# All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey



Ashkan Yousefpour
University of Texas at Dallas


Krishna Priyanka Kadiyala
University of Texas at Dallas


Fatemeh Jalali
IBM Research


Amirreza Niakanlahiji
University of North Carolina at Charlotte

# All One Needs to Know about Fog Computing and Related Edge Computing Paradigms: A Complete Survey

Open/Public Access*†

Ashkan Yousefpour‡
UT Dallas

Caleb Fung
UT Dallas

Tam Nguyen
UT Dallas

Krishna Kadiyala
UT Dallas

Fatemeh Jalali
IBM Research

Amirreza Niakanlahiji
UNC Charlotte

Jian Kong
UT Dallas

Jason P. Jue
UT Dallas

## ABSTRACT

With the Internet of Things (IoT) becoming part of our daily life and our environment, we expect rapid growth in the number of connected devices. IoT is expected to connect billions of devices and humans to bring promising advantages for us. With this growth, fog computing, along with its related edge computing paradigms, such as multi-access edge computing (MEC) and cloudlet, are seen as promising solutions for handling the large volume of security-critical and time-sensitive data that is being produced by the IoT. In this paper, we first provide a tutorial on fog computing and its related computing paradigms, including their similarities and differences. Next, we provide a taxonomy of research topics in fog computing, and through a comprehensive survey, we summarize and categorize the efforts on fog computing and its related computing paradigms. Finally, we provide challenges and future directions for research in fog computing.

## KEYWORDS

Fog Computing, Edge Computing, Cloud Computing, Internet of Things (IoT), Cloudlet, Mobile Edge Computing, Multi-access Edge Computing, Mist Computing

## 1 INTRODUCTION

In today's information technology age, data is the main commodity, and possessing more data typically generates more value in data-driven businesses. According to the International Data Corporation (IDC), the amount of digital data generated surpassed 1 zettabyte in 2010 [1]. Furthermore, 2.5 exabytes of new data is generated each day since 2012 [2]. Cisco estimates that there will be around 50 billion connected devices by 2020 [3]. These connected devices constitute the Internet of Things (IoT) and possibly generate a massive amount of data. With this astronomical amount of data, the current mobile network architectures will have trouble managing the momentum and magnitude of data. In current implementations of cloud-based applications, most data that needs storage, analysis, and decision making is sent to the data centers in the cloud [4].

As the data velocity and volume increases, moving the big data from IoT devices to the cloud might not be efficient, or might be even infeasible in some cases due to bandwidth constraints. On the other hand, as time-sensitive and location-aware applications emerge (such as patient monitoring, real-time manufacturing, self-driving cars, flocks of drones, or cognitive assistance), the distant cloud will not be able to satisfy the ultra-low latency requirements of these applications, provide location-aware services, or scale to the magnitude of the data that these applications produce [5]. Moreover, in some applications, sending the data to the cloud may not be a feasible solution due to privacy concerns.

In order to address the issues of high-bandwidth, geographically-dispersed, ultra-low latency, and privacy-sensitive applications, there is a quintessential need for a computing paradigm that takes place closer to connected devices. Fog computing has been proposed by both industry and academia [6, 7] to address the above issues and to quench the need for computing paradigm closer to connected devices. Fog computing bridges the gap between the cloud and IoT devices by enabling computing, storage, networking, and data management on the network nodes within the close vicinity of IoT devices. Therefore, computation, storage, networking, decision making, and data management occur along the path between IoT devices and the cloud, as data moves to the cloud from the IoT devices. Other similar computing paradigms to fog computing such as edge computing, mist computing, cloud of things, and cloudlets, have been proposed by the research community to address the mentioned issues. In this survey, we compare fog computing with other related computing paradigms, and argue that fog computing is a more general form of computing, mainly due to its comprehensive definition scope and flexibility.

In this article, we present a comprehensive survey on fog computing, and discuss how fog computing can meet the growing demand of applications with strict latency, privacy, and bandwidth requirements. A comparison of the related survey papers in the area of fog computing is included in Section 2. To gain a thorough understanding of fog computing, in Section 3, we will first look at cloud computing, then discuss how fog computing extends cloud computing to address the above issues of cloud, and finally, compare fog computing to other similar computing paradigms. Next, in Section 4, we describe our taxonomy of research topics in fog computing. Later, in a comprehensive survey, we summarize and categorize the efforts on fog computing and its related computing paradigms. In Section 5, we present the challenges and limitations in the fog computing area and provide future directions and potential starting

* A complete list of conferences, journals, and magazines that publish state-of-the-art research papers on fog computing and its related edge computing paradigms is available at https://anrlutdallas.github.io/resource/projects/fog-computing-conferences.html. We have included papers from the above list in this survey.

† The data (categories and features/objectives of the papers) of this survey are available at https://github.com/ashkan-software/fog-survey-data

‡ ashkan@utdallas.edu

points for those challenges. Finally, Section 6 concludes the paper. Fig. 1 shows the structure of the survey and a reading map for the reader.

## 2 RELATED SURVEYS

There are are some existing related studies in the area of fog computing that have attempted to provide a survey of the papers in the field of fog computing, edge computing, or MEC. The authors in [8] present a comprehensive review of current literature in fog computing with a focus on architectures and algorithms in fog systems. They further sketch the prospects of fog computing in terms of emerging technologies with a focus on Tactile Internet. Cihat et al. [9] state the importance of cooperation between edge and cloud computing, and motivate how edge computing can benefit from Software Defined Networking (SDN). They note the technical challenges in edge computing and propose using SDN as a solution for implementing edge computing infrastructure. The authors survey publications primarily on edge computing and SDN to support their argument and give future directions for SDN developments. The authors in [10] compile a comprehensive survey of recent efforts in fog-enabled network architectures, and provide various network applications of fog computing.

The recent survey in [11] focuses on connectivity and device configuration aspects of the fog computing and identifies major features that fog computing platforms need to build infrastructure for smart city applications. They further review existing approaches that have been proposed to tackle the challenges in fog computing for building such smart city infrastructure. Comparably, the authors of [12] focus on architecture design and system management of edge computing to provide a detailed and focused survey in the edge computing field. They also characterize fog and edge computing by comparing a list of related computing concepts, including peer-to-peer computing, mobile grid computing, and mobile crowd computing. The authors in [13] take a closer look at fog-assisted IoT applications, discuss security and privacy challenges in fog computing, and review and analyze promising techniques to resolve security and privacy issues in fog-assisted IoT applications.

There are a number of surveys in the area of MEC that also discuss similar concepts to fog computing and summarize papers applicable to fog computing research. The survey in [14] introduces a survey on MEC and focuses on the fundamental key enabling technologies in MEC. The paper also analyzes the MEC reference architecture, overviews the current standardization activities, and introduces main deployment scenarios. Similarly, the survey in [15] provides a survey of the recent state of MEC research with a focus on joint radio and computation resource management.

### 2.1 Contribution

Different from the mentioned surveys, the contribution of this paper is three-fold: (1) We provide a detailed tutorial on fog computing, how it is defined, and how it is related to or different from other similar computing paradigms, such as cloud computing, cloudlets, edge computing, and MEC (2) We propose an exhaustive taxonomy

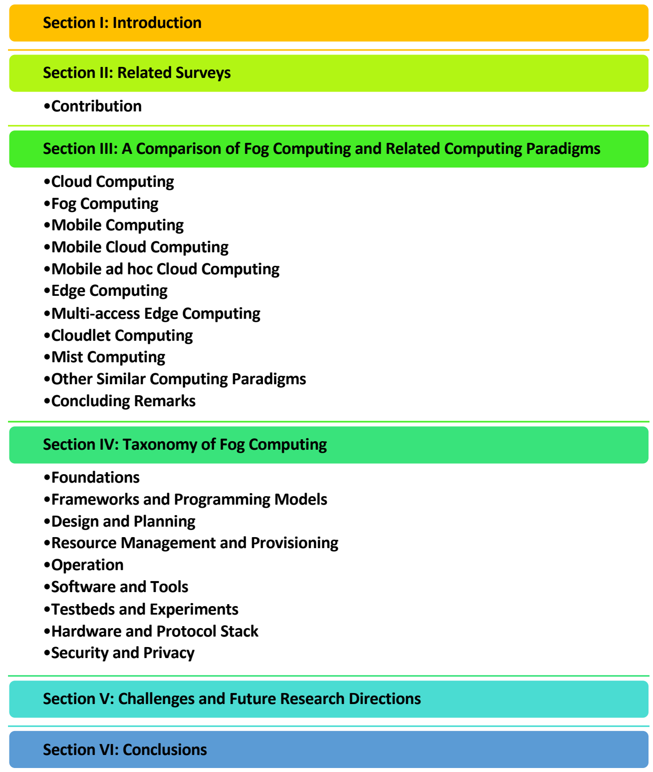


**Figure 1: The structure of the survey.**

of research topics in the area of fog computing, and present a comprehensive survey on fog computing.[1] (3) We have compiled a list of challenges and future directions for research in fog computing.

## 3 A COMPARISON OF FOG COMPUTING AND RELATED COMPUTING PARADIGMS

This section focuses on the comparison of fog computing and related computing paradigms to demonstrate the value of fog computing in a variety of use cases. Moreover, this section provides a better understanding of how these computing paradigms can benefit the current and future landscape of connected devices. We compare fog computing with cloud computing as well as other related computing paradigms and summarize this information in Tables 2 and 3.

### 3.1 Cloud Computing

Cloud computing has been instrumental in expanding the reach and capabilities of computing, storage, and networking infrastructure to the applications. The National Institute of Standards and Technology (NIST) defines cloud computing as a model that promotes ubiquitous, on-demand network access to shared computing

[1]We compiled a comprehensive list of conferences, journals, and magazines that publish state-of-the-art research papers on fog computing and its related edge computing paradigms The list is available at
https://anrlutdallas.github.io/resource/projects/fog-computing-conferences.html

resources [16]. Cloud data centers are large pools of highly accessible virtualized resources that can be dynamically reconfigured for a scalable workload; this reconfigurability is beneficial for clouds services that are offered with a pay-as-you-go cost model [17]. The pay-as-you-go cost model allows users to conveniently access remote computing resources and data management services, while only being charged for the amount of resources they use. Cloud providers, such as Google, IBM, Microsoft, and Amazon provide and provision large data centers to host these cloud-based resources.

*3.1.1 Cloud Services.* Cloud offers infrastructure, platform, and software as services (IaaS, PaaS, SaaS). Application developers can use a variety of these services depending on the needs of the applications they develop. Infrastructure as a service (IaaS) allows cloud consumers to directly access IT infrastructures for processing, storage, and networking resources [18]. Suppose Sam wants to set up a high-tech agricultural system that utilizes IoT devices to monitor the condition of crops. Sam contacts a cloud provider and acquires an IaaS for development of his system. Sam now can configure the IaaS (often offered as a standalone VM) in terms of hardware and corresponding software for his need. Control over infrastructure (IaaS) allows Sam to customize hardware configuration, such as the number of CPU cores and RAM capacity, in addition to systems-level software. Sam can obtain an IaaS from Amazon Web Services (AWS), Microsoft Azure, or Google Compute Engine (GCE).

On the other hand, platform as a service (PaaS) allows cloud consumers to develop software and fully supports software lifecycle – often with the help of a middleware – for software management and configuration. If Sam does not need to configure the infrastructure of the cloud, managing and configuration of hardware and software may detract from the productivity of Sam's business. Now, Sam could consider using PaaS offered by Apache Stratos, Azure App Services, or Google App Engine for his business. PaaS manages the underlying low-level processes and allows Sam to focus on managing software for his IoT-specific interactions. Moreover, PaaS providers often include tools for convenient management of databases and scaling applications.

Now suppose Sam is willing to spend more money and likes to get full software packages, and he does not want to take care of software issues, such as database scalability, socket management, etc. Software as a service (SaaS) provides Sam an environment to centrally host his applications and removes the need for him to install software manually. Sam's client software now can be hosted on Google Apps or as a Web application.

