109/TMC.2022.3178378.
[83] Z. Chen, W. Liao, K. Hua, C. Lu, W. Yu, Towards asynchronous federated learning for heterogeneous edge-powered internet of things, Digital Commun Netw 7 (2021) 317−326, https://doi.org/10.1016/J.DCAN.2021.04.001.
[84] Q. Cai, Y. Zhou, L. Liu, Y. Qi, Z. Pan, H. Zhang, Collaboration of heterogeneous edge computing paradigms: how to fill the gap between theory and practice, in: IEEE Wireless Commun, IEEE. 2023, pp. 1−9, https://doi.org/10.1109/MWC.014.2200283.
[85] W. Xu, Z. Yang, D.W.K. Ng, M. Levorato, Y.C. Eldar, M. Debbah, Edge learning for B5G networks with distributed signal processing: semantic communication, edge computing, and wireless sensing, IEEE J Selected Topics Signal Proc 17 (2023) (2023) 9−39. https://api.semanticscholar.org/CorpusID:249240326.
[86] Y. Zhan, J. Zhang, An incentive mechanism design for efficient edge learning by deep reinforcement learning approach, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE. 2020, pp. 2489−2498, https://doi.org/10.1109/INFOCOM41043.2020.9155268.
[87] S. Liao, J. Wu, S. Mumtaz, J. Li, R. Morello, M. Guizani, Cognitive balance for fog computing resource in Internet of Things: an edge learning approach, IEEE Trans Mobile Comput 21 (2020) 1596−1608, https://doi.org/10.1109/TMC.2020.3026580.
[88] L. Jia, Z. Zhou, F. Xu, H. Jin, Cost-efficient continuous edge learning for artificial intelligence of things, IEEE Internet Things J 9 (2021) 7325−7337, https://doi.org/10.1109/JIOT.2021.3104089.
[89] W. Zhang, P. Yao, B. Gao, Q. Liu, D. Wu, Q. Zhang, Y. Li, Q. Qin, J. Li, Z. Zhu, Y. Cai, Edge learning using a fully integrated neuro-inspired memristor chip, Science 381 (2023) 1205−1211, https://doi.org/10.1126/science.ade3483.
[90] G. Li, G. Xu, A.K. Sangaiah, J. Wu, J. Li, EdgeLaaS: edge learning as a service for knowledge-centric connected healthcare, IEEE Network 33 (2019) 37−43, https://doi.org/10.1109/MNET.001.1900019.
[91] D. Wen, M. Bennis, K. Huang, Joint parameter-and-bandwidth allocation for improving the efficiency of partitioned edge learning, IEEE Trans Wireless Commun 19 (2020) 8272−8286, https://doi.org/10.1109/TWC.2020.3021177.
[92] Y. Ding, Y. Feng, W. Lu, S. Zheng, N. Zhao, L. Meng, A. Nallanathan, X. Yang, Online edge learning offloading and resource management for UAV-assisted MEC secure communications, IEEE J Selected Topics Signal Proc 17 (2022) 54−65, https://doi.org/10.1109/JSTSP.2022.3222910.
[93] D. Liu, G. Zhu, Q. Zeng, J. Zhang, K. Huang, Wireless data acquisition for edge learning: data-importance aware retransmission, IEEE Trans Wireless Commun 20 (2020) 406−420, https://doi.org/10.1109/TWC.2020.3024980.

[94] Y. Sun, J. Shao, Y. Mao, J.H. Wang, J. Zhang, Semi-decentralized federated edge learning with data and device heterogeneity, IEEE Transact Netw Service Manag 20 (2023) 1487−1501, https://doi.org/10.1109/TNSM.2023.3252818.
[95] M. Zhang, G. Zhu, S. Wang, J. Jiang, Q. Liao, C. Zhong, S. Cui, Communication-efficient federated edge learning via optimal probabilistic device scheduling, IEEE Trans Wireless Commun 21 (2022) 8536−8551, https://doi.org/10.1109/TWC.2022.3166941.
