Multimedia Tools and Applications (2022) 81:1997–2042
https://doi.org/10.1007/s11042-021-11423-9

Towards effective offloading mechanisms in fog computing

Maryam Sheikh Sofla[1] · Mostafa Haghi Kashani[1] · Ebrahim Mahdipour[1] · Reza Faghih Mirzaee[2]

Received: 2 September 2020 / Revised: 6 February 2021 / Accepted: 10 August 2021 /
Published online: 19 October 2021

Abstract
Fog computing is considered a formidable next-generation complement to cloud computing. Nowadays, in light of the dramatic rise in the number of IoT devices, several problems have been raised in cloud architectures. By introducing fog computing as a mediate layer between the user devices and the cloud, one can extend cloud computing's processing and storage capability. Offloading can be utilized as a mechanism that transfers computations, data, and energy consumption from the resource-limited user devices to resource-rich fog/cloud layers to achieve an optimal experience in the quality of applications and improve the system performance. This paper provides a systematic and comprehensive study to evaluate fog offloading mechanisms' current and recent works. Each selected paper's pros and cons are explored and analyzed to state and address the present potentialities and issues of offloading mechanisms in a fog environment efficiently. We classify offloading mechanisms in a fog system into four groups, including computation-based, energy-based, storage-based, and hybrid approaches. Furthermore, this paper explores offloading metrics, applied algorithms, and evaluation methods related to the chosen offloading mechanisms in fog systems. Additionally, the open challenges and future trends derived from the reviewed studies are discussed.

Keywords Fog computing · Offloading · Internet of things (IoT) · Quality of service (QoS)

✉ Ebrahim Mahdipour
mahdipour@srbiau.ac.ir

Maryam Sheikh Sofla
m.sheikh@srbiau.ac.ir

Mostafa Haghi Kashani
mh.kashani@srbiau.ac.ir

Reza Faghih Mirzaee
r.f.mirzaee@qodsiau.ac.ir

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Computer Engineering, Shahr-e-Qods Branch, Islamic Azad University, Tehran, Iran

1 Introduction

Fog computing, which is also known as *fog or fog networking*, is introduced as an emergent and novel paradigm for extending cloud services. Although cloud computing is presented as a model that provides on-demand and ubiquitous access to a shared pool of computing and storage resources, cloud resources are far from users, and as a result, the cloud cannot support low-latency services alone. The fog can extend these computing and storage resources by incorporating a transitional layer between the IoT devices and the cloud, leading to a three-layer hierarchy: user devices layer, fog layer, and cloud layer. The middle fog layer consists of a set of base stations, routers, and gateways that are geographically distributed and places as near as possible to the IoT devices. It is widely known that fog brings *the cloud closer to the ground* (IoT devices) [72, 88].

Regarding the terminology, Cisco was the first which uses the word *fog computing* in 2012 [18]. Fog computing is now a notion that is open to the public. It is currently being endorsed by the OpenFog Consortium. In November 2015, a coalition of academia and industry, including Microsoft, Intel, Cisco, ARM, Dell, and Princeton University, instituted the OpenFog Consortium to catalyze, publicize, and promote the implementation of fog computing [88]. Fog computing delivers further benefits as compared to cloud computing. Low response time for requested services is one of them. While delay-tolerant and compute-intensive components usually occur in the cloud, fog computing can instead be employed to process some application-level components (e.g., delay-sensitive components) at the network edge. Moreover, being closer to the IoT devices enables fog to reduce the cost and latency of services and optimize power consumption [7].

If computation at the network edge is enabled, near to the user devices or *fog nodes*, fog servers can exploit processing at some specific spots, then these facilities provide more advantages than the cloud. In addition, fog provides exceedingly distributed nodes for collecting data produced by the user devices. This task is performed through the gateways, access points, and routers located near the sources at the network edge layer [132]. As discussed in the literature [19], there is a consensus that cloud computing is not applicable in some applications of IoT, and the best alternative for these applications is fog computing. However, fog is not limited to IoT and can be used to deliver content and other applications as well.

In this paper, another notion with which we deal is offloading. Offloading is a strategy that outsources tasks or services with the goal of performing the computation, reducing energy consumption, or storing data. Offloading can be done for a variety of reasons, such as overloaded resources or to improve QoS parameters. By employing this strategy continuously, we can achieve optimization to a higher extent [109]. In fog environments, end-users and IoT devices use applications with low response time in real-time. The aim of offloading in a fog environment is to outsource the load exchanged between fog nodes or between fog nodes and cloud. Using offloading strategies in a fog environment, which involves IoT devices, can optimize various quality parameters, such as power consumption, performance, throughput, response time, resource utilization, and cost.

Task offloading is an important strategy among several different offloading techniques. For IoT applications as well as cloud-fog computing, this strategy–that is–task offloading, is very attractive [9]. Task offloading can be used through user devices, edge devices, sensors, or fog nodes. Offloading deals with several different issues that affect optimization. They include data management, computational requirements of an application, latency management, energy management, and so forth [2, 110]. The task offloading mechanisms

are often employed to make room for resources that are used for computation and storage. For instance, it is required to utilize offloading and transfer the data (or tasks) to a new resource when fog node processes do not run properly on their primary resource. Furthermore, by using offloading mechanisms, we can increase the speed of the computation in fog-cloud environments. When a low-speed processor cannot compute a task, we can compute it faster by offloading the task to a high-speed processor. The main objective of this survey is to present a state-of-the-art review of offloading mechanisms in fog environments.

1.1 Related work and motivation

This section provides a quick explanation of the relevant studies in fog offloading mechanisms. We will refer to some papers in the field of offloading (or close to this topic), which are not exclusively about fog computing.

Concentrating on fog computing in the cloud-IoT environment, Aazam et al. [2] surveyed diverse types of recently proposed offloading techniques in the literature. However, in this paper, many recently published papers were not included. Also, this survey did not provide a systematic review of the literature and did not specify the process for selecting articles. Mobile devices are confined by some restrictions such as low storage, low execution time, and low energy. Based on a contextual computation offloading, Orsini et al. [89] suggested a strategy to overcome these restrictions. They also presented some categories of representative mobile cloud computing (MCC) approaches, and the satisfaction of requirements was assessed. However, this article did not specify the process for selecting articles and did not focus on fog computing.

In addition, Cao et al. [24] introduced the basic concept of multi-access edge computing (MAEC) and its main applications. They also reviewed the essential studies, which employed several different machine learning (ML) based approaches. The authors illustrated the necessity for augmented intelligence in MAEC and discussed basic ideas of typical ML-based approaches in the state-of-the-art research works. Nonetheless, analogous to the papers mentioned above, this work did not provide a systematic review of the literature. Further, the work was not directly related to fog computing either. Also, Wu [126] presented a survey of current studies on decision-making for offloading in the mobile cloud environment. In this regard, the author considered some critical metrics such as energy, cost, storage, and security to assess methods and compare them with one another. Nevertheless, this research was not a systematic paper and was not related to the fog.

Furthermore, some new offloading frameworks and techniques were suggested by Akherfi et al. [5]. They analyzed these techniques and frameworks along with the major critical points related to three major concepts: mobile cloud computing, cloud computing, and computation offloading. Additionally, various approaches that were applied by the frameworks to attain offloading were illustrated in this paper. However, the work did not include many recent published papers and was not directly related to fog computing. Also, this paper did not provide a systematic review of the literature. Mach and Becvar [69] introduced a concept to deal with the delay challenge, known as mobile edge computing (MEC). To enable highly demanding applications to run, the MEC moves storage and computation resources to the mobile network's edge at users' devices. However, this paper did not follow a systematic structure in reviewing the papers. What is more, the focus of the paper was not on fog computing.

In another study, Wang et al. [121] surveyed a collection of mobile data offloading strategies in detail. They suggested a study of cutting-edge offloading mechanisms. Based on

this paper, the two major classifications, infrastructure-less and infrastructure-based, can cover all emerging use cases of driven offloading mechanisms. Nevertheless, the work was not related directly to fog computing. This paper was not a systematic review, and the criteria for choosing the papers were not clear and specific. Chen and Cheng [26] reviewed the state of the art of computation offloading algorithms. They classified the algorithms of computation offloading in MEC into three classes: multiple users, single user, and enhanced MEC server. Also, the pros and cons of each class were defined. However, the focus of the paper was not on fog computing. Moreover, this paper did not follow a systematic structure in reviewing the papers.

Finally, other offloading literature reviews and surveys were investigated, such as [108] by surveying binary translation methods in native offloading based on mobile cloud computing in multimedia applications; [124] by surveying task offloading methods in edge/cloud computing; [65] by surveying offloading modeling approaches in edge computing; [104] by reviewing stochastic offloading techniques in mobile edge computing; [105] by reviewing ML-based offloading techniques in mobile edge computing; and [106] by reviewing game theory-based computation offloading techniques in mobile edge computing.

Almost none of the papers mentioned earlier were directly related to fog computing. They had neither the essential parts of a systematic review, including the process of paper selection, nor a classification of the selected papers. A summary of the previous surveys is given in Table 1, in which the review type, edge technology, publication year, process of paper selection, taxonomy, and covered year of each related study, are shown. In the literature, we sought papers published on or before November 2020, and there was only one work about offloading mechanisms in fog computing [2], which did not follow a systematic structure. Eventually, since there was no systematic and comprehensive review of the offloading mechanisms in fog computing, we surveyed only some relatively close papers to our concern. Briefly, the shortcomings of most of the previous surveys are as follows:

Table 1 Related work in the field of offloading

Review type	Ref	Edge technology	Publication year	Paper selection process	Taxonomy	Covered year
Survey	[2]	Fog computing	2018	Not clear	Yes	Not mentioned
	[89]	MCC	2015	Not clear	No	Not mentioned
	[24]	Edge computing	2019	Not clear	No	Not mentioned
	[126]	MCC	2018	Not clear	No	Not mentioned
	[5]	MCC	2018	Not clear	No	Not mentioned
	[69]	MEC	2017	Not clear	No	Not mentioned
	[121]	Mobile cellular computing	2019	Not clear	Yes	Not mentioned
	[26]	MEC	2019	Not clear	Yes	Not mentioned
	[108]	MCC	2016	Not clear	Yes	Not mentioned
	[124]	Edge computing	2020	Clear	Yes	2016–2020
	[65]	Edge computing	2020	Not clear	Yes	Not mentioned
Systematic review	[104]	MEC	2020	Clear	Yes	2016–2020
	[105]	MEC	2020	Clear	Yes	2013–2020
	[106]	MEC	2020	Clear	Yes	2013–2019
	Our study	Fog computing	2021	Clear	Yes	2016–2020

- Most of the related papers were not investigated in a fog environment.
- Most of the papers did not provide any classifications.
- Some papers did not contain future directions and open issues, which is considered as a major section in surveys.
- Researchers' paper selection methods and processes were not systematic and explicit.
- Recently published papers and state-of-the-art studies were not included in most of the studies.

1.2 Contribution and organization

What motivates us to arrange this study and investigate the offloading mechanisms in fog computing is that there are no systematic studies on this issue, and our scrutiny demonstrates that there is only one work on this subject [2] (on 30/11/2020). Nevertheless, it did not follow the systematic study's standards and structure, and accordingly, did not enjoy the critical sections of a systematic literature review. In this paper, firstly, we study the recent offloading mechanisms. Secondly, we compare the features of each offloading mechanism with others. Finally, we define certain mutual offloading mechanisms, frameworks, and challenges in fog computing. In the following, we mention the key contributions of this paper:

- Providing an exhaustive, comprehensive, and thorough literature study of the current offloading mechanisms in fog computing
- Providing a technical classification of the recent offloading approaches in fog computing
- Surveying the advantages and disadvantages, evaluation methods, and evaluation metrics of the offloading mechanisms in fog computing
- Exploring the open issues and providing possible methods for future works, which can optimize the effectiveness of offloading mechanisms in fog computing

The rest of this paper is structured as follows: After the introduction, we will provide a background of fog computing and offloading in Section 2. In Section 3, we will explore the research methodology and paper selection mechanisms. In Section 4, the selected papers will be reviewed and classified. In Section 5, we will discuss our review results and compare the factors that are evaluated in our study. In Section 6, we will discuss open issues and future trends. Finally, we conclude our work and explain the limitations of the study in Section 7.

2 Background

In this section, we introduce edge technologies and fog computing. Then, the concept and structure of offloading are debated. Moreover, the parameters affecting the efficiency of offloading in fog computing are expressed.

2.1 Edge technology and relevant notions

In this part, fog-like edge technologies, including mobile edge computing, cloudlet, and micro-data centers, are introduced, and then fog-related concepts such as cloud computing and the Internet of things are described.

- *Mobile edge computing (MEC)*: In 2014, MEC was first introduced as an industrial plan backed by ETSI (European Telecommunication Standards Institute) for mobile networks and virtual machines in virtualized technologies. It was developed in March 2017 to meet other networks' needs except for those of mobile networks and meet the needs of virtualizing techniques except those of the virtual machines [78]. MEC is an implementation of edge computing that provides computational capabilities within the Radio Access Network. To reach low response time and improve context awareness, MEC is employed. The nodes of MEC are usually co-sited with a macro base station or the Radio Network Controller. The MEC servers provide information for the network itself and user devices: (1) real-time information for the network, including the loaded tasks and data and also capacity of the network, (2) information for user devices connected to the servers, including transactions data and their location [29].
- *Cloudlet*: The concept of cloudlet was raised in 2009 [102]. This concept makes use of modern techniques like virtual machines in cloud computing. Servers, which are resource-rich and in the proximity of mobile devices, provide the capability to apply mobile devices' resources by using virtual machines. This model has a three-layer architecture composed of mobile devices, cloudlets, and cloud. Cloudlet is a trusted cluster of computers that are connected to the Internet. These clusters are resources available to use for mobile devices that near to the cluster. A Cloudlet may be used as a "data center in a box". This putative data center is run as a virtual device that can provide resources to users' devices in real-time. By employing cloudlets, the services provide access to resources through a high bandwidth. Therefore, low response time is an achievement of this technique of computing [29].
- *Micro-data center*: The micro-data center concept was introduced by Microsoft Research, supervised by Victor Bahl. A micro-data-center is a small data center that is entirely functional, having some servers, and can satisfy many virtual machines. Most technologies, like fog computing, can use micro-data center because it reduces latency and increases reliability, leading to saving in bandwidth consumption and accommodating many new services. Micro-data centers like cloudlets can satisfy the demands of applications with lower response time tasks and improve energy efficiency and quality of computation. A micro-data center has a secure computing environment and a self-contained cell that can perform computation and provide networking and storage equipment to execute customer's tasks. To satisfy end-users' different demands, a micro-data center's size varies from 1 to 100 kW [16].
- *Cloud computing*: Cloud computing, as NIST (National Institute of Standards and Technology) defined in [72], is a model to provide on-demand and ubiquitous access to a shared pool of computing resources (e.g., storage, networks, servers, services, and applications). Cloud computing can be quickly facilitated with minimal organization tasks or service provider interaction. In cloud computing, service models based on PaaS (Platform as a Service), SaaS (Software as a Service), and IaaS (Infrastructure as a Service) are considered. The key features of this computing are

described as follows. *Availability* is one of the main characteristics of this model. Heterogeneous client platforms such as laptops, cell phones, and tablets can access the services through standard mechanisms. *Automaticity* is another feature of cloud computing. A consumer can access needed computing capabilities, such as storage and server time, automatically without appealing human interaction. An advantage of cloud computing over other techniques of computing is *elasticity*. Capabilities can be unlimited accessible elastically in any quantity at any time in the cloud model. Another feature of cloud computing is to provide a measured service that can automatically control and optimize resource used by leveraging a metering capability.