As demonstrated by these examples, cloud services can be utilized for distinct use cases for a variety of end users. Figure 2 illustrates the relationship among IaaS, PaaS, and SaaS with the underlying cloud infrastructure, and illustrates what portion of the application stack is managed by cloud providers.

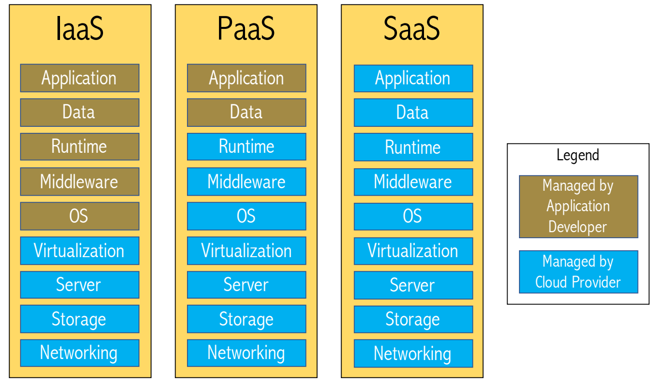


**Figure 2: Common cloud service models and their classifications relative what portion of the application stack is managed by cloud providers.**

*3.1.2 Cloud Resource Provisioning.* Since the demand for cloud resources is not fixed and can change over time, setting a fixed amount of resources results in either over-provisioning or under-provisioning, as depicted in Fig. 3. A foundation of cloud computing is based on provisioning only the required resources for the demand. This includes the use of virtualization for on-demand application deployment, and the use of resource provisioning to manage hardware and software in cloud data centers. Provisioning resources is an important topic in cloud computing that is widely explored. Since it is difficult to estimate service usage from tenants, most cloud providers have a pay-as-you-go payment scheme. As a result, providers can be more flexible on how to provision resources, and clients only pay for the amount of resources they actually use.

*3.1.3 Types of Cloud.* There are four types of cloud deployments: private cloud, community cloud, public cloud, and hybrid cloud [16]. Private clouds are designed for use by a singular entity and ensure high privacy and configurability. Private clouds are a good choice for organizations that require an infrastructure for their applications. This type of deployment is similar to traditional company-owned server farms and often do not benefit from a pay-as-you-go cost model. Community clouds are used by a community of users, and the infrastructure is shared between several organizations. A community cloud results in decentralized ownership of the cloud by multiple organizations within the community without relying on a large cloud vendor for the IT infrastructure. Public clouds are the typical model of cloud computing, where the cloud services are offered by cloud service providers, such as Amazon, IBM, Google, Microsoft, etc. Public clouds are generally more popular, easy-to-maintain, and cost-effective compared to private clouds. In contrast to private clouds, public clouds may benefit from the pay-as-you-go pricing model. However, public clouds do not always offer users complete customization of hardware, middleware, network, and security settings. Hybrid clouds are simply a combination of the cloud types mentioned above. Hybrid clouds allow users to have finer control over virtualized infrastructure, and combining the capabilities from different types of cloud deployments is accomplished through standardized or proprietary technology [19].

The cloud computing paradigm was initially established to allow users to access a pool of computing resources for ubiquitous computing. Even though cloud computing has helped bring forth accessible computing, the time required to access cloud-based applications may be too high and may not be practical for some mission-critical applications, or applications with ultra-low latency requirements. Also, the rapid growth in the amount of data generated at the network edge by an increasing number of connected devices requires cloud resources to be closer to where the data is

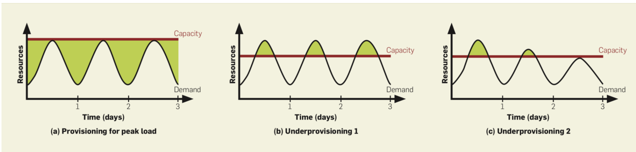


**Figure 3: Cloud provisioning is done based on the application demand [22].**

generated. Greater demand for high-bandwidth, geographically-dispersed, low-latency, and privacy-sensitive data processing has emerged – a quintessential need for computing paradigms that take place closer to connected devices and that support low-latency, high-bandwidth, decentralized applications. To address these needs, fog computing has been proposed by both industry and academia [6, 7]. In order to provide a detailed comparison among fog computing related paradigms, we introduce various computing paradigms, starting with fog computing.

## 3.2 Fog Computing

Fog computing bridges the gap between the cloud and end devices (e.g., IoT nodes) by enabling computing, storage, networking, and data management on network nodes within the close vicinity of IoT devices. Consequentially, computation, storage, networking, decision making, and data management not only occur in the cloud, but also occur along the IoT-to-Cloud path as data traverses to the cloud (preferably close to the IoT devices). For instance, compressing the GPS data can happen at the edge before transmission to the cloud in Intelligent Transportation Systems (ITS) [20]. Fog computing is defined by the OpenFog Consortium [6] as "a horizontal system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum." The "horizontal" platform in fog computing allows computing functions to be distributed between different platforms and industries, whereas a vertical platform promotes siloed applications [21]. A vertical platform may provide strong support for a single type of application (silo), but it does not account for platform-to-platform interaction in other vertically focused platforms. In addition to facilitating a horizontal architecture, fog computing provides a flexible platform to meet the data-driven needs of operators and users. Fog computing is intended to provide strong support for the Internet of Things.

*3.2.1 Fog vs. Cloud.* A common example that is often used to distinguished fog and cloud computing is whether latency-sensitive applications can be supported while maintaining satisfactory quality of service (QoS). Fog nodes can be placed close to IoT source nodes, allowing latency to be noticeably reduced compared to traditional cloud computing. While this example gives an intuitive motivation for fog, latency-sensitive applications are only one of the many applications that warrant the need for fog computing. Nodes in fog computing are generally deployed in less centralized locations compared to centralized cloud data centers. Fog nodes are wide-spread and geographically available in large numbers. In fog computing, security must be provided at the edge or in the dedicated locations of fog nodes, as opposed to the centrally-developed security mechanisms in dedicated buildings for cloud data centers. The decentralized nature of fog computing allows devices to either serve as fog computing nodes themselves (e.g. a car acts as a fog node for onboard sensors) or use fog resources as the clients of the fog.

The majority of differences between cloud and fog computing are attributed to the scale of hardware components associated with these computing paradigms. Cloud computing provides high availability of computing resources at relatively high power consumption, whereas fog computing provides moderate availability of computing resources at lower power consumption [23]. Cloud computing typically utilizes large data centers, whereas fog computing utilizes small servers, routers, switches, gateways, set-top boxes, or access points. Since hardware for fog computing occupies much less space than that of cloud computing, hardware can be located closer to users. Fog computing can be accessed through connected devices from the edge of the network to the network core, whereas cloud computing must be accessed through the network core. Moreover, continuous Internet connectivity is not essential for the fog-based services to work. That is, the services can work independently with low or no Internet connectivity and send necessary updates to the cloud whenever the connection is available. Cloud computing, on the other hand, requires devices to be connected when the cloud service is in progress.

Fog helps devices measure, monitor, process, analyze, and react, and distributes computation, communication, storage, control, and decision making closer to IoT devices [6] (refer to fig. 5). Many industries could use fog to their benefit: energy, manufacturing, transportation, healthcare, smart cities, to mention a few.

*3.2.2 Fog-Cloud Federation.* There are clear differences and trade-offs between cloud and fog computing, and one might ask which one to choose. However, fog and cloud complement each other; one cannot replace the need of the other. By coupling cloud and fog computing, the services that connected devices use can be optimized even further. Federation between fog and cloud allows enhanced capabilities for data aggregation, processing, and storage. For instance, in a stream processing application, the fog could filter, preprocess, and aggregate traffic streams from source devices, while queries with heavy analytical processing, or archival results could be sent to the cloud. An orchestrator could handle the cooperation between cloud and fog. Specifically, a fog orchestrator could provide an interoperable resource pool, deploy and schedule resources to application workflows, and control QoS [24]. Through the use of SDN, fog service providers will have greater control over how the network is configured with a large number of fog nodes that transfer data between the cloud and IoT devices.

*3.2.3 Fog RAN.* Fog computing can be integrated into mobile technologies in the form of radio access networks (RAN), to form what is referred to as fog RAN (F-RAN). Computing resources on F-RANs may be used for caching at the edge of the network, which enables faster retrieval of content and a lower burden on the front-haul. F-RAN can be implemented through 5G related mobile technologies [25]. On the other hand, cloud RAN (C-RAN) provides centralized control over F-RAN nodes. C-RAN takes advantage of virtualization, and decouples the base stations within a cell of the mobile network from its baseband functions by virtualizing those functions [26]. In

C-RAN a large number of low-cost Remote Radio Heads (RRHs) are randomly deployed and connected to the Base Band Unit (BBU) pool through the front-haul links. Both F-RAN and C-RAN are suited for mobile networks with base stations and are candidates for 5G deployments. Also, the use of F-RAN and C-RAN brings a more energy efficient form of network operation. We encourage the motivated reader to refer to reference [27] for more information about F-RAN.

Figure 4 shows a classification of computing paradigms related to fog computing and their overlap in terms of their scope. The figure illustrates our comparison of fog computing and its related computing paradigms. Table 1 lists the acronyms used for this figure and in the paper. We discuss the related computing paradigms in the order of their trend and show how some paradigms resulted in the emergence of others.

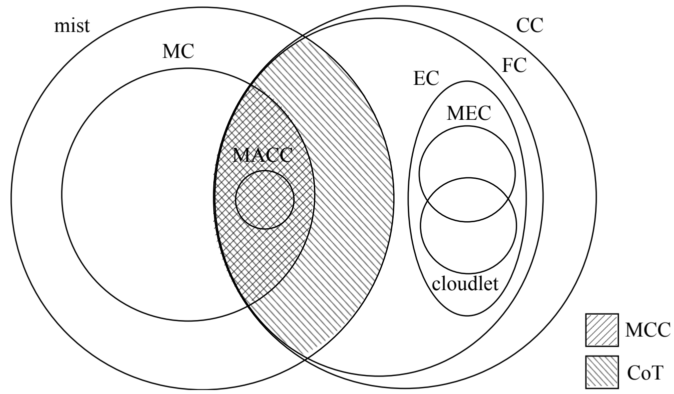


**Figure 4: A classification of scope of fog computing and its related computing paradigms. (Intersection of cloud computing and mist computing is Cloud of Things, and intersection of mobile computing and cloud computing is mobile cloud computing.)**

**Table 1: A list of computing paradigm acronyms used in this section.**

| | | | |
|---|---|---|---|
| IoT | Internet of Things | CC | Cloud Computing |
| MC | Mobile Computing | FC | Fog Computing |
| EC | Edge Computing | MEC | Multi-access Edge Computing |
| MCC | Mobile Cloud Computing | MACC | Mobile ad hoc Cloud Computing |
| CoT | Cloud of Things | mist | Mist Computing |

## 3.3 Mobile Computing

The advancement in fog and cloud computing is influenced by the groundwork set forth by the development of mobile computing. Mobile computing, or nomadic computing, is when computing is performed via mobile, portable devices, such as laptops, tablets, or mobile phones. Mobile computing can be utilized to create pervasive context-aware applications, such as location-based reminders.

At the heart of mobile computing is the vision for adaptation in an environment of low processing power and intermittent, sparse network connectivity. The peak of mobile computing technologies precedes cloud computing. A large number of fundamental challenges (such as user mobility, network heterogeneity, and low bandwidth) in mobile computing have been addressed in the literature before 2000. These issues have been addressed by advancements such as robust caching, transmission hardware and protocols, and compression algorithms [28]. Due to the evolving requirements of connected consumer devices, mobile computing alone is not suitable for many recent computing challenges.