[96] A. Taïk, Z. Mlika, S. Cherkaoui, Data-aware device scheduling for federated edge learning, IEEE Transact Cognitive Commun Netw 8 (2021) 408−421, https://doi.org/10.1109/TCCN.2021.3100574.
[97] Y. Sun, S. Zhou, Z. Niu, D. Gündüz, Dynamic scheduling for over-the-air federated edge learning with energy constraints, IEEE J Sel Area Commun 40 (2021) 227−242, https://doi.org/10.1109/JSAC.2021.3126078.
[98] X. Mo, J. Xu, Energy-efficient federated edge learning with joint communication and computation design, J Commun Info Netw 6 (2021) 110−124, https://doi.org/10.23919/JCIN.2021.9475121.
[99] Q. Zeng, Y. Du, K. Huang, K.K. Leung, Energy-efficient radio resource allocation for federated edge learning, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE. 2020, pp. 1−6, https://doi.org/10.1109/ICCWorkshops49005.2020.9145118.
[100] Q. Zeng, Y. Du, K. Huang, K.K. Leung, Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing, IEEE Trans Wireless Commun 20 (2021) 7947−7962, https://doi.org/10.1109/TWC.2021.3088910.
[101] A. Albaseer, B.S. Ciftler, M. Abdallah, A. Al-Fuqaha, Exploiting unlabeled data in smart cities using federated edge learning, in: 2020 International Wireless Communications and Mobile Computing (IWCMC), IEEE. 2020, pp. 1666−1671, https://doi.org/10.1109/IWCMC48107.2020.9148475.
[102] X. Lin, J. Wu, J. Li, X. Zheng, G. Li, Friend-as-Learner: socially-Driven trustworthy and efficient wireless federated edge learning, IEEE Trans Mobile Comput 22 (2023) 269−283, https://doi.org/10.1109/TMC.2021.3074816.
[103] J. Du, B. Jiang, C. Jiang, Y. Shi, Z. Han, Gradient and channel aware dynamic scheduling for over-the-air computation in federated edge learning systems, IEEE J Sel Area Commun 41 (2023) 1035−1050, https://doi.org/10.1109/JSAC.2023.3242727.
[104] J. Feng, W. Zhang, Q. Pei, J. Wu, X. Lin, Heterogeneous computation and resource allocation for wireless powered federated edge learning systems, IEEE Trans Commun 70 (2022) 3220−3233, https://doi.org/10.1109/TCOMM.2022.3163439.
[105] S. Luo, X. Chen, Q. Wu, Z. Zhou, S. Yu, HFEL: joint edge association and resource allocation for cost-efficient hierarchical federated edge learning, IEEE Trans Wireless Commun 19 (2020) 6535−6548, https://doi.org/10.1109/TWC.2020.3003744.
[106] Y. He, J. Ren, G. Yu, J. Yuan, Importance-aware data selection and resource allocation in federated edge learning system, IEEE Trans Veh Technol 69 (2020) 13593−13605, https://doi.org/10.1109/TVT.2020.3015268.
[107] W. Wen, Z. Chen, H.H. Yang, W. Xia, T.Q. Quek, Joint scheduling and resource allocation for hierarchical federated edge learning, IEEE Trans Wireless Commun 21 (2022) 5857−5872, https://doi.org/10.1109/TWC.2022.3144140.
[108] J. Guo, J. Wu, A. Liu, N.N. Xiong, LightFed: an efficient and secure federated edge learning system on model splitting, IEEE Trans Parallel Distr Syst 33 (2021) 2701−2713, https://doi.org/10.1109/TPDS.2021.3127712.
[109] G. Zhu, Y. Du, D. Gündüz, K. Huang, One-bit over-the-air aggregation for communication-efficient federated edge learning: design and convergence analysis, IEEE Trans Wireless Commun 20 (2020) 2120−2135, https://doi.org/10.1109/TWC.2020.3039309.