- *Internet of things*: The term Internet of things was first introduced in 1999 by Ashton [10] in supply management. Nowadays, there are diverse definitions of the IoT that can be addressed as a network that consists of physical things or devices, such as mobile phones, sensors, laptops, and RFID (radio-frequency identification) tags. By employing IoT, objects can be executed and controlled remotely within the Internet. This network enables us to integrate the physical world into the virtual world. Objects can stick together in an IoT network and be bundled into logical or geographical clusters. Several IoT clusters produce a massive amount of data at various locations, which have to be analyzed. To process this data efficiently, a combination of multiple computation models such as cloud-fog computing, mobile edge computing, and so forth, can be used [80].

2.2 Fog computing

As a system-level horizontal architecture, fog computing is named, defined, and outlined by the OpenFog Consortium. Resources, storage power, networking, computing services, and control functions are located everywhere, from the cloud to users' devices [87]. As a development of the cloud-based computing scheme, the fog computing model can be presented in multiple layers of a network's topology. In several circumstances, fog computing can work with cloud computing. Although *fog computing is often called edge computing by mistake*, there are essential differences between them. While fog does work with the cloud, the edge is employed as a computation that excludes the cloud. The fog has a hierarchical-layered architecture, whereas the edge is confined by the minimum number of layers, and it is surrounded by a bunch of peripheral devices. Applications that are run in a fog environment are executed in a multi-layer architecture. Allowing the dynamic reconfigurations for various applications, this architecture meshes and disconnects the software and hardware functions. In contrast, edge computing forms a direct transmission service and organize special applications architecturally at a fixed place [43, 77, 87].

In addition to computation, fog also considers several different issues, including storage, acceleration, control, and networking. As a platform, fog computing is exceedingly virtualized. This platform, which is not exclusively located at the network edge, offers storage, computation, and services between users' devices and the traditional cloud data centers [116]. We can characterize the essential fog computing features that distinguish fog computing from other methods of computing as follows. However, not all of the following characteristics need to be used in IoT users' applications [37, 43].

- *Heterogeneity*: Fog computing backs data processing and collection related to various form factors obtained through multiple types of network communication capabilities.

- *Geographical distribution*: A significant dissimilarity between fog and cloud-model computation is geographically distributed. Fog computing is widely distributed and can provide various services in a geographically wide-spread manner. For example, the high-quality streaming services, which serve vehicles in motion, are delivered by employing fog computing through access points, routers, and proxies that are geographically sited along highways and roads.
- *Interoperability and federation*: The consistent support of particular services (e.g., real-time streaming services) needs various providers' cooperation. On the other hand, fog computing components have to be capable of interoperating, and services have to be federated among domains.
- *Agility and scalability of federated, fog-node clusters*: Fog computing is essentially *adaptive*. This adaptation means that fog is applicable to the sub-level of fog, including cluster and sub-cluster levels (cluster-of-clusters). Fog computing can also be adaptive with changes in the resource pool, combinations of computation, network condition variations, and data-load modifications.
- *Real-time interactions*: Fog computing applications deal with real-time responses, not those processes that are batched.

2.2.1 Fog architecture

As an exceedingly distributed paradigm, fog computing enables us to extend networking and computing power into the edge of the network and ubiquitously access the resources. Consisting of fog nodes (virtual or physical) that reside between centralized (cloud) services and smart end-devices, the model eases the deployment of distributed, latency-aware services and applications [36, 96]. The proposed architecture, as one can see in Fig. 1, is made up of three layers: cloud layer, fog layer, and IoT or user device

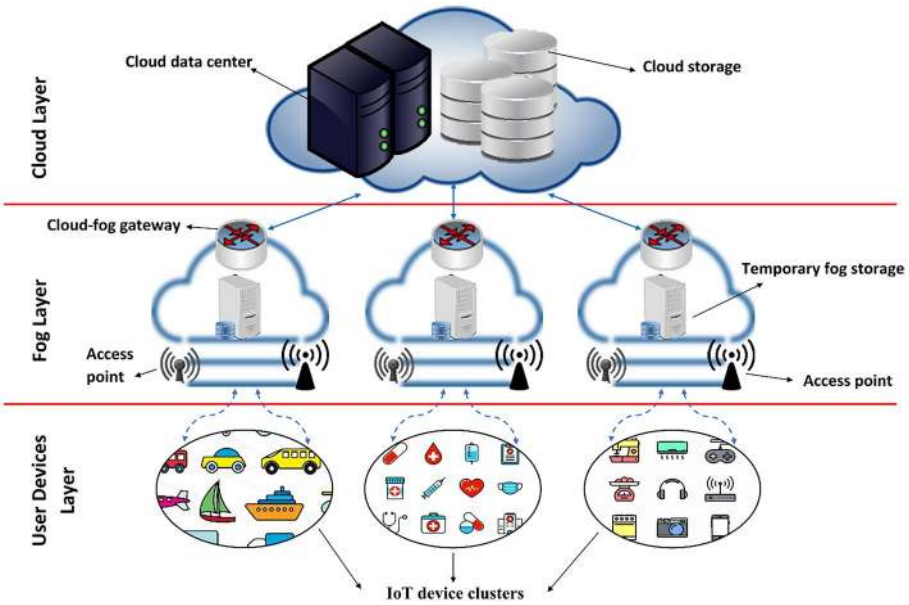


Fig. 1 Fog architectural paradigm [80, 81]

layer. The architecture provides federated and integrated cloud services to IoT or users' devices through a mediated layer fog. As presented in Fig. 1, the fog is made up of fog nodes such as gateways, routers, edge servers, base stations, and local access points. In the fog architecture, the entire fog deployment can be sited locally. This means that automation or an organization with a single office complex can host the fog services, or even the fog services can be allocated at a regional or partial area that sends data to the centralized services and systems. In the architecture, each active fog node is autonomous and self-diagnosis to certify uninterrupted processes of the facility service [113]. Each layer is introduced as follows:

- *Cloud layer*: As a computing strategy that is based on the Internet, cloud computing offers on-demand and ubiquitous access to a cluster of resources such as processors, services, storage, and applications, in other devices or computers. Despite the fact that the cloud computing strategy can handle a large amount of data from user's device bundles, transferring a huge amount of data to and from the servers that are embedded at the cloud layer is a challenging issue since the bandwidth is restricted. Therefore, it is necessary to compute the data close to the sources, and fog computing can appropriately address this challenge.
- *Fog layer*: As an architecture that distributes computation geographically, fog computing has a resource center that contains one or more heterogeneous devices. These devices are ubiquitously connected and are not merely supported by cloud services to furnish elastic computing, storage, and Internet-based communication collaboratively in isolated environments to a wide range of clients near [133]. Fog computing brings networking resources close to the underlying networks, as a network between the cloud(s) and the underlying network(s). The traditional cloud computing strategy is extended to the edge of the network by fog computing, enabling us to create better and refined services or applications. For wireless sensor networks and IoTs, fog is a micro-data center strategy and how computing occurs at the edge [1].
- *User device Layer*: Each fog node in fog computing is often required to procure computation services to different user's devices. Hence, the computation resources that are allocated for each user's device are restricted. Likewise, each user's device, in its vicinity, might have several different fog nodes. The user's device, to accelerate the computation of its tasks, might offload its tasks to several fog nodes. In face identification, for example, the burden of task delivery moved to fog nodes [40], which still requires many more computation resources. We can divide this task into some smaller tasks and execute them in parallel. In order to accelerate the execution, the IoT device makes efforts to offload the small tasks to several different fog nodes.

Based on the fog computing architecture, a fog node as a network component is located in the mediate layer between the cloud and the user devices. It is connected with devices and users, cloud, and other fog nodes. The connections are of three kinds: (1) wired/wireless connections between fog nodes, (2) wireless connections between users' devices and fog, and (3) wireless/wired connections between the cloud centers and fog nodes. These three kinds of wireless connections can be within mobile networks such as 3G and 4G, or other networks such as Wi-Fi, ZigBee, Wireless Local Area Networks, and Bluetooth, which support the fog applications [41].

2.2.2 Fog node

Fog nodes, as fog computing architecture core components, are either virtual (e.g., virtualized switches, cloudlets, and machines) or physical (e.g., gateways, routers, switches, and servers). They are closely coupled with access networks and the smart end-devices and procure computing resources to the devices mentioned. A fog node is conscious of its logical location and geographical distribution. In fog computing architecture, fog nodes between user devices and resources provide some forms of communication services and data management if required. In order to have a given fog computing capability deployed, fog nodes can function in a decentralized or centralized manner, can be configured as single fog nodes communicating among them for service delivery. They can also be united to form clusters to provide horizontal scalability over disperse geolocations via extension or mirroring mechanisms [43, 70]. To deploy the capability of fog computing quickly, fog nodes should support one or more of the features below [43, 114]:

- *Manageability:* Complex systems, being capable of performing most routine operations automatically, manage, and orchestrate fog nodes.
- *Hierarchical clustering:* Hierarchical structures are supported by fog nodes with various layers that create diverse subclasses of service functions while operating as a continuum.
- *Heterogeneity:* Fog nodes, coming in various form factors, can be applied to different environments.
- *Autonomy:* Fog nodes can operate independently to make local decisions at the node or cluster-of-nodes level.
- *Programmability:* Fog nodes can inherently be programmed at different levels by different stakeholders-like network operators, equipment providers, domain experts, or end-users.

2.2.3 Applications and services of fog

The installation of fog infrastructure can enhance some network services' quality by improving computation, optimizing energy, and allocating resources. In the following, the applications of fog are discussed briefly.

- *Transportation services:* Fog computing offers higher flexibility and reliability in transportation systems implementation than using cloud computing and cellular networks. In urban areas, the movement of traffic is slow. Slow-moving traffic gives the advantage of using some types of communication other than cloud and cellular networks. Distributed networks of fog nodes can be used to provide Wi-Fi connectivity to all agents of transportation systems. Moreover, fog nodes can also serve as intermediate level in communication. To reduce the size of data, fog nodes execute analysis operating and filtering on the communicated data. This computing method enables us to preserve the bandwidth of the network and reduce latency [80].
- *Healthcare systems*: Fog computing enables healthcare systems to make smart and quick decisions during an emergency for time-critical healthcare applications (e.g., COVID-19 diagnosis 111). It also helps protect sensitive data with reduced response time compared to other computing methods such as the cloud [6138, 3].

- *Video streaming*: In fog computing, a video streaming application allows mobile users to watch the recent video available on the screen. Due to the higher quality of live video streams, instead of transferring data to cloud centers, the data is sent to the nearest fog node. In a fog network, mobile devices such as smartphones and laptops can perform as a fog server, which runs a tracing method and processes raw video streams [97].

- *Smart Grid:* Smart grid is a new way to distribute the network of electric power. Smart grids include substations, transmission lines, and transformers. The smart grid can provide a distribution of power in which service providers and customers can do real-time control and monitoring of the price, production, and power consumption. Fog nodes are used to collect, filter, and process information about the power grid locally [97].
- In addition to the applications mentioned above, we can mention other applications such as smart homes [96], smart cities [114], online games [92], and smart industries [20].

2.3 Offloading

By using the offloading strategy, computations and data can be transferred from the resource-limited user devices to resource-rich fog/cloud nodes. With this strategy, such goals as load balancing and system performance improvement can be achieved [55]. Five dimensions for offloading can be introduced as follows [2, 121, 122]:

- *Offloading destination*: For designing and sketching an offloading algorithm, in the beginning, it is crucial to select fog or cloud servers carefully and precisely. IoT devices' tasks and workload at the run time must be offloaded to only one fog-could server for consecutive and frequent execution or to several different servers for parallel execution, leading to lower response latency.
- *Balance in offloading*: In some cases, the fog or cloud may not have access to available resources sufficiently, making it hard to satisfy the service level agreements of the application. In other words, some users' requests and application tasks may be rejected. In these cases, the uncertainty of the user's request satisfaction and the weakness of offloading methods may cause the workloads to become unbalanced among fog-cloud servers. Then, some parts of cloud-fog servers might have low loads, whereas other servers are almost bustling. To solve this problem, a controller, which can work online, can distribute and organize the tasks. On the other hand, the controller, which can work offline, can move the workloads and tasks to fog-cloud servers in proportion to the usage of resources.
- *Mobility of IoT devices*: Since an IoT user might move in various service areas, this device might quit the zone under which original network servers provide service, and this situation will create two challenges: First of all, we must make a decision whether the fog-cloud service should move out from the previous server to a new server to preserve the efficient communication and connection. The decision to transfer from one network service to another should decide over a tradeoff between migration cost and long-distance connection cost. Secondly, in several networks such as Wi-Fi or the new generation of mobile networks such as 3G/4G, the signal might be influenced by massive objects. These environments involve heterogeneous networks, data transfers, and the smart device connection protocols, specifically in the overlapped service zones.

There is the possibility of intermittent connection, and persistent connectivity cannot be guaranteed. We have different approaches in the literature to deal with this problem.

- *Application partitioning*: Due to the fact that the capacity of the battery cannot keep up with the fast progress in application technologies and programs like virtual reality, it is not possible to manage all parts of these applications entirely on the IoT device. Thus, when designing the algorithm of offloading, the organization and division of partitioned components are the primary basis. Therefore, the computation partitioning algorithms are followed and studied to define which elements of the users' applications are offloaded and in which order they must be performed.
- *Partition granularity*: Before the tasks and requests are offloaded to the fog-cloud servers, we should also address the size of components that could be executed from a distance. Since different developers design the applications with customized functional features, partition granularity is an essential parameter in improving the global quality of execution experience. The partition granularity is introduced as a different size of offloading ingredients.