With fog and cloud computing, computation is no longer tied to a local network; fog and cloud computing expand the scale and scope of mobile computing. The only type of hardware that mobile computing requires are mobile devices, which can be connected through Bluetooth, WiFi, ZigBee, and other cellular protocols. In contrast, fog and cloud computing require more resource-rich hardware with virtualization capabilities. Security in mobile computing must be provided on the mobile device itself. Compared to fog and cloud computing, mobile computing is more resource-constrained, but in recent years, advancements in mobile hardware and wireless protocols have significantly reduced this gap.

The power of mobile computing is from its distributed computing architecture. Distributed applications benefit from this architecture because mobile machines do not need a centralized location to operate. Mobile computing, however, comes with many drawbacks such as poor-resource constraints, the balance between autonomy and interdependence (prevalent in all distributed architectures), communication latency, and the need for mobile clients to efficiently adapt to changing environments [29]. These drawbacks often make mobile computing unsuitable for current applications that require low-latency or robustness, or that need large amounts of data to be generated, processed, and stored on devices.

## 3.4 Mobile Cloud Computing

As cloud computing matured, it became a valuable complement to mobile computing. This combination resulted in mobile cloud computing (MCC), which is defined as an infrastructure where both the data storage and data processing occur outside of the mobile device, bringing mobile computing applications to not just smartphone users but a much broader range of mobile subscribers [30]. NIST extends this definition to include mobile devices: cloud computing is the synergy between IoT devices, mobile devices, and cloud computing that enables data-intensive and CPU-intensive applications for IoT environments [31]. Some of these applications in MCC include crowdsourcing, healthcare, sensor data processing (such as optical character recognition and image processing), and task offloading [32, 33]. Mobile applications can be partitioned at runtime so that computationally intensive components of the application can be handled through adaptive offloading [34].

In mobile cloud computing, resource contained mobile devices can leverage resource-rich cloud services. MCC shifts the majority of computation from mobile devices to the cloud. MCC helps to run computation-intensive applications and to increase the battery life of mobile devices. MCC shares a blend of capabilities and characteristics in mobile computing and cloud computing. By adopting a combination of mobile computing and cloud computing objectives, high availability of computing resources is present in MCC as opposed to resource-constrained mobile computing. This allows for the emergence of high computation applications, such as mobile augmented reality. Also, the availability of cloud-based services in MCC is considerably higher than that of mobile computing. Similar to cloud computing and fog computing, MCC relies on cloud services for operating high-computation services. Computation in MCC can also be operated by mobile devices. Similar to cloud computing, security in MCC must be provisioned in both mobile devices and in the cloud. The authors in [35] design and implement an Android app that helps drivers find parking space availability using MCC.

MCC also suffers from the same limitations of mobile computing and cloud computing. First, while a centralized architecture in MCC is great for sharing a pool of computation resources, this may not be well suited for applications where pervasiveness of devices is desired. Second, since both cloud computing and MCC require cloud-based services, and as access to those services is through the network core by WAN connection, applications running on these platforms require connection to the Internet all the time. MCC shifts the majority of computation from mobile devices to the cloud, and this introduces connectivity challenges that were not present in mobile computing. Finally, offloading computation to the cloud causes the latency to be relatively high for delay-sensitive applications. The authors in [36] design a food recognition system based on MCC that distributes the data analytics between the mobile devices and the servers in the cloud. Mobile phones can perform light-weight computation on food images for food recognition, which allows the system to overcome some inherent limitations of traditional MCC paradigm, such as high latency and low battery life of mobile devices.

## 3.5 Mobile ad hoc Cloud Computing

Despite the pervasive nature of MCC, this computing paradigm is not always suitable for scenarios in which there is a lack of infrastructure or a centralized cloud. An ad hoc mobile network consists of nodes that form a temporary, dynamic network through routing and transport protocols; it is the most decentralized form of a network [37]. Mobile devices in an ad hoc mobile network form a highly dynamic network topology; the network formed by the mobile devices is highly dynamic and must accommodate for devices that continuously join or leave the network. Ad hoc mobile devices can form clouds that can be used for networking, storage, and computing. MACC could include use cases such as disaster relief, group live video streaming, and unmanned vehicular systems.

*3.5.1 MACC vs. Cloud Computing.* Mobile ad hoc cloud computing (MACC) is fundamentally different from cloud computing, mainly due to the ad hoc nature of the resources. MACC involves mobile devices that function as data providers, storage, and processing devices. Mobile devices in a mobile ad hoc cloud network are also responsible for routing traffic among themselves, because of the lack of network infrastructure. By pooling local mobile resources to form an ad hoc cloud, MACC offers reasonably high computation. These attributes differ from the target users, architecture, and connectivity in cloud computing. In a study done by researchers of Carnegie Mellon University [38], there is a tradeoff between offloading computation to distant clouds (labeled as “infrastructure cloud”) versus running them on nearby mobile devices (labeled as “mobile edge-clouds,” but in this paper we call them “mobile ad hoc clouds”). The authors compare the performance of executing some applications on a traditional infrastructure cloud versus running them in mobile ad hoc clouds.

*3.5.2 MACC vs. MCC.* MACC is also different from MCC in the hardware, service access method, and the distance from users, since computation is done on mobile devices in MACC, whereas it is far from mobile devices in MCC. MACC only requires mobile devices to operate, whereas MCC requires large-scale data centers used for cloud computing in addition to mobile devices. This results in high computation power, but also higher latency in MCC. Security in MACC must be provided only in mobile devices, but ensuring trust may be challenging in MACC without a secure collaboration framework. Finally, in MACC, services are only accessed through mobile devices that are connected via Bluetooth, WiFi, and other cellular protocols.

*3.5.3 MACC vs. Fog.* Although fog computing can be performed across a variety of resource-rich and resource deficient devices, mobile ad hoc cloud computing is better suited for highly decentralized, dynamic network topologies in which Internet connection is not guaranteed. Connected devices in MACC are more decentralized compared to fog computing, and this allows the devices to form a more dynamic network in places of sparely connected devices or a constantly changing network. An example of this is an ad hoc network for peer-to-peer file sharing [39].

*3.5.4 MACC vs. MANET.* A similar concept to MACC is a mobile ad hoc network (MANET). MANETs consist of mobile host devices that are connected to each other with single hop without base stations [40]. MANET devices form dynamic networks but do not necessarily form a cloud. In other words, the computing or storage resource pools are not necessary for MANETs. However, many solutions to MANETS, such as redundancy and broadcasting, can be applied to MACC. In a resource-constrained environment, peers may want to pool resources together to achieve a computationally demanding task that may not be feasible on a single mobile device. A use case for this is an unmanned vehicular system that consists of multiple unmanned vehicles and traffic devices.

## 3.6 Edge Computing

Similar to how MCC extends the capabilities of mobile devices, edge computing also enhances the management, storage, and processing power of data generated by connected devices. Unlike MCC, edge computing is located at the edge of the network close to IoT devices; note that the edge is not located on the IoT devices, but as close as one hop to them. It is worth noting that the edge can be more than one hop away from IoT devices in the local IoT network. OpenEdge Computing defines edge computing as computation done at the edge of the network through small data centers that are close to users [41]. The original vision for edge computing is to provide compute and storage resources close to the user in open standards and ubiquitous manner [41]. Edge computing is a crucial computing paradigm in the current landscape of IoT devices; it

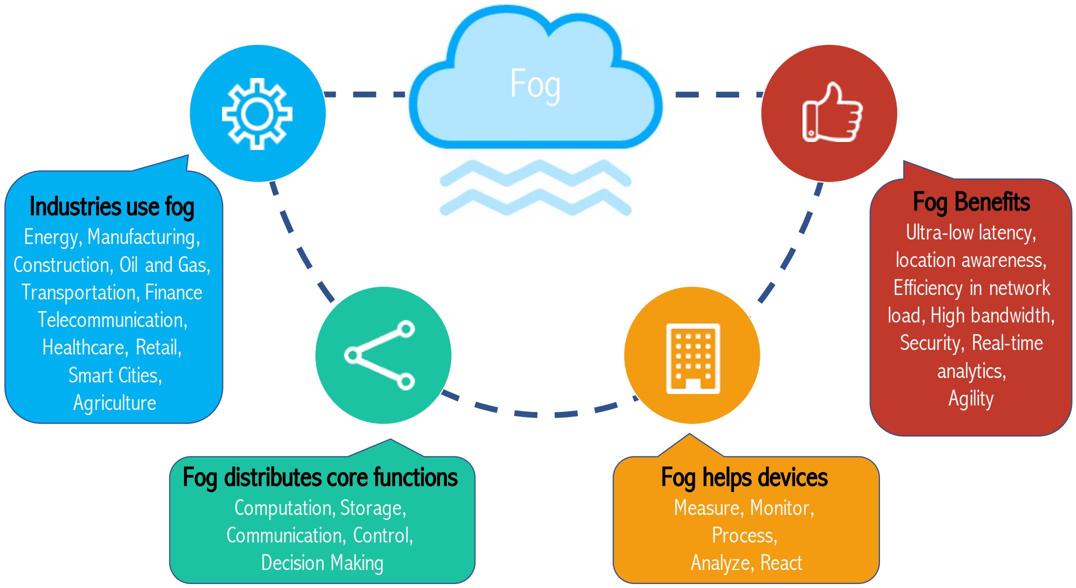


**Figure 5: Fog brings several benefits for the application developers, applications, and different industries by distributing the core functions.**

integrates the IoT devices with the cloud by filtering, preprocessing, and aggregating IoT data intelligently via cloud services deployed close to IoT devices [42].

Some issues that edge computing is well equipped to handle are privacy, latency, and connectivity. Due to its proximity to the users, latency in edge computing is typically lower than in MCC and cloud computing, if enough local computation power is provided; latency in edge computing can be slower than cloud or MCC if the local computation unit is not powerful enough. Service availability is also higher in edge computing because connected devices do not have to wait for a highly centralized platform to provide a service, nor are connected devices limited by the limited resources in traditional mobile computing. Compared to MACC, edge computing has small data centers, whereas MACC fundamentally does not need data centers. As a result, edge computing has higher service availability. Edge computing also can expand with broader computing capabilities than MACC by forming hybrid architectures with peer-to-peer and cloud computing models [43].

*3.6.1 Edge Computing vs. Fog Computing.* Although fog computing and edge computing both move the computation and storage to the edge of the network and closer to end-nodes, these paradigms are not identical. In fact, the OpenFog Consortium states that edge computing is often erroneously called fog computing; OpenFog Consortium makes the distinction that fog computing is hierarchical and it provides computing, networking, storage, control, and acceleration anywhere from cloud to things; while, edge computing tends to be limited to computing at the edge [6]. (Refer to Fig. 6.) Moreover, in a tutorial article [44] about fog and edge, the authors explain that "fog is inclusive of cloud, core, metro, edge, clients, and things," and "fog seeks to realize a seamless continuum of computing services from the cloud to the things rather than treating the network edges as isolated computing platforms," and "fog envisions a horizontal platform that will support the common fog computing functions for multiple industries and application domains, including but not limited to traditional telco services." [44]

*3.6.2 Where is edge?* It is worth mentioning that edge computing, cloudlets, fog computing, and mist computing (to be discussed in Section 3.9) are used interchangeably in some papers, as they all have "edge" as a common term. The term edge used by the telecommunications industry usually refers to 4G/5G base stations, RANs, and ISP (Internet Service Provider) access/edge networks. Yet, the term edge that is recently used in the IoT landscape [42, 45] refers to the local network where sensors and IoT devices are located. In other words, the edge is the immediate first hop from the IoT devices (not the IoT nodes themselves), such as the WiFi access points or gateways. If the computation is done on IoT devices themselves, this computing paradigm is referred to as mist computing (see Section 3.9). General Electric notes that fog computing focuses on interactions between edge devices (e.g., RANs, base stations, or edge routers), whereas edge computing focuses on the technology attached to the connected things (e.g., WiFi access points) [45].