[110] J. Kang, Z. Xiong, X. Li, Y. Zhang, D. Niyato, C. Leung, C. Miao, Optimizing task assignment for reliable blockchain-empowered federated edge learning, IEEE Trans Veh Technol 70 (2021) 1910−1923, https://doi.org/10.1109/TVT.2021.3055767.
[111] P. Saeedi, M. Hosseini Shirvani, An improved thermodynamic simulated annealing-based approach for resource-skewness-aware and power-efficient virtual machine consolidation in cloud datacenters, Soft Comput 25 (2021) 5233−5260, https://doi.org/10.1007/s00500-020-05523-1.
[112] Y. Ramzanpoor, M. Hosseini Shirvani, M. Golsorkhtabaramiri, Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure, Compl Intell Syst 8 (2022) 361−392, https://doi.org/10.1007/s40747-021-00368-z.
[113] A. Javadian Kootanaee, A.A. Poor Aghajan, M. Hosseini Shirvani, A hybrid model based on machine learning and genetic algorithm for detecting fraud in financial statements, J Optimiz Industr Eng 14 (2021) 169−186, https://doi.org/10.22094/joie.2020.1877455.1685.
[114] S. Azimi, C. Pahl, M.H. Shirvani, Particle swarm optimization for performance management in multi-cluster IoT edge architectures, in: Proceedings of the 10th International Conference on Cloud Computing and Services Science - CLOSER 1, 2020, pp. 328−337, https://doi.org/10.5220/0009391203280337.
[115] M.H. Shirvani, A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems, Eng Appl Artif Intell 90 (2020) 103501, https://doi.org/10.1016/j.engappai.2020.103501.
[116] R. Noorian Talouki, M. Hosseini Shirvani, H. Motameni, A hybrid meta-heuristic scheduler algorithm for optimization of workflow scheduling in cloud heterogeneous computing environment, J Eng Des Technol 20 (2022) 1581−1605, https://doi.org/10.1108/JEDT-11-2020-0474.
[117] K. Deb, A. Pratap, S. Agarwal, T.A. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans Evol Comput 6 (2002) 182−197. https://api.semanticscholar.org/CorpusID:9914171.
[118] M. Hosseini Shirvani, A.M. Rahmani, A. Sahafi, An iterative mathematical decision model for cloud migration: a cost and security risk approach, Software Pract Ex 48 (2018) 449−485, https://doi.org/10.1002/spe.2528.
[119] M.H. Shirvani, Web service composition in multi-cloud environment: a bi-objective genetic optimization algorithm, in: 2018 Innovations in Intelligent Systems and Applications (INISTA), IEEE. 2018, pp. 1−6, https://doi.org/10.1109/INISTA.2018.8466267.
[120] X.S. Yang, Bat algorithm for multi-objective optimisation, Int J Bio-Inspired Comput 3 (2011) 267−274, https://doi.org/10.1504/IJBIC.2011.042259.
[121] S. Mirjalili, S. Saremi, S.M. Mirjalili, L.D. Coelho, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, Expert Syst Appl 47 (2016) 106−119, https://doi.org/10.1016/j.eswa.2015.10.039.
[122] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurrency Comput Pract Ex 24 (2012) 1397−1420, https://doi.org/10.1002/cpe.1867.
[123] C. Canali, R. Lancellotti, S. Mione, Collaboration strategies for fog computing under heterogeneous network-bound scenarios, in: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), IEEE. 2020, pp. 1−8, https://doi.org/10.1109/NCA51143.2020.9306730.
[124] A. Rodríguez, A. Navarro, R. Asenjo, F. Corbera, R. Gran, D. Suárez, J. Nunez-Yanez, Exploring heterogeneous scheduling for edge computing with CPU and FPGA MPSoCs, J Syst Architect 98 (2019) 27−40, https://doi.org/10.1016/j.sysarc.2019.06.006.