Figure 2 shows the offloading process in fog computing between the user device, fog, and cloud layers. As we can see in Fig. 2, transferring the resource-intensive tasks to an external platform like a fog, a cluster, a grid, or a cloud is called computation offloading.

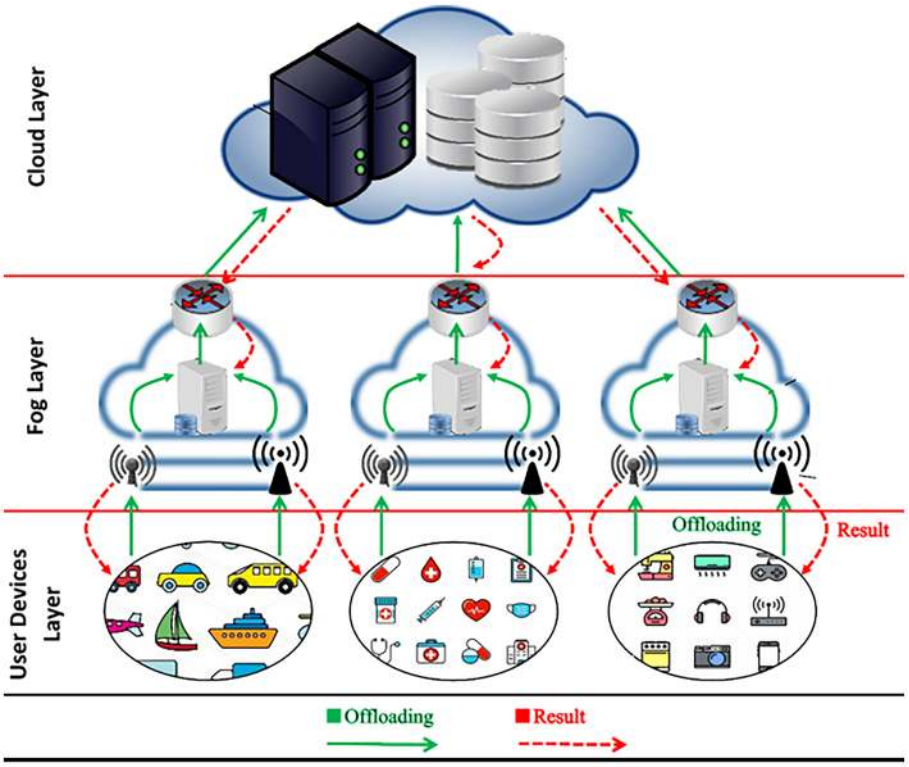


Fig. 2 Offloading process in fog computing [2, 34]

Because of the hardware limitations of devices such as storage, computational power, and dedicated bandwidth, offloading seems to be inevitable. Offloading is applied in edge computing, mobile cloud computing, and fog computing. The resource-intensive tasks might be for virus scanning, searching, image processing, computational decision making, and artificial intelligence [42].

Computation offloading can be defined as the task of sending computing-intensive application components to a remote server. These days, some computation offloading methods are outlined with several mobile application methods. However, computation offloading strategies are still in trouble with several issues. A huge research body exists on offloading decisions to save energy and improve performance, and offloading seems to be a great solution in satisfying the requirements of response time on mobile systems as applications get complicated day after day [11, 42].

The aims of offloading may be manifold. Satisfying real-time limitations might be another goal of offloading. Applications that execute various combinations of real-time computations consume vast amounts of power. Hence, offloading, which need power-intensive computations, can overcome real-time limitations to resources that have more capacity by transferring the tasks. Another offloading goal might be energy-saving. Energy is the main limitation in some services such as mobile systems. Smartphones are not just applied for voice communications anymore; but they are applied for surfing the web, watching videos, playing games, and other goals as well. Therefore, mobile systems are more power-consuming, and battery life is becoming shorter. Despite the continuous improvement of battery technology, it has not yet been capable of catching up with mobile phones' ever-increasing power consumption. By sending the energy-intensive parts from user devices towards servers, offloading might extend the life of the battery [60]. To improve the storage, quality of applications, and battery lifetime, computation offloading can deal with resource restrictions. Current studies in computation offloading for cloud/fog computing in the mobile network can be categorized into six subsets [135].

2.3.1 Offloading objectives

The offloading mechanisms span over several goals while attempting to reduce energy consumption, considering the availability of fog resources, computation gain, and communication costs [21, 69, 121].

- *Communication cost and profiling:* Considering the distributed Fog computing environment nature, communication characteristics that deal with data transmission cannot be ignored by energy-aware offloading. Availability and communication are closely related to each other when it comes to IoT device connectivity over networks. When availability is attained, the communication cost becomes a critical, delimiting factor in the quantification of the real benefits of fog computing offloading.
- *Energy Efficiency*: In offloading, high power consumption is a significant problem. However, the energy consumed in the offloading process in fog environments will be less than that consumed in other methods of computations. An energy-saving offloading technique may be more attractive to particular networks such as mobile and cellular networks.
- *Computation efficiency*: The extension of IoT device computation capabilities includes a primary fog computing functionality in which the cost of communication cannot be ignored because of its inherent presence in any IoT environments. Aiming to increase

the processing of applications and preserve IoT devices' energy, offloading schemes have made use of task sub-partitioning, cloud/fog resource limitations, and schedule based on task dependencies.

2.3.2 Offloading metrics

Some metrics are required to evaluate the offloading mechanisms. They are called qualitative metrics. Papers use different qualitative metrics, such as energy, latency, utilization, and throughput. The most prominent qualitative metrics for offloading in fog computing are described as follows:

- *Energy consumption*: The amount of energy consumed in the network is called energy consumption, which can be reduced by an effective offloading mechanism [96].
- *Execution time*: The time duration that is needed for a program to run is called execution time. It includes routing time, migration time, and re-association time [37].
- *Cost*: A specific amount of money to command the action is called cost [8].
- *Resource utilization*: The degree to which the network resources are utilized, including bandwidth, link, memory utilization, and processor, is referred to as resource utilization [25].
- *Response time*: The interval starting from a request or a job acceptance to an appeal or task response for the server is referred to as response time [91].
- *Throughput*: The data quantity that has been delivered from one spot to another during a specified period is called throughput [37].
- *Scalability*: A key feature of a system is to manage a colossal work by adding resources to the system [15]. This feature is a specific property for networks, networking protocols, computers, algorithms, applications, and programs.
- *Security*: *Security* is any action that is designed for this purpose so as to preserve the integrity and usability of the data and network, [71], employed in both platforms of hardware and software technologies.
- *Performance*: This feature as a quantitative and qualitative process reflects the collective network statistics and defines and measures an individual network's level of performance [35].
- *Availability*: Availability is defined as a capability that enables authorized users to gain access to the information. This term represents a user's ability to access resources or information in a certain location [17].
- *Confidentiality*: The protection of information against unauthorized parties is called confidentiality. Only authorized people can access sensitive data. A confidentiality violation refers to a situation where an unauthorized person has access to confidential information [90].
- *Jitter*: This term is the alteration in milliseconds' latencies between data packets over a network [127].

3 Research methodology

This section introduces a systematic review of offloading mechanisms in fog computing to show a vivid portrait of fog computing's offloading mechanisms. The purpose of a systematic review is to give an exhaustive and comprehensive summary of the current

and recent relevant literature. In contrast to a non-structured review process, a systematic review reduces bias and follows a rigorous and precise sequence of methodological stages to research literature. Systematic reviews rely on evaluated and accurate review methods to extract, analyze, and document results [13]. The first step is to search relevant papers in this domain. In the research methodology section, we will prepare a list of all the citation indexes and academic databases searched, such as ScienceDirect, ACM, IEEE, Taylor and Francis, Wiley, and Springer [14, 22, 57]. The titles and abstracts of the selected articles were scanned for their relevance and importance to our study. This article collection is directly related to the research questions. In Section 3.1, we formalize the research questions, and in Section 3.2, the method of paper selection is discussed.

3.1 Question formalization

Our research goal is to detect, collect, and explore the most related concerns and challenges for our subject, including architecture, technologies, and applications. The research questions (RQs) that are addressed by this study are:

- RQ1: What are the main practical motivations behind offloading mechanisms in fog computing?
- RQ2: Which offloading mechanisms are used in the fog system?
- RQ3: What are the evaluation metrics generally applied for offloading mechanisms in the fog system?
- RQ4: What evaluation methods are used to measure the offloading mechanisms in the fog system?
- RQ5: Which kinds of algorithms are applied in offloading mechanisms in fog computing?
- RQ6: What are the challenges, open issues, and future trends in offloading mechanisms in fog computing?

3.2 Paper selection process

This part provides the search and selection process in three stages, which are described as follows. Table 2 indicates the filtration criteria in these stages.

Stage 1: Automated search based on keywords and titles.

Stage 2: Initial selection based on the abstract, conclusion, and our inclusion/exclusion criteria.

Table 2 Filtration of automated search paper by inclusion/exclusion criteria

Inclusion	The papers proposing evaluations, experiences, or solutions of offloading mechanisms in fog computing
	JCR-indexed journal papers
	Papers published between 2016 and January-2020
Exclusion	Review and survey papers, conference papers, theses, books, and book chapters
	Studies not focusing on offloading mechanisms in fog computing
	Non-English scripts and non-peer-reviewed papers
	Short papers (less than six papers)

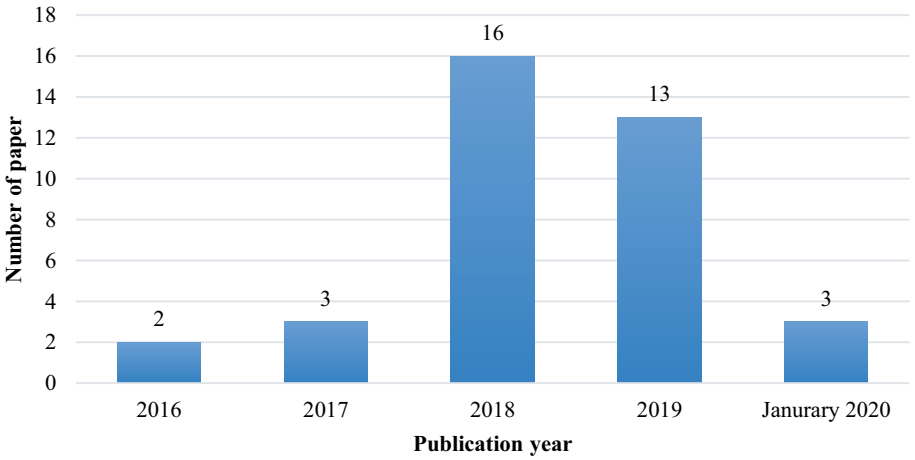
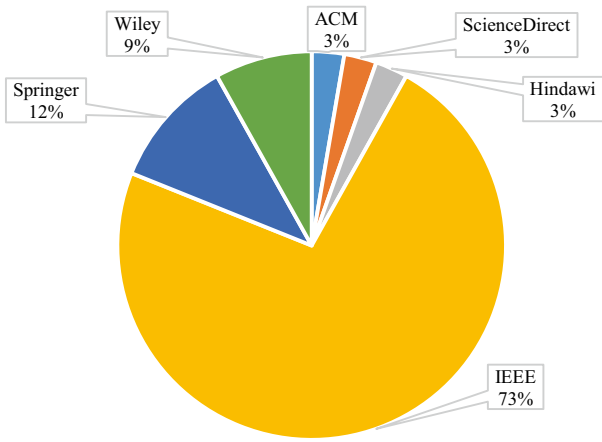


Fig. 3 The number of papers based on Stage 3

Fig. 4 Percentage of published papers in any publications based on Stage 3



Stage 3: Further selection based on the validity and reputation of the journals and full text.

In *Stage 1*, we explored the search string of *fog <AND> (offloading <OR> offload)* through Google Scholar as the main search engine based on popular academic publishers such as ACM, ScienceDirect, Hindawi, IEEE, Inderscience, SAGE, Springer, Taylor and Francis, and Wiley. Our findings' initial result between 2016 and January 2020 was 131 papers, including conference papers, journals, and book chapters.

In *Stage 2*, we selected 62 papers from Stage 1. We chose a set of criteria to include relevant papers and exclude irrelevant ones. Review and conference papers, books and book chapters, short papers (less than six papers), and irrelevant papers were subsequently excluded.

By searching in the JCR-indexed journal papers, in *Stage 3*, the search chain was confined by the provided experiential and realistic results. We finally selected 37 papers based on the full text.

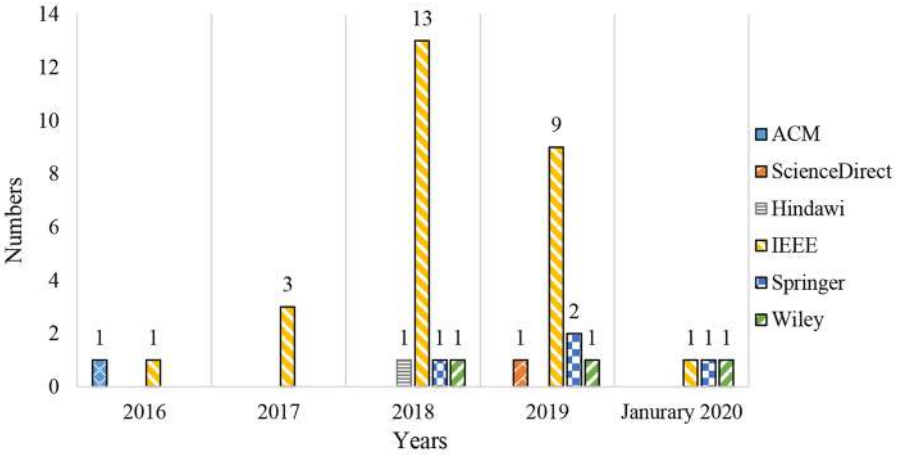


Fig. 5 The number of papers based on publishers as mentioned in Stage 3

Figure 3 shows that a huge number of journal papers were published in 2018. As shown in Fig. 4, we found that 73% of the papers were related to IEEE, 12% to Springer, 9% to Wiley, 3% to ScienceDirect, 3% to Hindawi, and 3% to ACM journal papers. Figure 5 presents the top rates of published journal papers concerning IEEE. Table 3 shows the details of the selected papers that used specific approaches. All of the proposed approaches are classified into four categories, including computation-base offloading, energy-based offloading, storage-base offloading, and hybrid offloading.