## 3.7 Multi-access Edge Computing

Mobile cloud computing is an extension of mobile computing through cloud computing. Analogously, multi-access edge (MEC) computing is an extension of mobile computing through edge computing. ETSI defines MEC as a platform that provides IT and cloud-computing capabilities within the Radio Access Network (RAN) in 4G and 5G, in close proximity to mobile subscribers [46]. Multi-access edge computing was previously referred to as "mobile edge computing," but the paradigm has been expanded to include a broader range

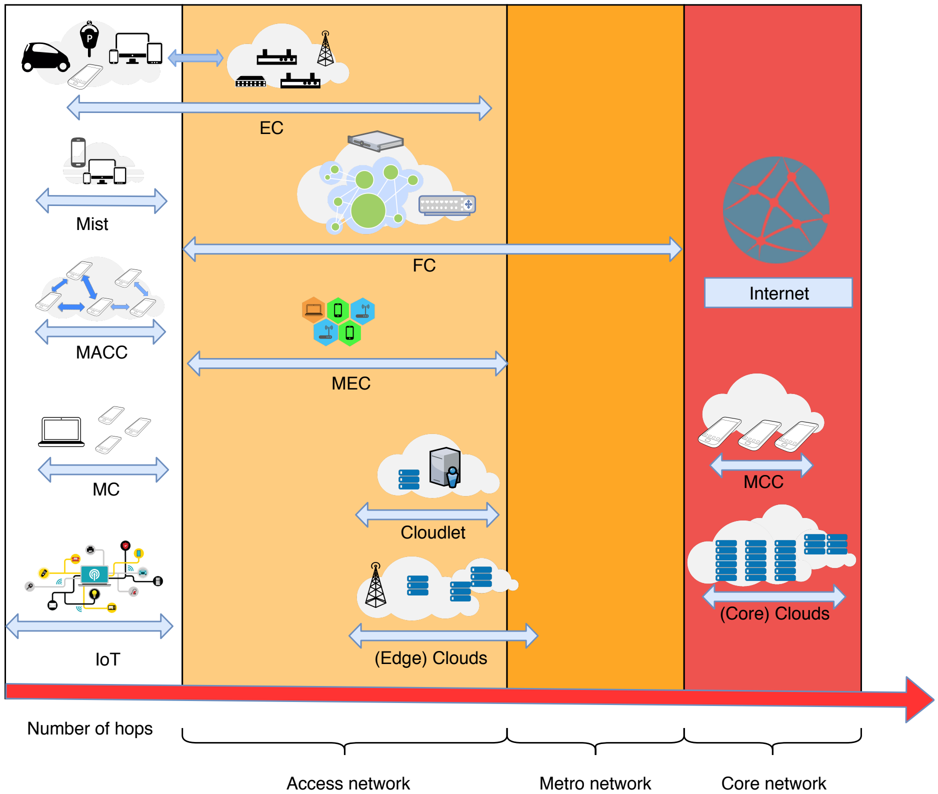


**Figure 6: Comparison of fog computing and its related computing paradigms in terms of their location and distance from the core clouds.**

of applications beyond mobile device-specific tasks. Examples of multi-access edge computing applications include video analytics, connected vehicles, health monitoring, and augmented reality.

MEC extends edge computing by providing compute and storage resources near low energy, low resource mobile devices. MEC allows RAN operators to add edge computing functionality to existing base stations. Similar to edge computing, small-scale data centers with virtualization capacity can also be used in MEC. Due to underlying hardware used in MEC and edge computing, available computing resources is moderate, in comparison to cloud computing. Furthermore, low-latency applications can be supported in MEC. MEC applications can benefit from real-time radio and network information hence can offer a personalized and contextualized experience to the mobile subscriber.

Both edge computing and MEC computing services operate from the edge of the Internet and can function with little to no Internet connectivity. MEC, however, establishes connectivity through a WAN, WiFi, and cellular connections, whereas edge computing generally can establish any form of connectivity (e.g., LAN, WiFi, cellular). MEC also primarily differs from MCC in its operations: Research in MCC focuses on the relationship between cloud service users (on mobile devices) and cloud service providers, whereas research in MEC focuses on (RAN-based) network infrastructure providers. MEC is expected to benefit significantly from the up-and-coming 5G platform [47]. Likewise, 5G is seen as an enabler of MEC as it allows for lower latency and higher bandwidth among mobile devices, and it supports a wide range of mobile devices with finer granularity.

MEC allows edge computing to be accessible to a wide range of mobile devices with reduced latency and more efficient mobile core networks [14]. MEC also allows for mission-critical delay-sensitive applications over the mobile network [47]. It has also incorporated the use of SDN and network function virtualization (NFV) capabilities, in addition to 5G technologies. SDN allows for virtual networking devices to be easily managed through software APIs [48], and NFV allows for reduced deployment times for networking services through virtualized infrastructure. Moreover, through SDN and NFV, network engineers and possibly enterprise application developers can develop their own orchestrator, whose goal is to coordinate the resource provisioning across multiple layers [49].

## 3.8 Cloudlet Computing

Proposed by Carnegie Mellon University, cloudlet computing is another direction in mobile computing that shares many traits with MCC and MEC. In fact, it addresses some of the disadvantages of MCC. A cloudlet is a trusted resource-rich computer or a cluster of computers with strong connection to the Internet that is utilized by nearby mobile devices [50]. Cloudlets are small data centers (miniature clouds, as the name suggests) that are typically one hop away from mobile devices. The idea is to offload computation from mobile devices to VM-based cloudlets located on the network edge [51]. Although academia mostly drives current studies in cloudlet computing, it has high potential in domains such as wearable cognitive assistance and web applications companies.

Cloudlet is the middle tier of a 3-tier continuum: mobile device-cloudlet-cloud. Given the nature of cloudlets as a small cloud close to mobile devices, operators for cloudlet computing could be cloud service providers who want their services to be accessible closer to mobile devices. Network infrastructure owners (e.g., AT&T, Nokia, etc.) can enable cloudlets with virtualization capacity to be situated closer to mobile devices, in smaller hardware footprints compared to the massive data centers used in cloud computing. The small footprint of cloudlets result in more moderate computing resources, but lower latency and energy consumption compared to cloud computing. Cloudlet computing is intended to serve devices in the local area.

Just as MACC greatly differs from cloud computing, it also highly differs from cloudlet computing. Cloudlet needs infrastructure with virtualization in the form of virtual machine (VM) capability, whereas MACC does not require such infrastructure. MACC and cloudlet computing both support mobility, but MACC is resource constrained and lacks virtualization support for real-time IoT applications. Cloudlets support local services for mobile clients by dividing tasks among cloudlet nodes in the proximity of mobile devices [52]. Although cloudlet computing fits well with the mobile-cloudlet-cloud framework [53], fog computing offers a more generic alternative that natively supports large amounts of traffic, and allows resources to be anywhere along the thing-to-cloud continuum. The concept of *mobile cloudlets* is similar to cloudlets, in which the cloudlets are a group of nearby mobile devices that are connected wirelessly, e.g., using WiFi or Bluetooth [54]. In mobile cloudlets, mobile devices can be providers as well as clients of computing service.

## 3.9 Mist Computing

Recently, mist computing has been introduced to capture a more extreme edge – the endpoints – of connected devices [55]. This computing paradigm describes dispersed computing at the extreme edge (the IoT devices themselves) and has been proposed with future self-aware and autonomic systems in mind [56]. Mist computing could be seen as the first computing location in the IoT-fog-cloud continuum; it can be informally labeled as "IoT computing" or "things computing." An IoT device may be wearable, a mobile device, a smart watch, or a smart fridge. Mist computing extends compute, storage, and networking across the fog through the things. In a sense, mist computing is a superset of MACC; since in mist, the networking may not be necessarily ad hoc, and the devices may not be mobile devices (refer to Fig. 4).

The authors in [57] introduce the idea of using mobile devices in the vicinity as a cloud computing environment for storage, caching, and computing purposes. They study the use of mist computing to reduce the load in traditional WiFi infrastructures for video dissemination applications. In this study, the spectators of a sport event organize themselves into WiFi-Direct groups and exchange video replays whenever possible, bypassing the central server and access points. This study is also another example of mist computing, in which IoT devices act not only as "thin clients," but also as "thin servers." Some other uses of mist computing are to preserve the privacy of users' data via local processing [58], and to efficiently deploy virtualized instances on single-board computers [59].

## 3.10 Other Similar Computing Paradigms

*3.10.1 Micro Data Center.* Cloudlet is sometimes referred to as micro data center (MDC) in some studies [60]. The term micro data center (MDC) was proposed by Microsoft Research in 2015 [60] and is defined as "an extension of traditional data centers used in cloud computing." An MDC can be an edge node or a cloudlet that is deployed between IoT devices and the cloud.

*3.10.2 Cloud of Things.* Another similar concept to mist computing is the Cloud of Things (CoT) [61], where IoT devices form a virtualized cloud infrastructure. In mist computing computation is done on IoT devices, possibly via message exchange, and not necessarily in a cloud of pooled resources. However, in Cloud of Things, computation is done over the cloud that is formed by pooled resources of IoT devices. Abdelwahab et al. [61] introduce the notion of Cloud of Things for sensing-as a service, which uses edge nodes as cloud agents sitting close to IoT nodes. The authors propose the idea of dynamically scaling up existing cloud resources (compute, storage, and network) by using the sensing capability of IoT devices. Edge nodes are used as cloud agents near the edge to discover, virtualize, and form a cloud network of IoT devices (CoT). This network is a geographically distributed infrastructure, in which cloud agents constantly discover resources of IoT devices and pool them as cloud resources. CoT enables remote sensing and in-network distributed processing of data. For instance, a cloud user can view pollution levels in cities from real-time temperature and $CO_2$ concentration sensors in vehicles with defined accuracy. The CoT framework is scalable to IoT networks, supports heterogeneity of IoT devices and edge computing nodes, and provides a foundation of sensing-as a service using fog computing.

Similar to CoT, the authors in [62] propose the concept of PClouds (personal clouds), which are distributed networked resources that are from both local/personal and remote/public devices and machines. PCloud can service end users even when remote cloud resources are not present or difficult to access due to insufficient network connectivity. Another novel idea similar to Cloud of Things and MACC is the work of the authors in [63], where they propose Cloudrone, an idea of deploying ad hoc micro cloud infrastructures in the sky using low-cost drones, single-board computers, and lightweight OS virtualization technologies. The drones in this scheme form a cloud computing cluster in the sky, which provisions the cloud services nearer to the user, even in the absence of a terrestrial infrastructure to access the remote cloud.

Similar to the concept of Cloud of Things, Femtoclouds have been proposed to tap into the computational capability and pervasiveness of underutilized mobile devices. Femtoclouds take advantage of clusters of devices that tend to be co-located in places such as schools, public transit, or malls. A hybrid edge-cloud workload management scheme is proposed in [64] for management of resources and tasks in femtoclouds, to provide low latency.