[125] H. Zhou, Y. Xiang, H.F. Li, R. Yuan, Task offloading strategy of 6G heterogeneous edge-cloud computing model considering mass customization mode collaborative

manufacturing environment, Math Probl Eng 2020 (2020) 1−8, https://doi.org/10.1155/2020/1059524.
[126] R. Beraldi, C. Canali, R. Lancellotti, G.P. Mattia, Distributed load balancing for heterogeneous fog computing infrastructures in smart cities, Pervasive Mob Comput 67 (2020) 101221, https://doi.org/10.1016/j.pmcj.2020.101221.
[127] J. Koo, S.R. Oh, Y.G. Kim, Device identification interoperability in heterogeneous IoT platforms, Sensors 19 (2019) 1433, https://doi.org/10.3390/s19061433.
[128] Y. Li, B. Wu, Software-defined heterogeneous edge computing network resource scheduling based on reinforcement learning, Appl Sci 13 (2022) 426, https://doi.org/10.3390/app13010426.
[129] R.A. Cooke, S.A. Fahmy, A model for distributed in-network and near-edge computing with heterogeneous hardware, Future Generat Comput Syst 105 (2020) 395−409, https://doi.org/10.1016/j.future.2019.11.040.
[130] S. Wang, C. Pan, C. Yin, Joint heterogeneous tasks offloading and resource allocation in mobile edge computing systems, in: 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE. 2018, pp. 1−6, https://doi.org/10.1109/WCSP.2018.8555725.
[131] N. Fernando, S.W. Loke, W. Rahayu, Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds, IEEE Transact Cloud Comput 7 (2016) 329−343, https://doi.org/10.1109/TCC.2016.2560163.
[132] T. Nishio, R. Shinkuma, T. Takahashi, N.B. Mandayam, Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud, in: Proceedings of the First International Workshop on Mobile Cloud Computing & Networking, 2013, pp. 19−26, https://doi.org/10.1145/2492348.2492354.
[133] P. Bellavista, A. Zanni, Feasibility of fog computing deployment based on docker containerization over raspberrypi, in: Proceedings of the 18th International Conference on Distributed Computing and Networking, 2017, pp. 1−10, https://doi.org/10.1145/3007748.3007777.
[134] H. Yao, C. Bai, M. Xiong, D. Zeng, Z. Fu, Heterogeneous cloudlet deployment and user-cloudlet association toward cost effective fog computing, Concurr Comput Pract Exp 29 (2017) e3975, https://doi.org/10.1002/cpe.3975.
[135] D.T. Nguyen, L.B. Le, V. Bhargava, Price-based resource allocation for edge computing: a market equilibrium approach, IEEE Transact Cloud Comput 9 (2018) 302−317, https://doi.org/10.1109/TCC.2018.2844379.
[136] Y. Zhang, H. Zhang, K. Long, X. Xie, V.C. Leung, Resource allocation in software defined fog vehicular networks, in: Proceedings of The 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Approach, IEEE Transactions on Cloud Computing. 2017, pp. 302−317, https://doi.org/10.1109/TCC.2018.2844379, 9.
[137] T. Rausch, A. Rashed, S. Dustdar, Optimized container scheduling for data-intensive serverless edge computing, Future Generat Comput Syst 114 (2021) 259−271, https://doi.org/10.1016/j.future.2020.07.017.
[138] K. Abedpour, M. Hosseini Shirvani, E. Abedpour, A Genetic-Based Clustering Algorithm for Efficient Resource Allocating of IoT Applications in Layered Fog Heterogeneous Platforms, Cluster Computing. 2023, pp. 1−19, https://doi.org/10.1007/s10586-023-04005-x.
[139] M. Hosseini Shirvani, Y. Ramzanpoor, Multi-objective QoS-aware optimization for deployment of IoT applications on cloud and fog computing infrastructure, Neural Comput Appl 35 (2023) 19581−19626, https://doi.org/10.1007/s00521-023-08759-8.