4 Classification of the fog offloading approaches

The classification of the related studies is presented in this section. The studies on fog computing are vast and diverse, so categorizing the relevant studies in a systematic way is not a trivial, unchallenging task. We outline the scheme of the suggested classification. We present four main categories: Computation-based, energy-based, storage-based, and hybrid mechanisms. Since most scholars address these approaches' issues, the literature review from these four categories is distinctly natural and the best choice. It enables us to classify the reviewed works under ordinary umbrellas. Although one of the primary purposes of this systematic study is to present the most practical taxonomy, other classifications might be feasible. We will discuss and describe the entire classification and every category in detail in Sections 4.1, 4.2, 4.3, 4.4. Based on the following guidelines [22, 58] and read the full text of each article, at the end of each section, we will summarize the advantages and disadvantages of selected papers in Tables 4, 6, 8, and 10. Moreover, the existing evaluation metrics of selected papers are compared in Tables 5, 7, 9, and 11. The evaluation metrics listed in these tables stem from parameters that selected papers evaluated in their (joint) optimization problem formulas and models.

Table 3 Details of selected papers

Category	Publisher	Year	Author(s)	Journal/conference name
Computation-based	ACM	2016	Fricker et al. [33]	ACM transactions on modeling and performance evaluation of computing systems (TOMPECS)
	IEEE	2016	Liang et al. [64]	China communications
		2017	Liu et al. [66]	IEEE internet of things journal
		2018	Wang et al. [119]	IEEE transactions on industrial informatics
		2018	Du et al. [31]	IEEE transactions on communications
		2018	Shah-Mansouri and Wong [103]	IEEE internet of things journal
		2018	Liu et al. [67]	IEEE transactions on vehicular technology
		2018	Jiang and Tsang [44]	IEEE internet of things journal
		2018	Ruan et al. [99]	Journal of communications and networks
		2019	Li et al. [62]	IEEE access
		2019	Wu et al. [128]	IEEE access
	Springer	2019	Zhang et al. [139]	Peer-to-peer networking and applications
	Wiley	2019	Rabie et al. [94]	Transactions on emerging telecommunications technologies
		2020	Wang and Chen [118]	Transactions on emerging telecommunications technologies
Energy-based	IEEE	2018	Jiang et al. [45]	IEEE systems journal
		2018	Wei and Jiang [125]	IEEE access
		2018	Zhang et al. [138]	IEEE internet of things journal
		2019	Li et al. [63]	China communications
		2019	Chen et al. [27]	Transactions on green communications and networking
	Wiley	2018	Vu et al. [117]	Transactions on emerging telecommunications technologies
Storage-based	Hindawi	2018	Quinton and Aboutorab [93]	Wireless communications and mobile computing
	IEEE	2018	Wang et al. [120]	IEEE internet of things journal
		2018	Chiti et al. [28]	IEEE internet of things journal
		2019	Shnaiwer et al. [107]	IEEE access

Table 3 (continued)

Category	Publisher	Year	Author(s)	Journal/conference name
Hybrid	IEEE	2017	Meng et al. [73]	IEEE access
		2017	Zhu et al. [141]	China communications
		2018	Yousefpour et al. [136]	IEEE internet of things journal
		2018	Liu et al. [68]	IEEE internet of things journal
		2019	Wang et al. [123]	IEEE transactions on industrial informatics
		2019	Wang et al. [123]	IEEE access
		2019	Misra and Saha [76]	IEEE journal on selected areas in communications
		2019	Adhikari et al. [4]	IEEE internet of things journal
		2020	Cai et al. [23]	IEEE internet of things journal
	Science direct	2019	Zaharia et al. [137]	Simulation modelling practice and theory
	Springer	2018	Mukherjee et al. [79]	The journal of supercomputing
		2019	Rahbari and Nickray [95]	Peer-to-peer networking and applications
		2020	Balasubramanian and Meyyappan [12]	Computing in engineering and technology

Table 4 A comparison of properties in computation-based offloading mechanisms

Paper	Main idea	Advantage	Disadvantage
[33]	Offloading the requests blocked at the data center	Low response time High performance of the system	Low security High cost High response time
[64]	Resource allocation for fog radio access networks	Low energy Numerical results High resource utilization	Low security
[66]	Utilizing queuing theory in fog computing	Low energy High scalability	Low availability
[119]	Designing an offloading algorithm on the Internet of vehicle	Real-time traffic management Optimization problem Resource utilization Low cost	High response time Low availability
[31]	Computation offloading in fog and cloud computing	Optimization of transmit energy Optimization of offloading Energy efficiency	Low security High response time
[103]	Allocation of fog computing resources to the IoT users	Low cost Low response time Optimizing performance Low energy	Low throughput Low security
[67]	Computation offloading with non-orthogonal multiple access	Optimized resource allocation Energy efficiency Reduce cost	Low scalability Low availability
[44]	Delay-aware task offloading in shared fog networks	Low response time Low cost Low energy Numerical results	Low scalability Low availability
[99]	Resource allocation in fog environment	Numerical results Low response time Reduce energy	Low scalability Low security Low availability

Table 4 (continued)

Paper	Main idea	Advantage	Disadvantage
[62]	Resource allocation balance in the framework for heterogeneous real-time tasks	Low energy Resource allocation balance and throughput	Low security Low scalability
[128]	Task offloading in fog and cloud computing	Low energy Optimizing performance Numerical results Low response time	Low scalability Low security Low availability
[139]	Emergency-level-based healthcare information offloading	Low response time	High cost Low efficiency Low security
[94]	optimization of computation cost and delay for offloading in fog	Low response time Low cost Numerical results	Low scalability Low security Low availability
[118]	Minimizing latency and resource allocation for IoT networks in fog	Low response time Optimized resource allocation Numerical results Reduce energy Optimizing performance	Low scalability Low security Low availability High cost

Table 5 Comparison of the existing evaluation metrics in the computation-based approaches

Paper	Cost	Energy	Response time	Performance	Resource utilization	Security	Throughput
[33]	✗	✗	✗	✓	✗	✗	✗
[64]	✗	✓	✗	✗	✗	✗	✗
[66]	✓	✓	✗	✗	✓	✗	✗
[119]	✓	✓	✗	✓	✗	✓	✗
[31]	✓	✓	✗	✗	✗	✗	✗
[103]	✓	✓	✓	✗	✗	✗	✗
[67]	✓	✓	✓	✗	✗	✗	✓
[44]	✓	✓	✓	✗	✓	✗	✗
[99]	✓	✓	✓	✗	✓	✓	✗
[62]	✓	✓	✓	✗	✓	✗	✓
[128]	✓	✓	✓	✓	✓	✗	✗
[139]	✗	✗	✓	✗	✓	✗	✗
[94]	✓	✗	✓	✓	✓	✗	✗
[118]	✗	✓	✓	✓	✓	✗	✗

Table 6 A comparison of properties in energy-based offloading mechanisms

Paper	Main idea	Advantage	Disadvantage
[45]	Offloading strategy for fog computing	Low energy Low response time Ensures QoS Decreased bandwidth	Low security
[125]	Improving the offloading efficiency	Low energy Low response time Low cost	Low scalability
[138]	Fairness scheduling metric for fog offloading	Low energy Low response time	Low scalability
[63]	Optimizing the computation offloading for the Internet of everything	Low energy Low cost	Low scalability
[27]	Energy-optimal dynamic offloading in fog	Low energy Low response time	Low scalability
[117]	Providing a scheme for upstream IoT offloading services in fog radio access networks	Low energy Low response time	Low scalability Low availability

Table 7 Comparison of the existing evaluation metrics in the energy-based approaches

Paper	Cost	Energy	Response time	Resource utilization	Scalability	Security	Throughput
[45]	✓	✓	✓	✗	✓	✗	✗
[125]	✓	✓	✓	✗	✗	✗	✗
[138]	✗	✓	✓	✓	✗	✓	✓
[63]	✓	✓	✓	✓	✗	✗	✗
[27]	✗	✓	✓	✗	✗	✗	✗
[117]	✓	✓	✓	✗	✗	✗	✗

Table 8 A comparison of properties in storage-based offloading mechanisms

Paper	Main idea	Advantage	Disadvantage
[93]	Efficient transmission schemes to offload	Low response time	Low scalability
[120]	Designing an architecture for computation and storage offloading	Low response time Low cost High feasibility High availability	Low scalability
[28]	Providing an efficient strategy to offload	High flexibility Low energy Low cost Numerical results	Low availability Low accuracy Low security
[107]	Offloading performance of fog radio access networks	Low response time Low cost	Low security

Table 9 Comparison of the existing evaluation metrics in the storage based approaches

Paper	Cost	Energy	Resource utilization	Response time	Scalability	Throughput
[93]	✗	✗		✓	✗	✓
[120]	✓	✓	✗	✓	✗	✗
[28]	✓	✓	✓	✓	✓	✗
[107]	✓	✗	✗	✓	✗	✓

4.1 Computation-based approaches

To increase system performance, some proposed approaches focus on the optimization of computation offloading. We classify the research studies, which used this approach under the computation-based category. First, we will describe the computation-based approach for offloading in fog computing. Fourteen selected papers are surveyed in Section 4.1.1. Then in Section 4.1.2, their advantages, disadvantages, main ideas, and evaluation methods will be explored and compared.

4.1.1 Overview of the selected computation-based approaches

For data centers in fog-based networks, Fricker et al. [33] proposed a low-time and high-performance computational offloading technique. They considered a scenario in which if an overloaded data center received a request, then forwarded it to a near data center. The purpose of the solution was to estimate the gains achieved through the collaboration between near data centers. Without any significant blocking at the big centers, the proposed technique could considerably improve the small data centers' clocking problem. However, the authors only concentrated on performance issues without considering security and scalability.

For fog radio access networks, Liang et al. [64] suggested an offloading service with coordinated computation and joint resource allocation. In centralized F-RANs, authors studied the coordinated offloading strategy and resource allocation policies and focused on

Table 10 A comparison of properties in hybrid offloading mechanisms

Paper	Main idea	Advantage	Disadvantage
[73]	Solving the hybrid computation offloading problem	Low energy Numerical results	Low scalability Low availability
[141]	Fog computing model and offloading policy	Low response time Low energy Low response time	Low availability Low capacity Low security
[136]	Providing a framework to reduce IoT service delay	Low response time Low energy Low cost Numerical results Availability	Low security Non-optimized resource utilization
[68]	Socially-aware dynamic computation offloading scheme	Optimization of the resource utilization Low energy High scalability	Low security High cost
[123]	Edge offloading in fog computing	Low energy Low response time Low cost	Low scalability Low security
[123]	Providing algorithms considering mobility	High performance of the system Low energy Low response time Optimized bandwidth	Low security Low scalability Low accuracy
[76]	Task offloading scheme	Low energy Low response time	Low security
[4]	Offloading for fog through swarm optimization	Low response time Low cost Better resource utilization	Low security Low scalability Low accuracy
[23]	Joint offloading of tasks and energy in the fog	Low response time Low cost Low energy Optimized availability High bandwidth Optimized capacity	Low security Low scalability Non-optimized resource utilization

Table 10 (continued)

Paper	Main idea	Advantage	Disadvantage
[137]	Machine learning-based offloading in fog	Low response time Low cost Low energy High throughput	Low security Low scalability Low accuracy
[79]	Cooperative offloading approach for indoor mobile cloud network	Low energy Low Jitter	Non-optimized resource utilization High response time
[95]	Providing an algorithm for decision parameters for selecting the best fog devices	Low energy Low response time Best cost Resource utilization	Non-optimized resource utilization Low bandwidth
[12]	Game theory-based offloading for the cloud of things in the fog	Low energy Low response time Resource utilization Optimized availability	Low security Low scalability High cost Low bandwidth

Table 11 Comparison of the existing evaluation metrics in the hybrid

Paper	Authentication	Availability	Cost	Confidentiality	Energy	Integrity	Jitter	Response time	Resource utilization	Throughput
[73]	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗
[141]	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
[136]	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
[68]	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓
[123]	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
[123]	✗	✗	✓	✗	✓	✗	✗	✓	✓	✓
[76]	✗	✗	✓	✗	✓	✗	✗	✓	✗	✗
[4]	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗
[23]	✗	✓	✓	✗	✓	✗	✗	✓	✗	✗
[137]	✗	✗	✓	✗	✓	✗	✗	✓	✗	✓
[79]	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓
[95]	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗
[12]	✗	✓	✗	✗	✓	✗	✗	✓	✓	✗

energy consumption. In this paper, the optimal challenge was formulated and outlined as minimizing the sum of the energy consumption. In comparison with traditional cloud computing, the numerical results showed that the suggested algorithm could warrant response time tolerance while decreasing energy consumption. However, the method was not implemented in a real large-scale network.

In a fog computing system, Liu et al. [66] suggested the queuing theory for an exhaustive study on the cost of computation offloading, response time, and energy consumption. Moreover, to reduce the cost jointly, response time performance, and energy consumption, the introduced architecture optimized the transmission power and probability of offloading for mobile devices. By finding the optimal point for transmitting power and offloading probability for each device and minimizing the cost jointly, response time performance, and energy consumption, a multi-objective optimization challenge was outlined and formulated with a joint objective. However, the primary flaw of this paper is that the researchers did not mention anything about availability, security, and resource allocation in this architecture.

A computation offloading algorithm was proposed by Wang et al. [119] for management of the real-time traffic in the fog-based Internet of vehicle systems to reduce the traffic management server average response time for messages. In this model, by queuing theory and mathematical formulation, an optimization problem was formulated by moving and parked vehicle-based fog nodes for the fog-enabled offloading problem; the authors regarded the management of traffic and roads' safety.

Du et al. [31] suggested the resource allocation approach and computation offloading, which were first considered for offloading decision making. To solve the optimization issue, they proposed a low-complexity algorithm, where the offloading decisions were gained through randomization and semi-definite relaxation. The resource allocation was gained by applying Lagrangian dual decomposition and fractional programming theory. According to this paper, the proposed model had higher performance, optimization of transmit power, bandwidth, and slower response time than other mentioned classification algorithms. However, the case user equipment was not regarded.