**Table 2: Attributes of fog-computing related paradigms**

| Attribute | CC | MC | FC | EC | MCC | MACC | MEC | cC | mist |
|---|---|---|---|---|---|---|---|---|---|
| **Users** | General | Mobile | General | General | Mobile | Mobile | Mobile | Mobile | General |
| **General Use Cases** | Scalable data storage, virtualized apps, distributed computing for large data sets (Google MapReduce) | Mobile sales transactions, location dependent queries (travel recommendations), multimedia applications on mobile devices | IoT, Connected vehicles, smart grid/smart city, health care, smart delivery (high-scale package drone delivery), real-time subsurface imaging, video surveillance | Local video surveillance, video caching, traffic control | Social networking, sensor data processing, health care (tele-monitoring and tele-surgery) | Networking and computing for disaster relief, group live video streaming, unmanned vehicular system | Content Delivery, Video analytics, connected vehicles, health monitoring, augmented reality | Optical character recognition (OCR), wearable cognitive assistance (Google Glass) | Parallel computation on IoT devices, autonomous vehicles, privacy-preserving local processing |
| **Operators** | Cloud service providers | Self-organized | Users and cloud service providers | Network infrastructure providers or local businesses | Users and cloud service providers | Self-organized | Network infrastructure providers (RAN-based) | Cloud service providers and network infrastructure providers | Self-organized or local businesses |
| **Service Type** | Global | Local | Less global | Local | Local | Local | Less global | Local | Local |
| **Hardware** | Large-scale data centers with devices with virtualization capacity | Mobile devices | Devices with virtualization capacity (servers, routers, switches, access points) | Edge devices with computing capability | Mobile devices or large-scale data centers with devices with virtual capability | Mobile devices | Small-scale data centers with devices with virtualization capacity, RAN in 4G and 5G | Devices with virtualization capability (micro and nano data centers) | IoT devices (e.g. sensors, cell phones, home appliance devices) |
| **Available Computing Resources** | High | Limited | Moderate | Moderate | High | Relatively less limited | Moderate | Moderate | Limited |
| **Main Driver** | Academia/ industry | Academia | Academia/ industry | Academia/ industry | Academia | Academia | Academia/ industry | Academia | Academia |
| **Distance from Users** | Far | Very close | Relatively close | Close | Far | Very close | Close | Close | Very close |
| **Main Standardization Entity** | CSA, DMFT, NIST, OCC, GICTF | MobileInfo | OpenFog Consortium, IEEE | — | NIST | — | ETSI, 3GPP, ITU-T | OpenEdge | — |
| **Application Type** | Ample computation | Distributed and mobile processing | High computation with lower latency | Low latency computation | High computation | High computation with lower latency | Low latency computation | High computation with lower latency | Distributed processing on IoT devices |
| **Architecture** | Centralized/ hierarchical | Distributed | Decentralized/ hierarchical | Localized/ distributed | Central cloud with distributed mobile devices | Distributed | Localized/ hierarchical | Localized | Localized/ distributed |

| Attribute | CC | MC | FC | EC | MCC | MACC | MEC | cC | mist |
|---|---|---|---|---|---|---|---|---|---|
| **Availability** | High | Low | High | Average | High | Low | Average | High | Low |
| **Latency** | Relatively high | Moderate | Low | Low | Relatively high | Moderate | Low | Low | Moderate |
| **Security** | Must be provided along cloud-to-things continuum | Must be provided on mobile devices | Must be provided on participant nodes | Must be provided on edge devices | Must be provided along cloud-to-things continuum and on mobile devices | Must be provided on mobile devices | Must be provided on edge network equipment (RAN, AP) | Must be provided on participant nodes | Must be provided on IoT devices |
| **Server Location** | Installed in large dedicated buildings | – | Can be installed at the edge or in dedicated locations | Near edge devices | Installed in large dedicated buildings | – | Can be installed at the edge | Near mobile devices | – |
| **Power Consumption** | Relatively high | – | Low | Low | Low on mobile devices | Low | High | Moderate | Low |
| **Internet Connectivity** | Must be connected to the Internet for the duration of services | Can operate with low or intermittent Internet connectivity | Can operate autonomously with no or intermittent Internet connectivity | Can operate autonomously with no or intermittent Internet connectivity | Requires Internet connection for offloading tasks or obtaining computation results from the cloud | Can operate autonomously with no or intermittent the Internet | May operate autonomously or connect to the Internet through RAN | Can operate with no or intermittent Internet connectivity; often requires connection to the Internet | Can operate with low or intermittent Internet connectivity |
| **Hardware Connectivity** | WAN | Bluetooth, WiFi, cellular, ZigBee | WAN, LAN, WLAN, WiFi, cellular | WAN, LAN, WLAN, WiFi, cellular, ZigBee | WAN | Bluetooth, WiFi, cellular, ZigBee | WAN, cellular | WAN, LAN, WLAN, WiFi, cellular | LAN, Bluetooth, WiFi, cellular, ZigBee |
| **Service Access** | Through core | Through mobile devices | Through connected devices from the edge to the core | At the edge of the Internet | Through core | Through mobile devices | At the edge of the Internet | Through resource-rich computers at the edge of the Internet | Through IoT devices |

*3.10.3 Edge Cloud.* When we talk about cloud computing, we mainly talk about "core" or "distant" clouds, which are far from the user or devices. Core clouds are further from connected things and are responsible for heavy computation. In contrast, "edge" clouds are smaller scale compared to core clouds and are closer to the devices. The concept of edge cloud [65] is similar to edge computing. The edge cloud extends cloud capabilities at the edge by leveraging user or operator-contributed compute nodes at the edge of the network. Similar to fog, in edge clouds the ability to run an application in a coordinated manner in both edge and the distant cloud is envisaged. Edge clouds are nodes at the edge, such as micro data centers, cloudlets, and MEC. [66].

Researchers have begun studying federation of both edge clouds and core clouds, and proposed the "osmotic computing" paradigm [67, 68]. Osmotic computing implies "the dynamic management of services and micro-services across cloud and edge data centers, addressing issues related to deployment, networking, and security" [67]. Osmotic computing utilizes both edge and cloud resources, each contained in two separate layers. Application delivery follows an osmotic behavior where virtualized micro-services are deployed opportunistically either in the cloud or edge layers. The ability to control how micro-services can be balanced between edge and cloud is a significant advantage of osmotic computing.

## 3.11 Concluding Remarks

The previous discussion about fog computing and related paradigms demonstrate the importance of understanding the characteristics of these platforms in the changing IT landscape. As demonstrated by the strength and weaknesses attributed to these computing paradigms, some paradigms may be better suited for a particular

**Table 3: Features of fog-computing related paradigms**

| Feature | CC | MC | FC | EC | MCC | MACC | MEC | cC | mist |
|---|---|---|---|---|---|---|---|---|---|
| Heterogeneity support | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| Infrastructure need | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| Geographically distributed | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Location awareness | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ |
| Ultra-low latency | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Mobility support | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Real-time application support | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ |
| Large-scale application support | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |
| Standardized | ✓ | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Multiple IoT Applications | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ |
| Virtualization support | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ |

use case than others. Even so, fog computing is suited for a large number of use cases in the current landscape of IoT and connected devices. The versatility of fog computing makes it suitable for many cases of data-driven computing and low-latency applications, even though it may not be suitable for a few extreme applications, such as disaster zones or sparse network topologies where ad hoc computing (e.g., MACC) or extreme edge clouds (e.g., mist, CoT) may be a better fit. Nonetheless, fog computing is considered a more general form of computing when compared to other similar paradigms (e.g., EC, MEC, cloudlet), because of its comprehensive definition scope, generality, and extensive presence along the thing-to-cloud continuum. Tables 2 and 3 summarize these characteristics. Fog computing offers a bright future for an open-standards environment of connected devices, as it is evident by IEEE Standard's adoption of OpenFog Reference Architecture [69].

There does not yet exist a globally unanimous distinction between fog computing and related computing paradigms, such as edge computing, mist computing, and cloudlets across researchers and industries, as shown in the previous sections of this paper. We attempt in this survey paper to clarify the distinctions between fog computing and the related computing paradigms. A comparison of the underlying infrastructure of fog computing and its related computing paradigms from the networking perspective is shown in Fig. 7. In the rest of this paper, we will mainly survey and discuss the recent literature on fog computing, but mention the studies on other related computing paradigms that could be easily extended or directly applied in fog.

## 4 TAXONOMY OF FOG COMPUTING

In this section, we will introduce a taxonomy of the research in fog computing that is the basis of this survey[2]. This taxonomy categorizes the research articles that focus primarily on fog computing from the networking perspective. We have also included research articles from other similar computing paradigms, such as edge computing, if the article is relevant and general enough that it could be easily extended to fog computing. The taxonomy is shown in Fig. 8, and the papers on fog computing, edge computing, cloudlet, etc. that are referenced in this survey are categorized into different categories in Table 4. We will discuss the literature on fog computing in this section, and we categorize the research papers according to this taxonomy. Moreover, we have rigorously checked the objectives of papers (e.g., QoS improvement) and the features they provide (e.g., scalability), and we summarize them in Table 6. The explanation of the features and objectives along with several examples for each objective/feature are included in Table 5. We extract these objectives/features such that they are comprehensive and useful for a fog system design, and are also closely in compliance with the pillars of OpenFog architecture [6].

The "Foundation" category consists of the research papers that either survey the fog computing area, or try to define and standardize the field of fog computing. The "Frameworks and Programming Model" category is where the reader can find research articles that introduce frameworks, architectures, and programming models for fog, or that use fog computing to introduce a new concept (such a vehicular fog computing). The next category is "Design and Planning," which includes the papers that discuss the design and planning of the network and computing infrastructure. The "Resource Management and Provisioning" category consists of the research papers that study the management and provisioning of the resources (e.g., service provisioning, VM placement, control and monitoring). The category "Operation" includes the papers that discuss operational aspects of fog computing systems (e.g., task scheduling, load balancing, and resource discovery). Each of the mentioned categories has subcategories, and we will discuss each subcategory in the following subsections. The category "Software and Tools" will list papers that focus on software, simulators, and tools for fog computing. Likewise, the "Hardware and Protocol Stack" category showcases articles that propose a protocol stack or introduce particular hardware for fog computing. Papers that focus on developing testbeds or doing extensive experiments for

[2]The data (categories and features/objectives of the papers) of this survey are available in the form of several datasets at https://github.com/ashkan-software/fog-survey-data

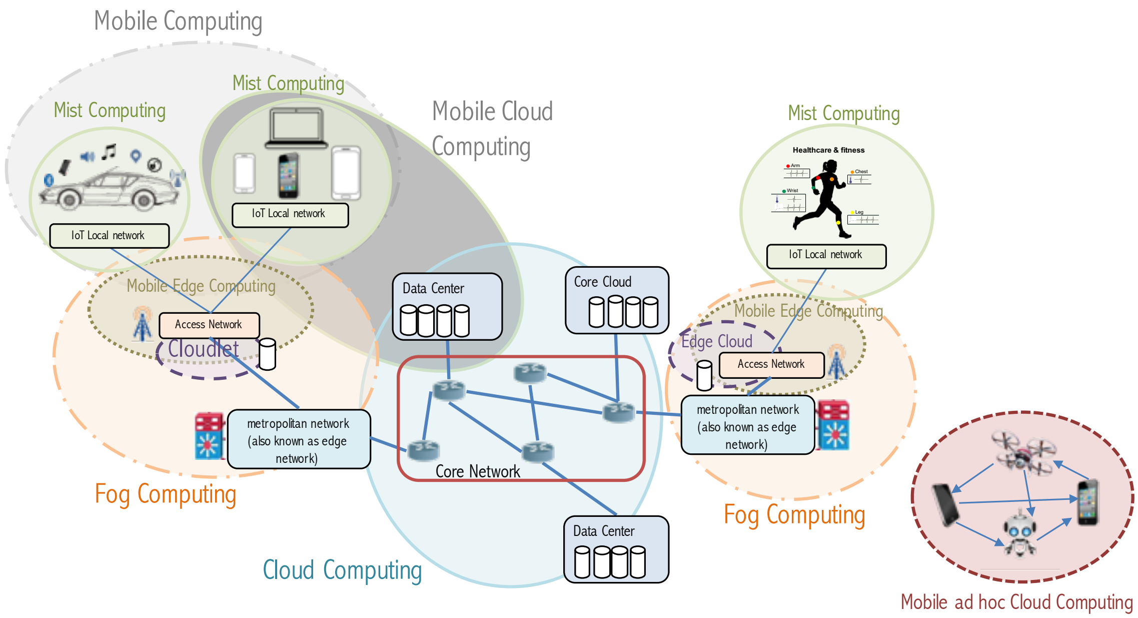


**Figure 7: Comparison of the infrastructure of fog computing and its related computing paradigms from the networking perspective.**

fog are summarized under the "Testbeds and Experiments" category. Finally, papers discussing security and privacy aspects of fog computing are included under the "Security and Privacy" category.