For the resources used in fog-based IoT systems, Shah-Mansouri and Wong [103] proposed an allocation mechanism. The aim of the paper was to pick a computation offloading decision for each task received by the users' devices, where each user attempted to maximize the performance quality of applications. However, the paper was not developed as an online strategy to allocate resources and did not consider the dynamic arrival of computation tasks.

Liu et al. [67] concentrated on the issue of computation offloading and suggested an integrated cloud and fog computing managing resources. Aiming to decrease the response time of the users' devices, total cost, and energy of the system, the authors proposed some strategies for scheduling, offloading, and resource allocation as an optimization problem. Nevertheless, the authors, to improve the utilization of resources and optimize the users' experiences, ignored the combination of content caching and efficient computation offloading.

Jiang and Tsang [44] investigated the allocation of resources for multiple heterogeneous tasks from the end-users in a shared fog system. They aimed to create a mathematical model for computational task offloading in a shared fog network to capture the fog system's required and distinct features with exact influential solution techniques. They also targeted a beneficial and practical scheme in order to schedule tasks with heterogeneous delay sensitivities in a shared fog system. The numerical results demonstrated that this model had low power, cost, and response time. The authors did not pay

attention to more complex issues like the dynamic sizing of the provisioned fog facility resources.

Ruan et al. [99] formulated F-RAN architecture with a shared pattern-oriented approach. In this architecture, the optimization issue reduced energy consumption and the development of an algorithm in which the online computational task was offloaded. Furthermore, this architecture provided a full-process, exhaustive, and systematic solution for shared services quality, which was promoted with F-RAN. The numerical results showed that the constructed architecture had low energy and power and proved that the proposed strategy was well-organized and efficient. However, the paper did not make an attempt to improve the reliability, security, and safety of fog-based networks.

For heterogeneous real-time tasks, Li et al. [62] introduced a framework in which computation task offloading issues and resource allocation balance were considered. To optimize and improve the task accomplishment ratio, throughput, balance, and resource allocation, three parallel algorithms, including buffering, resource allocation, and offloading, were considered in this framework. This paper attempted to optimize the response time, energy consumption, and cost of the offloading processes under queuing. However, the authors did not evaluate this scheme as a scalability aspect.

Wu et al. [128] explored the optimal computation offloading strategy to regard the occupied vehicle departure and tried to formulate the offloading problem in the vehicular fog and cloud computing system as a semi-Markov decision process model with the defined action space, state space, transition probabilities, and the discounted reward model. To increase the long-term reward of the mechanism and solve the problem of finding the optimal task offloading scheme, the authors presented an algorithm that was a relative value. However, the parameters that vehicles frequently received by the system were not considered.

A simulated annealing computation offloading algorithm was suggested by Zhang et al. [139] to make a policy of node access allocation for the users, according to healthcare information types, to be able to receive the fastest response. Based on the simulation results, an algorithm of emergency-based measures could help the patients find the quickest communication way to the professional medical units. However, the authors only regarded traffic offloading rather than other offloading factors to decrease the delay time in the medical environment. For instance, the information was not sent off from just a single Hospital.

Rabie et al. [94] regarded the cost and delay and studied the problem of resource allocation to multi-vehicles users. They referred to a hypothesis of a VEC (vehicle edge computation) model in a Vehicular Fog Network (VeFN) and suggested a joint optimization challenge for the system cost and offloading response time in this scheme. To assign computing tasks, in VeFN, to vehicles on the road, the authors used an offloading strategy, which was designed with two layers. In this method, firstly, according to the location of each cell, the vehicles were grouped. Then, the members of the groups produced dynamic changes, and due to energy consumption, some vehicles avoided providing computing services. However, this paper did not focus on more real and wide-ranging circumstances such as the changes in speed and acceleration of vehicles, more complex pathways, and the change in directions of vehicles.

For the IoT networks in a fog environment, Wang and Chen [118] developed a resource allocation strategy and an offloading decision in which the latency was minimized. They formulated a joint optimization issue that the parameters were the offloading decision, computation capability locally, and a fog node's computing resource allocation. Yet, the authors neither explored the model of multiuser multitask nor assessed the applicability

and efficiency of study schemes in applications that were executed in the real-world, e.g., the applications that were performed in real industrial networks.

4.1.2 Summary of computation-based approaches

A side-by-side comparison of the selected methods in the computation-based category, as well as their most important advantages and disadvantages, are shown in Table 4. Some of the advantages are quick response time, high resource utilization, energy efficiency, balanced resource allocation, and high throughput. On the other hand, some of the disadvantages are low security and low availability. The selected papers are evaluated for qualitative metrics. Ten metrics are detected, and the results are accordingly depicted in Table 9.

4.2 Energy-based approaches

Some of the proposed approaches focus on the optimization of energy in offloading. We classify the research studies, which use this approach under the energy-based category. First, we will describe the energy-based approach for offloading in fog computing. Six selected papers are reviewed in Section 4.2.1. Then in Section 4.2.2, their advantages, disadvantages, main ideas, and evaluation methods will be explored and compared.

4.2.1 Overview of the selected energy-based approaches

Jiang et al. [45] suggested an energy-efficient offloading decision mechanism to reduce energy consumption and, at the same time, to meet the response time limitations. A schedule-delay aware offloading strategy was developed in this paper to ensure the service quality in real-time tasks and decrease the energy consumption of fog-cloud devices, leading to maximization of the device life span. However, this paper did not study the challenges of energy-saving for more complicated fog environments, and the suggested model was not proved to be secure.

To improve offloading efficiency and reduce energy consumption, downlink non-orthogonal multiple access, as proposed by Wei and Jiang [125], was applied in fog computing systems. To reach the maximal utility of the system, the transferring power to the fog nodes and the input data's size to the user's task buffer was optimized. The challenge in this technique was transformed into an online optimization challenge. However, the authors did not address the computation capacity of users' devices and cloud servers. Also, the distribution of the user's tasks to their cloud servers, local processers, and the fog nodes was neglected.

Zhang et al. [138] presented the fairness scheduling for every fog node regarding the historical average energy, the offloading energy consumption, and the priority. Also, they suggested lower energy and shorter response time, which could define the transmission power, which fog node to offload, and the offloading subtask size in an energy-reduced method. The analytical results related to the optimal target fog node, the terminal node's optimal transmission power, and the optimal subtask size were gained in an energy-reduced and proper way. When the cluster radius shifted, the system fairness level that adopted the greedy task offloading algorithm remained constant, which seems to be the shortcoming of this study.

Li et al. [63] suggested optimizing the computation offloading and low energy architecture, addressing the problem of resource allocation and computation offloading for cloud collaborated fog-based computing in heterogeneous networks. By computing resource

allocation, transmitting power, and joint optimization of the computation offloading strategy, these networks aim to minimize the system's cost. This paper presented an offloading computation technique, which was energy-efficient, and a resource allocation strategy to reduce the cost of the system. However, one of this paper's deficiencies was that the authors neither noticed the security issues enough, such as privacy, nor addressed stability, scalability, and accuracy.

In a fog computing scenario Chen et al. [27] developed an energy-optimal dynamic computation offloading scheme for industrial IoT in a fog-based computing setup. The authors aimed to minimize energy consumption when the tasks were performed within a desired energy-delay and overhead. However, some techniques can be used to achieve the purpose. For instance, for accomplishing the smart partial offloading, the machine learning methods can be integrated into the solving method.

In fog radio access networks, for upstream IoT offloading Vu et al. [117] proposed a joint energy and latency optimization (JELO) strategy. This strategy developed the optimal task assignment challenge to balance the energy consumption. The experimental results revealed that the JELO strategy overcame the literature's proposed approaches, which could manage load balancing, energy consumption, and response time. However, in this paper, other requirements for the upstream offloading services, i.e., low energy, high capacity, or low response time, were not considered.

4.2.2 Summary of energy-based approaches

A side-by-side comparison of the selected methods in the energy-based category and their most important advantages and disadvantages are shown in Table 6. Some of the advantages are quick response time, QoS, and energy efficiency. On the other hand, some of the disadvantages have been low security and low scalability. The selected papers have been evaluated for qualitative metrics. Eleven metrics have been detected, and the results are accordingly depicted in Table 7.

4.3 Storage-based approaches

Some of the proposed approaches focus on the optimization of data management and data storage in offloading. We classify the research studies, which use this approach under the storage-based category. First, we will describe the storage-based approach for offloading in fog computing. Four selected papers are reviewed in Section 4.3.1. Then in Section 4.3.2, their advantages, disadvantages, main ideas, and evaluation methods will be explored and compared.

4.3.1 Overview of the selected storage-based approaches

Quinton and Aboutorab [93] suggested efficient transmission schemes offload traffic from storage devices by applying IDNC (instantly decodable network coding) and proposed users' cooperation. By applying a maximum weighted vertex search approach, the greedy heuristic approach was used in the IDNC graph with induced subgraphs, and this paper applied the technique of stochastic shortest path over IDNC. As it is conspicuous, network coding could improve throughput, reduce the delay, and provide more robust networks. However, this paper did not address a system model with precise feedback from the users.

A differential synchronization approach was designed by Wang et al. [120]. They suggested architecture for storage offloading and computation based on fog computing and extended this method by introducing the Reed-Solomon code for security measures. They designed an algorithm in which a differential synchronization was utilized to reduce the communication cost and delay. Based on the experiment results and compared with other methods, this architecture performed much better with feasibility. However, this was not the research emphasis in this paper.

Chiti et al. [28] proposed an efficient scheme to offload computationally intensive tasks from end-user storage devices to fog nodes. They evaluated them by resorting to the computer simulations in terms of the worst mean waiting, total completion time, and mean full completion time per task. Furthermore, the authors proposed a distributed deferred acceptance algorithm-based matching method to present an efficient offloading of computation-intensive tasks in fog computing for IoT systems, enabling low response time applications and services. However, the authors did not evaluate this scheme as a security aspect.

Shnaiwer et al. [107] studied the influence of applying clients' heterogeneous enhanced remote radio heads and dual interfaces on the Cloud Base Station (CBS) offloading traffic in the storage of fog radio access networks (F-RAN). Comparing the traditional homogeneous F-RAN strategy and the derived lower bounds, considering throughput and CBS offloading, the simulations could quantify the suggested heterogeneous F-RAN solution's benefits. Based on this paper's simulation results, the suggested heuristic solution performance is comparable to the overloaded networks' optimal performance. However, the authors did not consider security.

4.3.2 Summary of storage-based approaches

A side-by-side comparison of the selected methods in the storage-based category and their most important advantages and disadvantages are demonstrated in Table 8. Some of the advantages are quick response time, low cost, availability, and high flexibility. On the other hand, some of the disadvantages have been low scalability and low availability. The selected papers have been evaluated for qualitative metrics. Six metrics have been detected, and the results are accordingly depicted in Table 9.

4.4 Hybrid approaches

Some of the proposed approaches focus on simultaneous improvements in offloading. We classify the research studies, which use this approach under the hybrid-based category. First, we will describe the hybrid-based approach for offloading in fog computing. Thirteen selected papers are reviewed in Section 4.4.1. Then in Section 4.4.2, their advantages, disadvantages, main ideas, and evaluation methods will be explored and compared.

4.4.1 Overview of the selected hybrid approaches

Considering two kinds of computation and communication capabilities, fog computing and cloud computing, Meng et al. [73] explored a hybrid offloading issue. According to the maximum tolerable response time and the computation energy efficiency of offloading in cloud-fog networks, the authors provided a classification in which the problem was divided into four subproblems. According to this paper, the numerical results showed that the

proposed solution achieved low energy consumption. However, the authors did not evaluate this scheme as an available aspect.

Task offloading decision in a fog computing system was proposed by Zhu et al. [141], who provided offloading protocols and fog computing schemes. This system could effectively locate the fog computing power source nearer to the users' devices, especially mobile devices. The experimental results showed the superiority of the proposed approach in terms of execution time and energy consumption. The presented algorithm was specifically designed to save more energy for mobile devices and minimize the complexity of the offloading formulation. However, some other problems and challenges, such as virtual machine migration and dynamic offloading, were not considered in this paper.

Another study was related to the problem of fog offloading for minimizing IoT service response time. It presented a general framework to understand, formulate, evaluate, and model the service delay in a fog- cloud environment. This project was proposed by Yousefpour et al. [136]. An offloading and delay-minimizing collaboration policy was proposed by the authors for fog-capable devices aiming at reducing the service delay for IoT applications. They also mentioned how offloading policies could help minimize the delay and response time in the IoT environment. Nevertheless, they did not regard other aspects of IoT requests, like the amount of data carried by request.

For fog computing Liu et al. [68] proposed social relationships of the energy harvesting mobile devices (MDs) into the computational offloading model's design. Considering the fact that the MDs were well-equipped with the capabilities of energy harvesting, the energy harvested could be saved in the battery to be applied for data transmission or local execution. In radio and computation aspects, the idea could improve scalability. However, the computational resource in the fog could not be treated as much as the traditional cloud.

Wang et al. [123], concentrating on the edge offloading problem in fog computing, suggested that consumption of energy caused by data transition was lower than that by task execution when the number of instructions was large. The simulation results demonstrated that the scheme performed much better than the other normative schemes in terms of service delay and energy consumption. On the one hand, the service delay and the energy cost increased with the rise in traffic size and computation workload. On the other hand, the offloading ratio fell with an increase in the traffic size and rose with an increase in the computation workload. However, this paper did not consider the advantage of 5G technology.

A high-performance system algorithm was suggested by Wang et al. [123]. They introduced a migration scheme and a mobility-aware task offloading in a generic three-layer computing network architecture. The algorithm could efficiently deal with user equipment mobility scenarios in fog computing networks to maximize the user equipment total revenue. On the other hand, the suggested scheme was capable of reducing the migration time and improving the user equipment income. The authors, however, did not consider how the migration cost decreases for the migrated tasks.

A task offloading scheme was suggested by Misra and Saha [76] for software-defined networks. In these networks, multi-hop IoT access points connect IoT devices to fog computing nodes. A linearization technique was applied by the authors to define the problem integer linear programming formulation. They suggested that the scheme could decrease the average energy consumption and delay. According to this paper, the fog nodes and the access points were considered fixed in a realistic IoT scenario. However, there might be mobile access points and dynamic network conditions which were not taken into account in the existing schemes of an IoT scenario.

Adhikari et al. [4] suggested an optimal strategy of application offloading in a hierarchical fog-cloud environment by applying the technique of APSO (accelerated

particle swarm optimization). An optimal objective function was formulated by the authors with multiple QoS parameters as resource utilization and total cost. A model of energy-efficient fog computing can be regarded to satisfy different QoS parameters of the real-time IoT applications.

Cai et al. [23] studied energy and tasks in fog-enabled IoT networks as the joint offloading. The offloading strategy was applied to minimize the task execution delay, reduce energy consumption for a special task, and analyze a beneficial condition for the joint offloading of tasks and energy. However, they did not evaluate this research strategy's effectiveness and applicability in real-world applications, especially for real industrial applications.

Zaharia, et al. [137] suggested a complete offloading solution and proposed different profiles aiming at diverse goals: to save the battery, to get the maximum rate of data, or to balance these factors. They also offered a simulator to reproduce the manner of the devices in an almost real environment. However, the offloading and the simulator solutions did not assume if any access points or Wi-Fi networks were added.

Mukherjee et al. [79] proposed and implemented a cooperative offloading strategy for mobile cloud networks in indoor environments. The proposal provided an offloading scheme as collaborative code offloading in a femtolet-based fog. The paper also utilized a fog-based strategy in computing and femtolet to minimize energy consumption and response time in offloading. The simulation results illustrated that this method significantly decreased the average energy consumption, response time, and jitter. Yet, it seems that the suggested approach did not consider the optimization of power.

A module placement method classification algorithm (MPCA) and regression tree algorithm were suggested by Rahbari and Nickray [95]. The decision parameters of MPCA to select the most appropriate fog devices included confidentiality, authentication, availability, integrity, speed, capacity, and cost. Based on the results, including response time, power consumption, and performance, the suggested methods were much better than other prior ones. The proposed method could reduce energy consumption against the limitations of completion time. The power consumption of RAM, CPU, and Wi-Fi of MPCA was less than the First-Fit method; however, this method did not regard testing and researching the suggested methods on mobile fog computing with trust and fault-tolerant.

In the fog-cloud environment, Balasubramanian and Meyyappan [12] presented a new technique that used non-cooperative game theory to offload, organize, and reorganize the tasks. This offloading approach enabled the migration of smart gateway for the cloud of things. However, this research scheme did not evaluate the effectiveness and applicability of real-testbed applications, for example, real industrial network applications.

4.4.2 Summary of the hybrid approaches

A side-by-side comparison of the selected methods in the hybrid-based category and their most important advantages and disadvantages are shown in Table 10. Some of the advantages are quick response time, QoS, balanced resource utilization, and energy efficiency. On the other hand, some of the disadvantages are low accuracy and low availability. The selected papers have been evaluated for qualitative metrics. Twelve metrics have been detected, and the results are accordingly presented in Table 11.

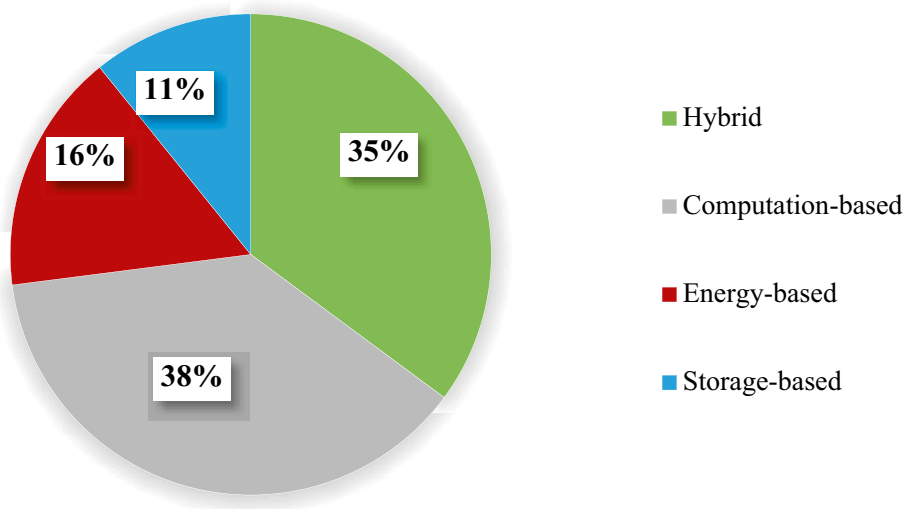


Fig. 6 Percentage of each category of the selected papers

5 Discussion

This section provides a comparison between various offloading mechanism metrics, evaluation and algorithm types, and categories of the selected papers. In the previous section, all of the categories with solutions for improving the offloading mechanisms in fog-based

Table 12 A summarization of the advantages and disadvantages of the discussed categories

Category	Advantages	Disadvantage
Computation-based	• Better energy • Better response time • Better cost	• Low capacity • Low bandwidth • Unacceptable throughput • Low availability • Unacceptable scalability
Energy-based	• Better energy • Better response time • Better cost	• Low capacity • Low bandwidth • Unacceptable security • Low availability • Unacceptable throughput
Storage-based	• Better response time • Better throughput • Better cost	• Low capacity • Low bandwidth • Unacceptable scalability • Low availability • Unacceptable security
Hybrid	• Better energy • Better response time • Better cost	• Low bandwidth • Unacceptable resource utilization • Unacceptable scalability • Low availability • Unacceptable security

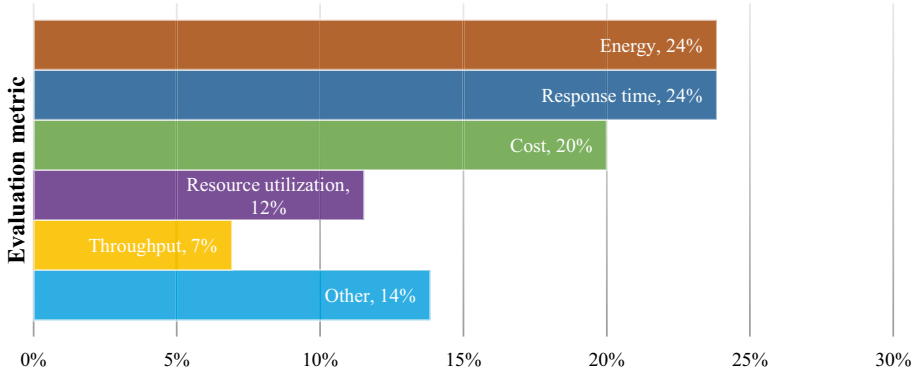


Fig. 7 Percentage of evaluation metrics in all papers

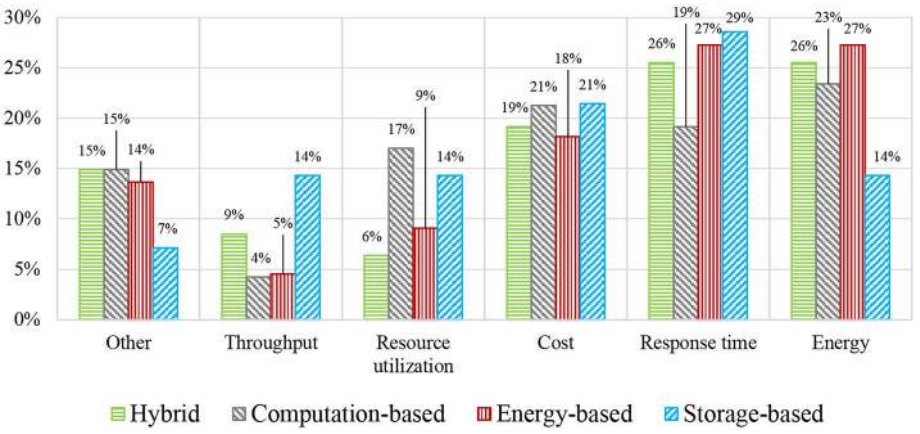


Fig. 8 Percentage of evaluation metrics in categorized papers

networks were reviewed. Now, we provide answers to the research questions proposed in Section 3.1. In response to RQ2, given the presented classification in Table 3 and Section 3, we compared the percentages of various offloading mechanisms in Fig. 6. The computation-based approach was the most interesting mechanism, with 38% in the literature. Hybrid mechanisms with 35% were the next popular mechanisms. Moreover, 16% and 11% of offloading usage belonged to energy-based and storage-based mechanisms, respectively.

According to the collected information from Tables 4, 6, 8, and 10, we briefly present the key strengths and weaknesses of the four approaches in Table 12. According to this table, in the computation-based category, the metrics such as throughput, bandwidth, capacity, availability, and scalability are ignored, while the parameters such as energy, response time, and cost are more respected. In the energy-based category, better energy, response time, and cost are acknowledged, whereas the metrics such as security, bandwidth, capacity, availability, and throughput are overlooked. While better response time, throughput, and cost are remarked in the storage-based category, but focus on bandwidth, capacity, scalability, availability, and security is not acceptable. Finally, in the hybrid category, though the authors adequately scrutinize energy, response time, and cost as

evaluation metrics, they poorly consider resource utilization, bandwidth, scalability, availability, and security.

In response to RQ3, we cannot specifically address which approach is better, but defining which kind of evaluation metric is popular in the category is possible. Figures 7 and 8 show the main evaluation metrics in the categorized papers provided in Tables 5, 7, 9, and 11. In Fig. 7, by formula (1), the number of occurrences of each evaluation metric, as the target parameter, is counted alone and divided by the sum of the number of occurrences of all evaluation parameters [46].

$$\text{Imp\_percentage}(i) = \frac{Occurr\_no(j)}{\sum_{j=1}^{param\_no} Occurr\_no(j)} \tag{1}$$

As we can observe in Fig. 7, 24% of the papers attempted to improve the energy and response time equally, while 20% of the papers tried to reduce the cost. By multiplying the values to 100, the result is the percentage of each evaluation metric's ratio to all of the other evaluation metrics. Figure 8 shows the number of occurrences of all evaluation metrics and their percentages. According to Fig. 8, in the storage-based category, 29% of the papers tried to improve the response time, and 21% of authors reduced the cost. In the energy-based class, in 27% of the papers, the energy is reduced. Moreover, in the same number of papers, response time is considered. In the computation-based category, 23% of the authors attempted to improve the energy, and 21% of them tried to reduce the cost. Finally, in the hybrid category, 26% of the papers made attempts to improve the energy, and the same number of them improved the response time.

In response to RQ4, in Fig. 9, we presented the most critical types in evaluating all papers. According to these tables, the *simulation* type is dominant in all categories except for the computation-based category, in which *design* is the most used type of evaluation.

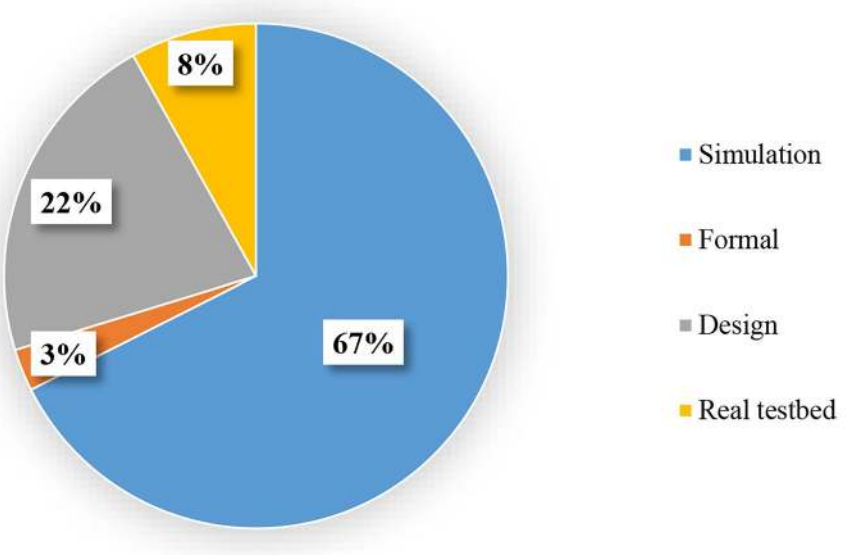


Fig. 9 Percentage of each evaluation method in all works

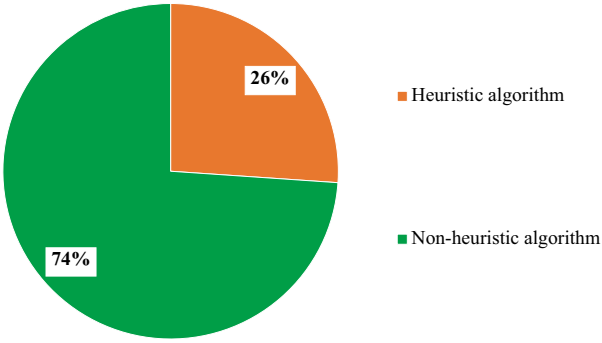


Fig. 10 Percentage of algorithm types in all works

According to Fig. 9, 67% of the research works have used *simulation* type in developing an offloading mechanism. In addition, we detected that 22% of the research papers applied *design* to evaluate the proposed case studies in offloading mechanisms. Moreover, 8% of the research studies had a *real testbed* proposed evaluation method, and also 3% had a *formal* proposed evaluation method.

In response to RQ5, according to Fig. 10, we considered two algorithms that include heuristic algorithms and non-heuristic ones. The non-heuristic algorithms had the highest percentage by 74% in all works, and the heuristic algorithms were applied by 26% in all works.

6 Open issues and future trends

This review presents that some critical issues have not been studied in offloading mechanisms in fog computing. Therefore, in response to RQ6, some open research problems are pointed out in this section.

- *Trustworthiness and security of data and users*
  When it comes to offloading in fog environments, the greater the data we have, the more dangerous it will be in case of data misuse and theft. When systems and multiple nodes have communications with one another, data storage and communication are subject to theft and intrusion. The data might be misused and lead to such big losses as intimidation of lives in a medical environment, malfunctions of manufacturing in an industrial environment, and other hazards. There is no need to say that robust and efficient data security measures are needed. In other words, parameters should be used to have precise decisions in offloading. Therefore, data and users' trustworthiness and security can be an interesting topic for further research [2, 6, 115].
- *Multi-objective mechanisms*
  To address some QoS parameters for offloading in fog computing, we detected that there might not be a single strategy and unique mechanism. In the selected papers, one can discern that some mechanisms (e.g., scheduling and offloading) consider energy consumption and delay. At the same time, other evaluation metrics, such as availability, security, bandwidth, and cost, were overlooked. Thus, to make a trade-off among several different QoS parameters, a mechanism that is optimally multi-objective can be an important and engrossing dimension for further studies.