Note that the three categories Design and Planning, Resource Management and Provisioning, and Operation reflect conventional steps in design and operation of distributed computing systems and computer networks. First, in the Design and Planning step, network designers estimate and analyze the required resources for a given network, design the infrastructure and topology of the network, and determine the hardware and resources that must be placed in a particular design. Next, in the Resource Management and Provisioning step, network operators try to manage and provision the resources for better utility and efficiency. For instance, service orchestration and migration methods are used to intelligently allocate and provision the available resources of the network nodes; monitoring techniques are used to monitor the resource usage of the nodes, for instance for placement decisions (e.g., VM or container placement). Finally, in the Operation step, the final improvements for resource usage and efficiency are performed, such as task offloading and scheduling, load balancing, and efficient resource discovery.

## 4.1 Foundations: Definition and Standards

In this subsection, we survey the articles that are concerned with defining and standardizing fog computing and concepts related to fog computing. The very definition of fog computing and fog nodes is a topic of ongoing discussion, and there is no common consensus on what a fog node is [70]. There are some early efforts to define fog and fog nodes [6, 70]. OpenFog Consortium is one of the pioneers in standardizing and defining fog computing. The OpenFog architecture is established to provide a nonproprietary fog architecture and standard to support current cloud computing in addition to diverse IoT and edge-oriented ecosystems. The white paper introduces security, scalability, openness, agility, among other "pillars" of an open fog architecture [6]. Later, IEEE Standards Association adopted OpenFog Consortium's reference architecture as a standard for fog computing through IEEE 1934 [69].

Vaquero et al. [71] take into account mobile device ubiquity, network management, fog network connectivity, and privacy to propose their definition for fog: *a large amount of heterogeneous, ubiquitous, and decentralized devices that can cooperate to form a network for storage and processing without third-party intervention.* The authors in [72] focused on the theoretical modeling and performance metrics of the fog computing architecture. They propose a mathematical formulation for fog computing by defining its components for a generic fog architecture.

Current communication technologies and standards that could be used in fog networks are presented in [73]. In the paper, a classification of layers and technology settings related to IoT and fog computing is described. On the other hand, the authors in [74] define "class of service" for fog applications, a classification of fog services according to their QoS requirements. They also introduce a mapping between certain classes of services and the corresponding processing layers of the fog computing reference architecture.

With emerging availability of IoT devices and their large volume of data that they produce, timely and reliable transfer of large data streams to a centralized location is a requirement of deep learning models. The authors in [68] introduce "deep osmosis" and analyze the research challenges involved with developing edge-cloud-based holistic distributed deep learning algorithms and their corresponding resource models and architecture for cloud and fog computing.

The study in [75] defines the concept of "content island" for fog computing, which interconnects groups of devices to interchange data and processing among themselves and with other content islands. The islands are based on the integration of a publish/subscribe system with disruption-tolerant network (DTN) techniques to provide higher flexibility with respect to data and computation sharing. Another definition in fog computing area is the "reliability factor" of a node, which defines the probability of a node being online or available, and is defined in [76]. In [77] ten economic aspects of fog computing (referred to as fogonomics) are introduced.

## 4.2 Foundations: Surveys

In this section, we discuss the previous work in the fog computing domain that are of survey or tutorial nature. Several comprehensive survey papers are already discussed in Section 2; in this subsection, we will discuss some other survey papers that are not as comprehensive as the ones discussed in Section 2. Klas from the Vodafone Group presents a clear picture of edge computing and its benefits [78]. This paper highlights relevant edge computing applications and similar research areas (e.g., fog and cloudlet), surveys the latest industrial efforts and available edge computing technologies, and discusses edge computing's potential to improve the telecommunications industry.

Historically, the field of computing has seen cycles between decentralization and centralization [43]. The authors of [43], in a survey study, advocate for edge-centric computing, a more decentralized paradigm that utilizes peer to peer (P2P) networking at the edge of the network while maintaining access to the cloud.

The authors in [79] present a comprehensive study of edge computing, starting with the factors leading up to its development, the advantages of edge computing, requirements for successful implementation, application use-cases, and challenges. Several use cases, such as, gaming, real-time image processing, smart grid, smart transportation, are used to emphasize the range of problems that edge computing can help alleviate. Weisong et al. define edge computing and survey its use cases, research issues, and future research directions in [80]. A recent survey on MEC can be found in [81].

Stojmenovic and Wen in [82] highlight privacy and security concerns of fog computing gateways as major issues. These issues include man-in-the-middle attacks and lack of encryption in gateways that serve as fog nodes. The study in [83] surveys the main features of fog computing, describes its architecture and design goals, and discusses some potential issues of fog computing in 5G. One of the early surveys in fog architecture and taxonomy is the work of Zhang and Chiang [84]. The authors further describe the IoT challenges for which fog computing can provide solutions. The survey in [85] distinguishes and explains edge computing, fog computing, and cloud computing. The study reviews various system architectures, application characteristics, and platform abstractions of fog, edge, and cloud.

In the paper [86], the authors overview fog computing model architecture, key technologies, and applications. They present the hierarchical architecture of fog computing and its characteristics and compare it with cloud computing and edge computing. Then, the key technologies for fog are introduced to see how they support fog computing deployments.

## 4.3 Frameworks and Programming Models: Architectures and Frameworks for Fog

Many researchers have independently proposed various architectures and frameworks for fog computing. In this section, we summarize the previous work that have proposed general architectures or frameworks for fog computing.

*4.3.1 General Architecture for Fog Computing.* A recent study suggests an architectural model for combining MEC and fog computing for 5G networks [87]. The authors claim that fog computing and MEC separately have weaknesses and incompleteness; they further claim the need for convergence of the two computing paradigms for overcoming such limitations. A three-layer general logical architecture for fog computing is introduced in [88] and [89]. The layers are IoT, fog, and cloud, where each layer is partitioned into domains. Similarly, a three-layer architecture including the cloud, MEC, and IoT is proposed in [90]. In the three-layer architecture proposed, the user plane consists of mobile users and IoT devices, the edge computing plane consists of edge servers in close proximity to the users, and the cloud computing plane is the core of the network and contains multiple cloud servers and data centers. Comparably, we propose our three-layer architecture for fog computing in Fig. 9.

The authors in [91] propose a fog-to-cloud architecture, consisting of a layered management structure that can bring together different heterogeneous cloud/fog layers into a hierarchical architecture. The paper [92] designs a hierarchical edge cloud architecture, to efficiently utilize the cloud resources for serving the peak loads from mobile users. The proposed architecture consists of servers at the edge, which directly receive workloads from mobile devices via wireless links. These edge servers are connected to higher tiers of edge cloud servers and remote data centers through the Internet backbone. Different from directly using a flat collection of edge cloud servers, the proposed architecture aggregates the peak loads that exceed the capacities of lower tiers of edge cloud servers to other servers at higher tiers in the edge cloud hierarchy.

*4.3.2 Fog Computing Resource Model.* One challenge in fog computing is defining who the fog resource providers are. Is it the case that fog service providers must provide fog resources? Can end users can bring their devices and share their resources? Do network providers offer their edge resources for renting? The articles [93] and [94] present a unified computing, caching, and communication (3C) solution for 5G that allow service, content, and function providers to deploy their services/content/functions near the end users. The solution also allows for the exploitation of extreme edge resources by enabling their owners to form virtual fogs (vFogs) cooperatively; that is, end users will have the ability to become 3C resource providers to the 5G ecosystem. The authors also propose their architecture for fog computing, which consists of vFogs, hyper fogs (constellations of vFog networks to facilitate processing and data exchange that requires resources from more than one


**Figure 8: Taxonomy of fog computing that is used in this survey.**

vFog), super extreme edge node, regular extreme edge node, and orchestrator.

One of the many efforts to design a reference framework and infrastructure for the fog-based IoT that considers resource sharing of consumers is Indie Fog [95]. The Indie Fog infrastructure utilizes consumers' equipment (e.g., WiFi access points) to provide fog services for IoT devices. The authors suggest that network infrastructure providers or cloud service providers can make use of the consumer premises equipment to provide their fog-based services. Under this model, they claim that consumers will be willing to share their equipment with the providers for offering their services. Indie Fog uses the general fog architecture proposed by the OpenFog Consortium [6] and adds the Indie Fog services to it, which are interconnected via virtual connections to private and public fog networks.

*4.3.3 Fog Architecture Design Decisions.* In an article from Cisco [96], the author proposes a general high-level architecture for fog networks, fog software, and fog nodes. This paper is an early attempt to review and characterize the design/decision parameters of fog networks. The author names these decision criteria "fog architectural imperatives," and discusses them in detail. The fog architectural imperatives are decisions related to design requirements that are difficult to implement on networks with sole reliance on cloud or IoT devices, and that can only be satisfied by using fog resources [96].

*4.3.4 Fog Architectures for 5G and IoV.* Fog computing is seen as a promising enabler for some of the emerging paradigms, such as 5G, autonomous cars, and Internet of Vehicles (IoV). In their article [158], the authors propose an SDN-based framework for cloud-fog interoperation in 5G wireless networks. Vilalta et al. propose TelcoFog - a fog computing architecture that is deployed at the network edge for telecom operators to provide cost-effective 5G services for low latency and scalability [135]. TelcoFog consists of three main types of components: scalable TelcoFog nodes, TelcoFog controller, and TelcoFog services. The paper [145] introduces the challenges of handling big data in the IoV environments. The authors emphasize on the role of fog servers and describe a regional cooperative fog computing (CFC) architecture to support IoV applications. The proposed CFC-IoV architecture consists of two layers - the fog layer and edge layer. The fog layer is a federation of geographically distributed fog servers, a coordinator server, and the cloud servers, whereas the edge layer includes the vehicular ad hoc network (VANET), IoT applications, and mobile cellular networks. Other effort suggesting fog architectures for 5G or IoV are [83, 87, 93, 128, 134].

*4.3.5 ICN-based Fog Architecture.* The study in [143] brings together fog computing and information-centric networking (ICN),

**Table 4: Overall categories of the papers cited in this survey**

| Category | Subcategory | Papers |
|---|---|---|
| Foundations | Surveys | [8, 9, 13, 43, 67, 71, 78–86, 97–105] |
| | Definitions & Standards | [6, 66, 68, 70–77, 85, 87, 106–123] |
| Frameworks and Programming Models | Architectures and Frameworks for Fog | [6, 27, 70–72, 75, 77, 83, 87–96, 106, 109, 110, 112, 115, 124–170] |
| | Concepts and Frameworks using Fog | [61, 64, 118, 123, 171–193] |
| | Programming Models and Data Modeling | [98, 107, 132, 134, 135, 143, 184, 194–209] |
| Design and Planning | Infrastructure Design | [27, 121, 128–130, 134, 210–220] |
| | Resource Analysis and Estimation | [23, 76, 113, 117, 119, 121, 211, 216, 221–230] |
| Resource Management and Provisioning | Service Provisioning (Orchestration & Migration) | [24, 126, 132, 135, 138, 139, 145, 156, 162, 169, 179, 197, 217, 231–254] |
| | Placement (VM/Service) | [92, 110, 149, 155, 157, 166–168, 174, 184, 217, 255–278] |
| | Control and Monitoring | [9, 54, 106, 127, 129, 130, 134, 135, 137, 151, 160, 238, 239, 245, 259, 279–287] |
| Operation | Scheduling, Offloading, and Load Balancing | [54, 64, 88, 90, 111, 124, 131, 140, 145, 160, 163, 164, 186, 210, 212, 218, 219, 230, 235, 255, 262, 286, 288–321] |
| | Resource Discovery | [61, 116, 123, 138, 143, 196, 197, 204, 221, 239, 290, 310, 322–329] |
| | Applications | [20, 51, 61, 100, 154, 159, 174, 175, 190, 191, 247, 263, 271, 276, 287, 306, 320, 321, 323, 330–379] |
| Software & Tools | | [66, 150, 195, 207, 240, 322, 344, 355, 380–396] |
| Testbeds & Experiments | | [123, 139, 142, 169, 173, 185, 193, 348, 376, 386, 394, 397–418] |
| Security & Privacy | | [66, 82, 99, 102, 104, 105, 108, 114, 163, 165, 178, 282, 283, 304, 313, 326, 337, 339, 341, 349, 364, 368, 371, 373, 389, 419–433] |
| Hardware & Protocol Stack | | [27, 75, 116, 122, 128–131, 146, 152, 158, 164, 170, 171, 176, 179, 200, 212, 214, 219, 266, 322, 384, 408, 409, 434, 435] |

which enables flexible and efficient data distribution at the network layer. In the introduced ICN-Fog architecture, at the lowest layer are heterogeneous end devices that connect to fog nodes, which run ICN-specific protocols to communicate with other fog nodes. Apart from connecting to other fog nodes, each fog node is also connected to the cloud. The authors note that ICN-Fog relies on the principles of ICN for building smart, horizontal fog-to-fog data communication that leads to reducing the application's dependency on the cloud and distributed processing in the fog. Similarly, the authors of [100] explore the idea of combining Information-Centric Networking(ICN) with MEC to address mobility related issues in the MEC approach that relies heavily on the underlying host-centric networking model.

*4.3.6 Resource Allocation Frameworks.* Sun and Ansari introduced EdgeIoT, a hierarchical architecture that aims to allocate resources through the use of VMs while maintaining user privacy [106]. Sun and Nirwan use OpenFlow SDN switches to provide network management for aggregated data from IoT devices. The authors of [110] propose a hierarchical MEC architecture for resource allocation in MEC. The architecture introduces the notion of field, shallow, and deep cloudlets, where the field cloudlets are collocated with the base stations, the shallow cloudlets are at aggregation points, and the deep cloudlet is at the mobile backhaul. The architecture can handle peak loads efficiently by utilizing the shallow and deep computing facilities at higher levels when the computing capacity of a field cloudlet is not enough to handle the loads from its corresponding mobile users.

Lingen et al. [134] focus on a unified approach for computing in fog and cloud computing. They argue that fog computing and cloud computing should not be complementary paradigms, but instead should be fused together. As a result, through the authors' architecture, compute nodes in the fog and cloud have the same architecture, and resources are managed in a unified way. The architecture is extended from the European Telecommunications Standards Institute (ETSI)'s standardized reference architecture for NVF management and orchestration (MANO).

## 4.4 Frameworks and Programming Models: Concepts and Frameworks using Fog

Several studies utilized the concept of fog computing to propose new concepts, ideas, and frameworks based on fog computing.

*4.4.1 Vehicular Fog Computing.* The authors in [172, 176] proposed the idea of vehicular fog computing (VFC) by utilizing vehicles as the infrastructures for communication and computation. VFC takes advantage of a dynamic group of vehicles to help increase computational power and decrease latency issues. Different from the vehicular cloud computing, the proposed VFC supports geo-distribution, local decision making, and real-time load-balancing. Moreover, VFC depends on the collaboration of near-located vehicles, instead of relying on the remote servers, which reduces the deployment costs and delay.

An architecture for VFC is presented in [176] and is comprised of three layers: the application and services layer, the policy management layer, and the abstraction layer. The application and services layer offers a variety of real-time applications as well as new services to users, whereas the policy management layer is responsible for allocating resources to the tasks and handling basic issues such as monitoring system state dynamically. The abstraction layer is responsible for managing, provisioning, and interfacing with the physical resources and for the security and privacy of the VFC architecture. The benefits, architecture, use cases, and potential issues of VFC are presented in [178]. The authors proposed a high-level architecture of vehicular fog computing, which comprises of three types of entities, namely smart vehicles as the data generation layer, roadside units/fog nodes as the fog layer, and cloud servers as the cloud layer.

Similar to VFC, unmanned aerial vehicles (UAVs) have been considered as means to provide computing capabilities [186]. In this model, UAVs act as fog nodes and provide computing capabilities with enhanced coverage for IoT nodes. Similarly, the concept of vehicular micro clouds based on map information is introduced, and, by a simulation study, investigated in [183]. Vehicular micro clouds are virtual edge servers and are essentially clusters of cars that help to aggregate and preprocess data that is transferred to the cloud.

The study in [177] argues that cloud and fog computing using the current mobile networks may not be ideally suited to provide the desired levels of QoS for moving electric vehicles in vehicle-to-grid (V2G) services. They propose a hybrid computing model called "Foud," in which the cloud and fog come together and are made available to the V2G systems. In the proposed model, the cloud allows virtualized computing, storage, and network resources to be available to the V2G system entities, whereas the fog temporarily integrates the stationary and mobile computing resources located at the edge of V2G networks to expand the computing capacity.

*4.4.2 Beyond Conventional Fog Nodes.* Prazeres et al. [173] proposed a new paradigm called fog of things (FoT) which uses fog computing platforms for the IoT. The authors note that, in the proposed FoT, IoT services are defined at the edge of the network and are distributed through message and service-oriented middleware. Additionally, fog of things is self-organized, consists of FoT devices, FoT gateways, and FoT servers, and can deliver IoT services in a distributed manner. With the described FoT paradigm, the authors further propose a platform for the actual implementation of the FoT paradigm. The authors in [185] propose human-driven edge computing (HEC) as a new model to ease the provisioning and to extend the coverage of traditional fixed MEC solutions by utilizing devices that humans carry.

The study in [175] looks at the latency issues that may be experienced by delay-sensitive IoT applications due to the geographical distances between the physical IoT devices and the data centers. The authors consider the mobile IoT federation as a service (MIFaaS) paradigm that leverages the pool of devices managed by individual cloud providers as a whole in order to help support delay-sensitive applications. The network model considered is a cellular IoT environment with multiple LTE femtocells as the network edge nodes that supports the MIFaaS paradigm.

*4.4.3 Fog for Transparent Computing.* The paper [179] examines the question of how to leverage transparent computing to build scalable IoT platforms and proposes a tailored, transparent computing architecture for IoT applications. Transparent computing eliminates the dependency of hardware and software and allows the provisioning of cross-platform and on-demand services on lightweight IoT devices. The proposed architecture consists of several layers - end user layer, edge server layer, core network layer, cloud layer, and the management and interface layer. The end user layer consists of a variety of IoT devices, and the edge server layer is responsible for distributing computing, control, and storage functions to end user devices at the edge of the network. The core network layer is the core of the Internet, and the cloud layer is composed of a cluster of servers with massive computing and storage resources.

*4.4.4 Volunteer Edge Computing.* Researchers from the University of Minnesota studied the possibility of using volunteer resources near the edge for both computing and storage, and proposed Nebula. Some existing systems that exploit volunteer edge computing and data sharing are Grid and peer-to-peer systems such as BitTorrent, BOINC [436], and SETI@home [437]. While these volunteer platforms either are for compute-intensive applications (such as BOINC and SETI), or file-sharing systems (e.g., BitTorrent), Nebula supports distributed data-intensive applications through a close interaction between compute and storage resources [182]. Nebula utilizes edge resources for in-situ data-intensive computing, through location-aware data and computation placement, replication, and recovery.

*4.4.5 Path Computing.* Path computing is paradigm based on the fog computing paradigm, where a multi-tier cloud architecture is deployed over the geographic span of the network. Path computing provides storage and compute resources along a succession of data centers of increasing size, located between the IoT devices and the cloud data centers, and enables the deployment of a multi-level hierarchy of data centers along the path that traffic follows [184]. The authors in [184] propose path computing and CloudPath (a platform for path computing). CloudPath enables dynamic installation of light-weight stateless functions, and a distributed eventual consistent storage system. CloudPath also automatically migrates application data across data centers to minimize latency and bandwidth usage.

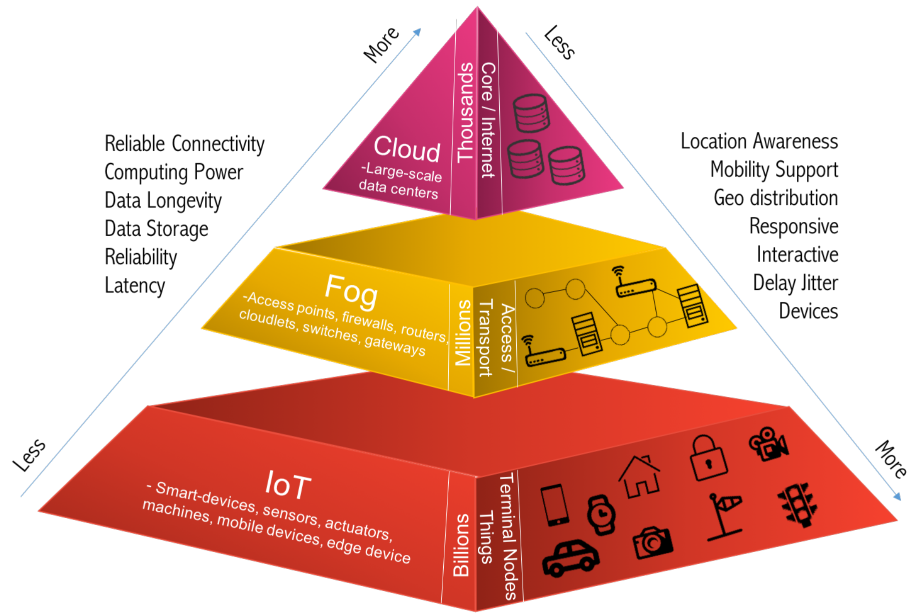


**Figure 9: Our three-layer architecture for IoT-fog-cloud ecosystem.**

*4.4.6 Fog Nodes as IoT Hubs.* A proposed application of fog nodes to allow for the interoperability of heterogeneous IoT devices is presented in [171]. These specialized fog nodes are referred to as "IoT Hubs." IoT devices have different computational power and different energy requirements; thus they are restricted to different communication protocols. The IoT Hub serves as a bridge between all the different physical networks and then merges them all using an all-IP network.

## 4.5 Frameworks and Programming Models: Programming Models and Data Modeling

In this subsection, we survey the studies that introduce and propose programming models or data modeling tools for fog computing.

*4.5.1 Distributed Data Modeling and Frameworks.* With the rise of IoT, the demand for distributed big data analytics follows; however, data is rarely shared between stakeholders due to resource, security, and privacy concerns. Zhang et al. [194] proposed a new programming model, called Firework, that takes advantage of fog computing to enable safe and reliable distributed data sharing between stakeholders. Firework merges geographically distributed data through the creation of virtually shared data views that end users can access via interfaces defined by the data owners. An example use case of Firework proposed by Zhang et al. is the shared data from security cameras. Security cameras from different sources in a common geographical location can be a part of a single Firework instance. Police departments can then request access to a specific footage at a particular time from the datasets in the Firework instance to track a person or object of interest.

One fundamental question in fog computing is the distribution of tasks between fog nodes and the cloud. How should the workload be balanced? Should the user be able to choose the balance? If yes, how so? The authors in [98] propose the design of WM-FOG, a flexible software architecture that allows developers to customize policies regarding the workflow. WM-FOG allows the developer to define the synchronization policy, to choose how much data is sent to the back-end cloud and how much data is sent to the fog node.

The study in [198] focuses on the distributed deployment of process-aware IoT applications. The authors propose a mechanism to design a distributed IoT application in one place and annotate the different components of the application which are supposed to be deployed at different edge nodes with their location information. After defining deployment locations, the model identifies application fragments which need to be deployed in different locations, and annotates each fragment with the deployment location. It then decomposes the process-aware IoT application into a set of fragments based on the annotated locations. Crystal [205] provides an abstraction for fault-tolerant and distributed fog application development. A fog application using Crystal can take full advantage of location transparency, self-healing, auto-scaling, and mobility support.