- *Offloading and big data*
  In the IoT, the generation of big data, as well as processing big real-time data and storage to achieve rich services, is growing. To execute big data-related services in user devices and limit resources in these devices, these services' execution is done through effective offloading mechanisms in fog and cloud data centers [12948, 32]. Designing effective offloading mechanisms for big real-time data transmission can be an attractive research direction.
- *Offloading for the new generation of mobile networks*
  In the future, by permitting a number of mobile users in any place, 5G, the new generation of mobile networks, will enable the networks that have much broader bandwidth and more accessible devices. Even though mobile devices will also be equipped with high speed and availability of network access, the spectrum's resources will be restricted. Moreover, it creates a much heavier burden on the services developing a new generation of mobile networks. It may cause an increment of the cost when fog resources are accessed through the new generation mobile network, such as 5G. Therefore, using 5G networks to offload the data, computation, tasks, and energy in fog computing architecture may remain a challenge and research direction for future works [112, 122].
- *Alternative algorithms*
  In the existing literature, some offloading approaches, which are utilized in fog environments, fall into the NP-complete or NP-hard category in terms of complexity. For solving them, heuristic and meta-heuristic techniques are used. There are some algorithms that have not been addressed in the previous works such as genetic algorithm [84], memetic algorithm [49, 100], simulated annealing [50], ant colony optimization [51, 85], artificial bee colony [47, 52, 53, 101], grey wolf Optimizer [75], firefly algorithm [131], bat algorithm [130], bacterial colony optimization [86], gravitational search algorithm [98], particle swarm optimization [30, 54], glowworm swarm optimization [59], and ant lion optimizer [74]. Therefore, several different algorithms affect fog computing under question and are a research direction for future works.
- *Network management*
  Considering the limitation of resources for offloading in fog networks, software-defined network (SDN) support can make the management of fog networks conformable. SDN, as an emergent networking and computing paradigm, is a method of implementation in network virtualization. The data plane and control plane are separated by this architecture to realize the network traffic flexible control. A centralized server controls and decides the node communication path. SDN, in fog-based networks, can assist in managing the fog networks that are heterogeneous efficient. SDN paradigm, along with fog computing, possibly solve such challenges as high packet loss rate, collisions, and irregular connectivity [56]. The network function virtualization (NFV) and fog computing together create flexibility and speed in constructing, deploying, and managing the novel applicant-based services. Based on NFV, the function of the network is disconnected from the hardware of a dedicated physical network by the technology of device abstraction and leveraging virtualization. In other words, to attain new service deployment and rapid development, the resources can be flexibly and fully shared. The technology of NFV can considerably make the telecommunication service provisioning flexible. NFV beneficial to fog computing in different ways; switches, gateways, and firewalls can be placed and virtualized on fog nodes. To put it more simply, NFV can make the resources capable of seamless augmentation (e.g., storage, computation, and communication) and can orchestrate the functionalities in the widely geo-distributed

and heterogeneous fog networks [39]. It is interesting to research SDN/NFV technologies' application to support QoS needs and demands in fog computing [71, 83, 134].

- *Implementation challenges*

  Those offloading strategies in the fog networks designed to satisfy the QoS parameters need to be executed in the real-testbed and real networks. It obviously detected only 10% of selected papers to validate the proposed mechanisms and used the real-testbed strategy. A task scheduling strategy, for example, must be executed in real patterns, since in practice, many of the proposed architectures, algorithms, and designs may respond contrary to the expected predictions. Hence, one of the robust challenges of the proposed mechanisms is the implementation of those mechanisms in the real world.
- *Scalability of application*

  Some proposed offloading mechanisms for fog networks must be capable of operating on networks that have a huge number of users' devices, and that is on a large scale. These mechanisms' validity cannot be warranted using some nodes or devices in an environment whose scale is small. Although this criterion was essential, the approaches were mostly evaluated in small scale scenarios, which seems to be a new direction for future studies. Scalability, therefore, is an evaluation parameter that can be explored for future studies.
- Carbon-aware offloading for geo-distributed

Data centers that are geographically distributed host such fog applications as social networks, web searches, and video streaming, and in terms of the huge carbon emission and energy demand, they have reached the spotlight. As estimated, by 2020, data centers might consume around 8% of the electricity worldwide and might produce 2.6% carbon emission globally. To decrease the cost of energy and distribute the requests, geographical offloading can use the server heterogeneity of geo-distributed data centers. However, some mechanisms and protocols addressing carbon-aware and green offloading might remain major challenges [82, 140].

7 Conclusion and limitation

In conclusion, in this paper, a systematic study was presented with a focus on the current research studies of offloading mechanisms in fog computing, including its architecture, technologies, and application. In this study, by applying our search query, 131 publications were selected at the initial selection. At the final selection, we selected 37 papers with reference to the research questions and classified them based on their contents. According to RQ2, the applied mechanisms in the offloading of fog computing were classified into four groups, with the highest percentage of studies done in computation-based mechanisms with 38%, hybrid mechanisms with 35%, energy-based mechanisms with 16%, and storage-based mechanisms with 11% of all types of applied mechanisms. They were compared and analyzed according to their significance and crucial evaluation metrics. The key differences, advantages, disadvantages, and important factors of each of the selected works were addressed in the concept of offloading in fog. Based on RQ3, the most important metrics in various proposed approaches were energy and response time by 24% and cost by 20%. According to RQ4, the simulation method (67% of the papers) was dominant in most categories as a method of evaluation, followed by design (22% of the papers) and real-testbed

(8% of the papers). In addition, with respect to the RQ5, the most common algorithms were the non-heuristic ones by 74% and the heuristic ones by 26%. Furthermore, based on RQ6, the existing fog offloading mechanisms have faced several open issues and future trends such as trustworthiness and security, multi-objective mechanisms, big data analytics, new generation mobile networks, network management, and carbon-aware offloading for geo-distributed. Lucidly, the most important challenges are scalability and real-testbed implementation.

The systematic, exhaustive, and thorough investigation of offloading mechanisms in fog computing, offered in this paper, have some drawbacks and limitations as denoted below:

- *Study domain*: Several sources, in print or online, contained and covered the studies on offloading mechanisms in fog computing. The key academic international journals, specifically JCR journals, were included to reach the best reliability and competency. In this paper, we ignored and eliminated the papers that were published in national journals, conference papers, book chapters, and review papers.
- *Study and publication bias*: Google Scholar, Springer, IEEE, ScienceDirect, and Wiley were selected as reliable databases. The surveys demonstrate that these electronic sources often suggest valid papers and most related articles. However, academics cannot select all of the relevant papers in the area of offloading mechanisms in fog computing.
- *Reliability of result*: Another possible challenge in paper reviewing is the reliability of the data results. By involving multiple researchers, according to guidelines provided in [57], we tried to give a clarified, replicable, and assessable process with clear steps that can be externally controlled, checked and evaluated. According to the guidelines, the metrics' statistics provided in tables and figures were found by full-text reviews of multiple researchers' selected papers. However, though we were committed to following the procedures provided in references, the study may occasionally have diverged from them.
- *Study queries*: We selected five questions for our study as research questions. Nevertheless, some other questions can be addressed, as well.
- *Classification*: The papers were grouped into four categories, including computation-based, energy-based, storage-based, and hybrid. However, they can be classified otherwise.

References

1. Aazam M, Huh E-N (2016) Fog computing: the cloud-IoT/IoE middleware paradigm. IEEE Potentials 35(3):40–44
2. 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
3. Ahmadi Z, Haghi Kashani M, Nikravan M, Mahdipour E (2021) Fog-based healthcare systems: A systematic review. Multimedia Tools and Applications
4. Adhikari M, Srirama SN, Amgoth T (2019) Application offloading strategy for hierarchical fog environment through swarm optimization. IEEE Internet Things J 7:4317
5. Akherfi K, Gerndt M, Harroud H (2018) Mobile cloud computing for computation offloading: issues and challenges. Appl Comput Inf 14(1):1–16
6. Ali M et al (2017) IoTFLiP: IoT-based flipped learning platform for medical education. Digit Commun Netw 3(3):188–194

7. Al-Zinati M, Alrashdan R, Al-Duwairi B, Aloqaily M (2021) A re-organizing biosurveillance framework based on fog and mobile edge computing. Multimed Tools Appl 80:16805–16825
8. Asghari P, Rahmani AM, Javadi HHS (2018) Service composition approaches in IoT: a systematic review. J Netw Comput Appl 120:61–77
9. Asghari P, Rahmani AM, Javadi HHS (2019) Internet of things applications: a systematic review. Comput Netw 148:241–261
10. Ashton K (2009) That 'internet of things' thing. RFID J 22(7):97–114
11. Balan RK (2006) Simplifying cyber foraging. School of Computer Science, Carnegie Mellon University
12. Balasubramanian S, Meyyappan T (2020) Game theory based offload and migration-enabled smart gateway for cloud of things in fog computing. Computing in engineering and technology. Springer, Heidelberg, pp 253–266
13. Bazzaz Abkenar S, Haghi Kashani M, Mahdipour E, Jameii SM (2020) Big data analytics meets social media: A systematic review of techniques, open issues, and future directions. Telemat Inform 2020:101517
14. Bazzaz Abkenar S, Haghi Kashani M, Akbari M, Mahdipour E (2021) Twitter spam detection using machine learning approaches: A systematic review. IEEE Transactions on Knowledge and Data Engineering, submitted for publication
15. Ben Hamida E, Chelius G, Gorce JM (2009) Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. SIMULATION 85(9):574–588
16. Bilal K, Khalid O, Erbad A, Khan SU (2018) Potentials, trends, and prospects in edge technologies: fog, cloudlet, mobile edge, and micro data centers. Comput Netw 130:94–120
17. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech 2008(10):P10008
18. 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, 2012: ACM, pp. 13–16
19. Bonomi F, Milito R, Natarajan P, Zhu J (2014) Fog computing: a platform for internet of things and analytics. Big data and internet of things: a roadmap for smart environments. Springer, Heidelberg, pp 169–186
20. Bouachir O, Aloqaily M, Tseng L, Boukerche A (2020) Blockchain and fog computing for cyber-physical systems: the case of smart industry. Computer 53(9):36–45
21. Boukerche A, Guan S, Grande RED (2019) Sustainable offloading in mobile cloud computing: algorithmic design and implementation. ACM Comput Surv 52(1):11
22. Brereton P, Kitchenham BA, Budgen D, Turner M, Khalil M (2007) Lessons from applying the systematic literature review process within the software engineering domain. J Syst Softw 80(4):571–583
23. Cai P, Yang F, Wang J, Wu X, Yang Y, Luo X (2020) JOTE: joint offloading of tasks and energy in fog-enabled IoT networks. IEEE Internet Things J 7:3067
24. Cao B, Zhang L, Li Y, Feng D, Cao W (2019) Intelligent offloading in multi-access edge computing: a state-of-the-art review and framework. IEEE Commun Mag 57(3):56–62
25. Chen T-Y (1997) A measurement of the resource utilization efficiency of university libraries. Int J Prod Econ 53(1):71–80
26. Chen Z, Cheng S (2019) Computation offloading algorithms in mobile edge computing system: a survey. International conference of pioneering computer scientists, engineers and educators. Springer, Heidelberg, pp 217–225
27. Chen S, Zheng Y, Lu W, Varadarajan V, Wang K (2019) Energy-optimal dynamic computation offloading for industrial IoT in fog computing. IEEE Trans Green Commun Netw 4(2):566
28. Chiti F, Fantacci R, Picano B (2018) A matching theory framework for tasks offloading in fog computing for IoT systems. IEEE Internet Things J 5(6):5089–5096
29. Dolui K, Datta SK (2017) Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. In: 2017 Global Internet of Things Summit (GIoTS), 2017: IEEE. pp. 1–6
30. Dordaie N, Navimipour NJ (2017) A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. ICT Express 4:199
31. Du J, Zhao L, Feng J, Chu X (2018) Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. IEEE Trans Commun 66(4):1594–1608
32. Fathi M, Haghi Kashani M, Jameii S M, Mahdipour E (2021) Big Data Analytics in Weather Forecasting: A Systematic Review. Archives of Computational Methods in Engineering

33. Fricker C, Guillemin F, Robert P, Thompson G (2016) Analysis of an offloading scheme for data centers in the framework of fog computing. ACM Trans Model Perform Eval Comput Syst 1(4):16
34. Ghobaei-Arani M, Souri A, Rahmanian AA (2019) Resource management approaches in fog computing: a comprehensive review. J Grid Comput 18:1–42
35. Guo C et al (2009) BCube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM Comput Commun Rev 39(4):63–74
36. Haghi Kashani M, Ahmadzadeh A, Mahdipour E (2021) Load balancing algorithms in fog computing: A systematic review. IEEE Transactions on Services Computing, submitted for publication
37. Haghi Kashani M, Rahmani AM, Jafari Navimipour N (2020) Quality of service-aware approaches in fog computing. Int J Commun Syst 33:e4340
38. Haghi Kashani M, Madanipour M, Nikravan M, Asghari P, Mahdipour E (2021) A systematic review of IoT in healthcare: Applications, techniques, and trends. Journal of Network and Computer Applications 192:103164
39. Han B, Gopalakrishnan V, Ji L, Lee S (2015) Network function virtualization: challenges and opportunities for innovations. IEEE Commun Mag 53(2):90–97
40. Hu P, Ning H, Qiu T, Zhang Y, Luo X (2016) Fog computing based face identification and resolution scheme in internet of things. IEEE Trans Ind Inf 13(4):1910–1920
41. Hu P, 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
42. Huang D, Wang P, Niyato D (2012) A dynamic offloading algorithm for mobile computing. IEEE Trans Wirel Commun 11(6):1991–1995
43. Iorga M, Feldman L, Barton R, Martin MJ, Goren NS, Mahmoudi C (2018) Fog computing conceptual model. NIST, Gaithersburg
44. Jiang Y, Tsang DH (2018) Delay-aware task offloading in shared fog networks. IEEE Internet Things J 5(6):4945–4956
45. Jiang Y-L, Chen Y-S, Yang S-W, Wu C-H (2018) Energy-efficient task offloading for time-sensitive applications in fog computing. IEEE Syst J 13:2930
46. Jula A, Sundararajan E, Othman Z (2014) Cloud computing service composition: a systematic literature review. Expert Syst Appl 41(8):3809–3824
47. Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697
48. Karimi Y, Haghi Kashani M, Akbari M, Mahdipour E (2021) Leveraging big data in smart cities: A systematic review. Concurrency and Computation: Practice and Experience
49. Kashani MH, Jahanshahi M (2009) A new method based on memetic algorithm for task scheduling in distributed systems. Int J Simul Syst Sci Technol 10:25
50. Kashani MH, Jahanshahi M (2009) Using simulated annealing for task scheduling in distributed systems. In: 2009 International Conference on Computational Intelligence, Modelling and Simulation: IEEE. pp. 265–269
51. Kashani M, Sarvizadeh R (2011) A novel method for task scheduling in distributed systems using Max-Min Ant Colony Optimization. In: 2011 3rd International Conference on Advanced Computer Control: IEEE. pp. 422–426
52. Kashani MH , Jameii SM, Akbari M, Tayebi RM (2011) Utilizing bee colony to solve task scheduling problem in distributed systems. In: 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN): IEEE. pp. 298–303
53. Kashani MH, Sarvizadeh R, Jameii SM (2012) A new distributed systems scheduling algorithm: a swarm intelligence approach. In: Fourth International Conference on Machine Vision (ICMV 2011): Computer Vision and Image Analysis; Pattern Recognition and Basic Technologies. vol. 8350: International Society for Optics and Photonics, p. 835005
54. Kashani MH, Zarrabi H, Javadzadeh G (2017) A new metaheuristic approach to task assignment problem in distributed systems. In: 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI): IEEE, pp. 0673–0677
55. Khan AUR, Othman M, Khan AN, Shuja J, Mustafa S (2017) Computation offloading cost estimation in mobile cloud application models. Wirel Pers Commun 97(3):4897–4920
56. Kim H, Feamster N (2013) Improving network management with software defined networking. IEEE Commun Mag 51(2):114–119
57. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic literature reviews in software engineering–a systematic literature review. Inf Softw Technol 51(1):7–15
58. Kitchenham B, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE-200701, Keele Univ. and Univ. of Durham

59. Krishnanand K, Ghose D (2009) Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell 3(2):87–124
60. Kumar K, Lu Y-H (2010) Cloud computing for mobile users: can offloading computation save energy? Computer 4:51–56
61. Kumari A, Tanwar S, Tyagi S, Kumar N (2018) Fog computing for Healthcare 4.0 environment: opportunities and challenges. Comput Electr Eng 72:1–13
62. Li L, Guan Q, Jin L, Guo M (2019) Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system. IEEE Access 7:9912–9925
63. Li Q, Zhao J, Gong Y, Zhang Q (2019) Energy-efficient computation offloading and resource allocation in fog computing for internet of everything. China Commun 16(3):32–41
64. Liang K, Zhao L, Zhao X, Wang Y, Ou S (2016) Joint resource allocation and coordinated computation offloading for fog radio access networks. China Commun 13(Supplement2):131–139
65. 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
66. Liu L, Chang Z, Guo X, Mao S, Ristaniemi T (2017) Multiobjective optimization for computation offloading in fog computing. IEEE Internet Things J 5(1):283–294
67. Liu Y, Yu FR, Li X, Ji H, Leung VC (2018) Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access. IEEE Trans Veh Technol 67(12):12137–12151
68. 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
69. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor 19(3):1628–1656
70. Marín-Tordera E, Masip-Bruin X, García-Almiñana J, Jukan A, Ren G-J, Zhu J (2017) Do we all really know what a fog node is? Current trends towards an open definition. Comput Commun 109:117–130
71. Mayes JC, Coile BW (1998) Security system for network address translation systems. US Patent 5,793,763
72. Mell P, Grance T (2011) The NIST definition of cloud computing. NIST, Gaithersburg
73. Meng X, Wang W, Zhang Z (2017) Delay-constrained hybrid computation offloading with cloud and fog computing. IEEE Access 5:21355–21367
74. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
75. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
76. Misra S, Saha N (2019) Detour: dynamic task offloading in software-defined fog for IoT applications. IEEE J Sel Areas Commun 37(5):1159–1166
77. Mouradian C, Naboulsi D, Yangui S, Glitho RH, Morrow MJ, Polakos PA (2017) A comprehensive survey on fog computing: State-of-the-art and research challenges. IEEE Commun Surv Tutor 20(1):416–464
78. 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
79. 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(6):2412–2448
80. Munir A, Kansakar P, Khan SU (2017) IFCIoT: integrated fog cloud IoT: a novel architectural paradigm for the future internet of things. IEEE Consumer Electron Mag 6(3):74–82
81. Naas MI, Parvedy PR, Boukhobza J, Lemarchand L (2017) iFogStor: an IoT data placement strategy for fog infrastructure. In: 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC): IEEE. pp. 97–104
82. Nan Y et al (2017) Adaptive energy-aware computation offloading for cloud of things systems. IEEE Access 5:23947–23957
83. Neghabi AA, Navimipour NJ, Hosseinzadeh M, Rezaee A (2018) Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. IEEE Access 6:14159–14178
84. Nikravan M, Kashani M (2007) A genetic algorithm for process scheduling in distributed operating systems considering load balancing. In: Proceedings 21st European Conference on Modelling and Simulation Ivan Zelinka, Zuzana Oplatkova, Alessandra Orsoni, ECMS: Citeseer
85. Nikravan M, Kashani M (2007) Parallel Min-Max Ant colony System (MMAS) for dynamic process scheduling in distributed operating systems considering load balancing. In: Proceedings of the 21st ECMS International Conference on High Performance Computing & Simulation (HPCS), Prague, Czech Republic

86. Niu B, Wang H (2012) Bacterial colony optimization. Discret Dyn Nat Soc. https://doi.org/10.1155/2012/698057
87. OpenFog Consortium Architecture Working Group (2016) Openfog architecture overview. White Paper OPFWP001, 216, 35
88. OpenFog Consortium Architecture Working Group (2017) OpenFog reference architecture for fog computing. OpenFog Consortium, Fremont
89. Orsini G, Bade D, Lamersdorf W (2015) Context-aware computation offloading for mobile cloud computing: requirements analysis, survey and design guideline. Procedia Comput Sci 56:10–17
90. Pai S et al (2008) Transactional confidentiality in sensor networks. IEEE Secur Priv 6(4):28–35
91. Pandya S, Hadfield A (2004) Response time and resource consumption management in a distributed network environment. US Patent 10/704,494, 27 May 2004
92. Plumb JN, Stutsman R (2018) Exploiting Google's Edge Network for Massively Multiplayer Online Games. In: 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC). pp. 1–8
93. Quinton B, Aboutorab N (2018) Network coding for backhaul offloading in D2D cooperative fog data networks. Wirel Commun Mob Comput. https://doi.org/10.1155/2018/1245720
94. Rabie AH, Ali SH, Ali HA, Saleh AI (2019) A fog based load forecasting strategy for smart grids using big electrical data. Clust Comput 22(1):241–270
95. Rahbari D, Nickray M (2019) Task offloading in mobile fog computing by classification and regression tree. Peer-to-Peer Netw Appl 13:1–19
96. Rahimi M, Songhorabadi M, Kashani MH (2020) Fog-based smart homes: a systematic review. J Netw Comput Appl 153:102531
97. Rahman G, Chuah CW (2018) Fog computing, applications, security and challenges, review. Int J Eng Technol 7(3):1615–1621
98. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
99. Ruan L, Liu Z, Qiu X, Wang Z, Guo S, Qi F (2018) Resource allocation and distributed uplink offloading mechanism in fog environment. J Commun Netw 20(3):247–256
100. Sarvizadeh R, Kashani MH (2012) A swarm intelligence based memetic algorithm for task allocation in distributed systems. In: Fourth International Conference on Machine Vision (ICMV 2011): Machine Vision, Image Processing, and Pattern Analysis, 2012, vol. 8349: International Society for Optics and Photonics, p. 83490X
101. Sarvizadeh R, Kashani MH, Zakeri FS, Jameii SM (2012) A novel bee colony approach to distributed systems scheduling. Int J Comput Appl 42(10):1–6
102. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4):14–23
103. Shah-Mansouri H, Wong VW (2018) Hierarchical fog-cloud computing for IoT systems: a computation offloading game. IEEE Internet Things J 5(4):3246–3257
104. Shakarami A, Ghobaei-Arani M, Masdari M, Hosseinzadeh M (2020) A Survey on the computation offloading approaches in mobile edge/cloud computing environment: a stochastic-based perspective. J Grid Comput 18:639
105. Shakarami A, Ghobaei-Arani M, Shahidinejad A (2020) A survey on the computation offloading approaches in mobile edge computing: a machine learning-based perspective. Comput Netw 182:107496
106. Shakarami A, Shahidinejad A, Ghobaei-Arani M (2020) A review on the computation offloading approaches in mobile edge computing: a game-theoretic perspective. Software 50(9):1719–1759. https://doi.org/10.1002/spe.2839
107. Shnaiwer YN, Sorour S, Al-Naffouri TY, Al-Ghadhban SN (2019) Opportunistic network coding-assisted cloud offloading in heterogeneous fog radio access networks. IEEE Access 7:56147
108. Shuja J et al (2016) Towards native code offloading based MCC frameworks for multimedia applications: a survey. J Netw Comput Appl 75:335–354
109. Shuja J et al (2018) SIMDOM: a framework for SIMD instruction translation and offloading in heterogeneous mobile architectures. Trans Emerg Telecommun Technol 29(4):e3174. https://doi.org/10.1002/ett.3174
110. Shuja J, Mustafa S, Ahmad RW, Madani SA, Gani A, Khan MK (2017) Analysis of vector code offloading framework in heterogeneous cloud and edge architectures. IEEE Access 5:24542–24554
111. Singh V K, Kolekar M H (2021) Deep learning empowered COVID-19 diagnosis usingchest CT scan images for collaborative edge-cloud computing platform. Multimedia Tools Appl
112. Singh S, Chiu Y-C, Tsai Y-H, Yang J-S (2016) Mobile edge fog computing in 5G era: Architecture and implementation. In: 2016 International Computer Symposium (ICS): IEEE. pp. 731–735

113. Singh SP, Nayyar A, Kumar R, Sharma A (2018) Fog computing: from architecture to edge computing and big data processing. J Supercomput 75(4):2070–2105
114. Songhorabadi M, Rahimi M, Farid AMM, Kashani MH (2020) Fog computing approaches in smart cities: a state-of-the-art review. Accessed https://arxiv.org/abs/2011.14732
115. Su Z, Biennier F, Lv Z, Peng Y, Song H, Miao J (2017) Toward architectural and protocol-level foundation for end-to-end trustworthiness in Cloud/Fog computing. IEEE Trans Big Data. https://doi.org/10.1109/TBDATA.2017.2705418
116. Tran M-Q, Nguyen DT, Le VA, Nguyen DH, Pham TV (2019) Task placement on fog computing made efficient for IoT application provision. Wirel Commun Mob Comput. https://doi.org/10.1155/2019/6215454
117. Vu DN et al (2018) Joint energy and latency optimization for upstream IoT offloading services in fog radio access networks. Trans Emerg Telecommun Technol 30:e3497
118. Wang Q, Chen S (2020) Latency-minimum offloading decision and resource allocation for fog-enabled internet of things networks. Trans Emerg Telecommun Technol 31(12):e3880
119. Wang X, Ning Z, Wang L (2018) Offloading in Internet of vehicles: a fog-enabled real-time traffic management system. IEEE Trans Ind Inf 14(10):4568–4578
120. Wang T, Zhou J, Liu A, Bhuiyan MZA, Wang G, Jia W (2018) Fog-based computing and storage offloading for data synchronization in IoT. IEEE Internet Things J 6:4272
121. Wang T, Li P, Wang X, Wang Y, Guo T, Cao Y (2019) A comprehensive survey on mobile data offloading in heterogeneous network. Wirel Netw 25(2):573–584
122. Wang J, Pan J, Esposito F, Calyam P, Yang Z, Mohapatra P (2019) Edge cloud offloading algorithms: Issues, methods, and perspectives. ACM Comput Surv 52(1):2
123. Wang D, Liu Z, Wang X, Lan Y (2019) Mobility-aware task offloading and migration schemes in fog computing networks. IEEE Access 7:43356–43368
124. Wang B, Wang C, Huang W, Song Y, Qin X (2020) A survey and taxonomy on task offloading for edge-cloud computing. IEEE Access 8:186080–186101
125. Wei Z, Jiang H (2018) Optimal offloading in fog computing systems with non-orthogonal multiple access. IEEE Access 6:49767–49778
126. Wu H (2018) Multi-objective decision-making for mobile cloud offloading: a survey. IEEE Access 6:3962–3976
127. Wu E-K, Chen M-Z (2004) JTCP: Jitter-based TCP for heterogeneous wireless networks. IEEE J Sel Areas Commun 22(4):757–766
128. Wu Q, Ge H, Liu H, Fan Q, Li Z, Wang Z (2019) A task offloading scheme in vehicular fog and cloud computing system. IEEE Access 8:1173
129. Xu X et al (2019) A computation offloading method over big data for IoT-enabled cloud-edge computing. Future Gener Comput Syst 95:522–533
130. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, Heidelberg, pp 65–74
131. Yang X-S (2013) Multiobjective firefly algorithm for continuous optimization. Engineering with Computers 29:175-184
132. Yang Y, Liu X, Guo W, Zheng X, Dong C, Liu Z (2020) Multimedia access control with secure provenance in fog-cloud computing networks. Multimed Tools Appl 79(15):10701–10716
133. Yi S, Hao Z, Qin Z, Li Q (2015) Fog computing: platform and applications. In: 2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb): IEEE. pp. 73–78
134. Yi S, Li C, Li Q (2015) A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 workshop on mobile big data: ACM. pp. 37–42
135. Yousefpour A et al (2019) All one needs to know about fog computing and related edge computing paradigms: a complete survey. J Syst Architect 98:289
136. Yousefpour A, Ishigaki G, Gour R, Jue JP (2018) On reducing IoT service delay via fog offloading. IEEE Internet Things J 5(2):998–1010
137. Zaharia G-E, Ciobanu R-I, Dobre C (2019) Machine learning-based traffic offloading in fog networks. Simul Model Pract Theory 101:102045
138. Zhang G, Shen F, Liu Z, Yang Y, Wang K, Zhou M-T (2018) FEMTO: fair and energy-minimized task offloading for fog-enabled IoT networks. IEEE Internet Things J 6:4388
139. Zhang C, Cho H-H, Chen C-Y (2019) Emergency-level-based healthcare information offloading over fog network. Peer-to-Peer Netw Appl 13:1–11
140. Zhou Z, et al (2013) Carbon-aware load balancing for geo-distributed cloud services. In: 2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems: IEEE. pp. 232–241

141. Zhu Q, Si B, Yang F, Ma Y (2017) Task offloading decision in fog computing system. China Commun 14(11):59–68