*4.5.2 Fog Programming Models.* The paper [107] studied QoS-aware application deployment in fog infrastructure from the programming perspective. They present a prototype to support application deployment in the IoT-fog-cloud scenario[3]. As a future direction, one could consider other QoS attributes and include cost information to get a richer classification of eligible fog deployments. Moreover, one could account for multiple and multi-tenant deployments on the same fog infrastructure. Renart et al. proposed an edge-based programming framework that enables users to define data-driven reactive behaviors based on the content and source of data streams

[3] Available at https://github.com/di-unipi-socc/FogTorch

[202]. Users define how data stream must be processed on the edge based on their content and the location of the data source.

Saurez et al. propose Foglets, a programming model that facilitates distributed programming across fog nodes [197]. Foglets provides APIs for spatio-temporal data abstraction for storing and retrieving application-generated data on the local nodes. Through the Foglets API, Foglets processes are set for a certain geospatial region and Foglets manages the application components on the Fog nodes. Foglets is implemented through container-based visualization. The Foglets API takes into account QoS and load balancing when migrating persistent (stateful) data between fog nodes.

*4.5.3 Fog Software Stack for Android.* The authors in [195] have developed a software stack for Android to implement fog computing by using smartphones as fog nodes. To do this, several challenges need to be considered. For example, mobile operating systems typically do not have priorities when dealing with memory allocation, it is hard to track the resources used by each app, and it is difficult to ensure adaptability with current software. The proposed software stack is split into four components, each handling the above challenges.

*4.5.4 PaaS for Fog.* Current platform as a service (PaaS) models are intended for traditional cloud applications that are not latency-sensitive nor large-scale. However, using fog computing can be difficult due to the orchestration of highly dynamic and heterogeneous fog resources. Hong et al. [199] propose a PaaS model called Mobile Fog that provides a programming abstraction and allows applications to easily use fog resources while supporting dynamic scaling at runtime. The Mobile Fog communication API is composed of event handlers that must be implemented and standard functions that can be called by the application. The same code can be run on a different device like smartphones, vehicles, or cameras; the developer only needs to write the code once.

Similarly, the authors in [132] have proposed a PaaS architecture for automating application provisioning in hybrid fog/cloud environment. To extend existing PaaS, the authors used Cloud Foundry droplets (an open-source PaaS service), Docker containers, and REST to provide interactions between PaaS and the fog. This enabled their architecture to provide development, deployment, and management phases of hybrid cloud/fog applications. In the proposed architecture, the controller exposes an API for PaaS users to allocate resources for their applications in the deployment phase. The Cloud Foundry droplets are responsible for running the applications in the management phase.

*4.5.5 Service Modeling.* Lingen et al. [134] used the YANG modeling language to model fog services and devices, which allows for better resource orchestration and system design. In addition, modeling services and devices using YANG allows the authors to transform service intention into service instantiation in their architecture. The authors then break down the complexity of service modeling for reuse and building higher-level services for better overall service management. Through the YANG modeling language, the authors in [135] are able to effectively model IoT services description for their proposed controller to handle a large volume of data. By using the YANG modeling language, the controller can optimally allocate resources across multiple networks.

## 4.6 Design and Planning: Infrastructure Design

Network planning and design of fog networks is an important research topic, and yet not many studies are performed in this area. This is due to lack of standardized protocols and definitions for fog in general. Nevertheless, there are some early efforts, and we discuss them in this subsection. The authors of [214] propose a framework for cloudlet-based network design and planning. The goal of the study is to design a network based on time division multiplexed passive optical networks (TDM-PON) to optimize the network infrastructure cost while satisfying latency constraints. Fan et al. address the problem of placing the cloudlets to minimize the deployment cost of cloudlet providers and to minimize the end-to-end delay of user requests [210]. Since the total deployment cost of a cloudlet provider depends on the location of cloudlets and the amount of resources for cloudlets, cloudlet providers must take into account both the end-to-end delay of user requests and the deployment cost.

*4.6.1 Virtualization-based Infrastructure.* The paper [213] addresses the tradeoff between computing and communication and propose an architecture based on F-RAN. The F-RAN-based architecture consists of radio access equipment, F-RAN nodes that provide the computing resources, end devices, and an F-RAN controller that is in charge of receiving service requests and distributing tasks to fog nodes. The authors observe that this architecture can meet the demands of ultra-low latency applications by relying on front-haul wireless communications and distributing computation tasks to multiple F-RAN nodes near the end users.

Fog networks can be implemented by SDN and virtualization to reduce the management costs, and to improve the scalability and resource utilization. The authors in [130] propose an integrated network architecture for software-defined and virtualized radio access networks with fog computing. The proposed architecture is hierarchical and has two control levels: the higher level that is the SDN controller and the lower layer that is the local controller, which could be collocated with fog nodes. The SDN controller instructs the local controllers how to process specific applications or requests. The authors further introduce a SaaS called OpenPine that enables virtualization at the network level and user control of network operation.

*4.6.2 Capacity Planning.* The authors of [216] address the questions of where the edge data centers must be located and how much compute capacity needs to be allocated to each DC for cost-effectiveness while also satisfying the bandwidth and performance requirements of applications. They conclude that adding edge layer data centers results in high-cost savings for network-intensive applications while adding an intermediate data center close to the root data center is beneficial for medium to low demand compute-intensive applications. Noreikis et al. improve edge resource utilization by taking advantage of network resources such as GPUs in addition to CPUs [211]. Their study includes initial capacity planning in edge nodes to meet QoS requirements.

**Table 5: Objectives of the papers and their corresponding explanation.**

| Objective | Explanation | Example |
|---|---|---|
| QoS | The proposed scheme deals with improving the Quality of Service or quality of experience (e.g., by minimizing or controlling latency or success rate) in the fog. | Algorithms for enabling real-time applications, migration engines, task offloading, dynamic fog service provisioning. |
| Cost | The work considers cost parameters, such as operational cost or capital cost, in the proposed scheme. | Cost-aware replica placement, cost estimation or capacity planning for designing a fog network. |
| Energy | The authors analyze the energy consumption or power in the paper. | Energy-aware computation offloading, energy-aware mobility management, federation of constrained devices. |
| Bandwidth | (efficiency) The paper discusses and proposes algorithms that affect bandwidth and throughput using fog computing. | Fog resource sharing, in-network processing, edge analytics, capacity planning for designing a fog network. |
| Security | Security and privacy aspects of fog computing are considered in these studies. These include vulnerabilities, security mechanisms, privacy issues, and security protocols (e.g.authentication). | Anomaly detection using fog, location privacy, authentication schemes for fog nodes. |
| Foundation | The fundamental and foundations of fog computing are discussed in these papers. | Surveys, standards, reference architectures, reference frameworks, new concepts based on fog. |
| RAS | Reliability, Availability, Survivability. The proposed scheme improves the reliability, availability or survivability of the fog, or uses fog to provide/improve the reliability, availability or survivability, in the event of a network/node failure. | Survivable replica placement, availability analysis of fog services, availability-aware VM placement. |
| Mobility | The paper considers the mobility of IoT devices or fog nodes. | Mobility-aware fog node placement, service migration based on mobility. mobility-aware service placement. |
| Scalability | The proposed scheme can efficiently scale to the large magnitude of IoT networks. | Edge analytics, scalable IoT node management, computation offloading and task assignment (not per task). |
| Heterogeneity | The paper discusses heterogeneity or proposes frameworks that handle heterogeneity of devices. The algorithms and frameworks in the paper do not assume any particular type of node or network. | Cloud of things, computation offloading among fog nodes, federation of fog nodes or IoT nodes. |
| Management | The paper proposes management, monitoring, or federation schemes, where fog nodes, (or IoT nodes) are managed, monitored or federated using some method of management (e.g., SDN). | SDN-enabled control of fog nodes, fog operating system, orchestration of IoT services on fog nodes, orchestration of fog nodes. |
| Programmability | The proposed framework is a programming language, programming framework or data modeling for fog computing. | Fog YANG models, distributed data flow for fog, data modeling and labeling. |

## 4.7 Design and Planning: Resource Analysis and Estimation

As many publications on fog computing highlight the importance of this technology for ubiquitous access, service provisioning, and service discovery, several studies also focus on the resource pricing, estimation, and analysis of fog computing. In this subsection, we describe these studies.

*4.7.1 Fog Resource Pricing.* In fog computing, fog service providers aim to offer their services to their customers as close as possible to the customers' locations. On the other hand, infrastructure providers try to maximize their infrastructure utility by renting their edge resources to the cloud/fog service providers. An attempt to model the edge resource pricing and auction between service and content providers and edge infrastructure providers is described in [113]. Aazam and Huh in [221] formulated an estimation for pricing of fog services based on CPU, memory, storage, and bandwidth. Their pricing model also includes incentivized pricing for active customers and takes into account fog device mobility and customer history to determine a more fair price.

*4.7.2 Fog Energy Estimation.* Centralized data centers consume a significant amount of energy compared to small distributed servers, or nano data centers. Jalali et al. identify applications that are more energy-efficient when implemented on nano data centers than centralized data centers [23]. In their models, Jalali et al. use a "flow-based" energy consumption model for equipment shared by many users and services, while for network equipment close to end-users, the authors use a "time-based" energy consumption model based on the amount of time the equipment needs to provide access to services. The authors conclude the best energy savings with nano data centers is in applications that generate and distribute a large quantity of data near end-users that is not frequently accessed (e.g., local video surveillance in homes).

Similar to the above study, the study in [222] focuses on the energy consumption of IoT applications using both fog and cloud. To increase the power efficiency of IoT applications, Jalali et al.

propose the use of both fog and microgrids [222]. In their paper, the authors find that if a centralized grid provides power, fog computing is more energy efficient if there is little computation to be done with the data. However, cloud computing is more efficient if high data processing is required.

*4.7.3 Fog Resource Estimation.* The allocated resources to fog nodes must be elastic, scalable, and dynamic, because heterogeneous IoT nodes forming IoT networks are highly dynamic, both spatially and temporally. To address this challenge, the authors in [223] propose an analytical model that analyzes the required fog resources for some offered IoT workload and scales fog nodes according to incoming IoT workloads. Dynamic scaling of fog nodes is scaling up or down allocated resources according to the incoming IoT workload. Ideally, when the workload is high, more fog nodes (or more resources on the fog nodes) are provisioned, whereas when the workload is low, it may be possible to free up some fog nodes or release the allocated fog resources.

Similarly, the authors in [224] take a close look at server utilization. Specifically, the paper studies the relationship between the extent to which MEC is deployed and the resulting average server utilization and latency. The metric for latency is defined as the distance between the base stations and the servers processing the traffic, and the efficiency metric for server deployment and utilization is defined as the ratio between the average and peak traffic processed by the servers.

*4.7.4 Estimating Load and Response Time.* Fog computing can reduce the web response times of modern websites by enabling the generation of dynamic web contents at edge nodes close to end users. In [225], the authors proposed a simple formula to estimate the lower bound of this reduction of response time. Moreover, they measured the response times of about 1000 popular web pages from 12 locations across the world to evaluate whether edge computing is suitable for offering dynamic web content. Based on their experiments, edge computing can reduce web responses across the world except in North America and Europe where the round-trip times are considerably short. In [226], a location-aware load prediction for edge data centers is proposed. For each edge data center, the load is predicted using its historical load time series and those of its neighboring data centers.

## 4.8 Resource Management and Provisioning: Service Provisioning (Orchestration and Migration)

Due to the limited storage capacity of fog nodes, proper resource utilization of fog nodes has a significant impact on their performance.

*4.8.1 Service Provisioning.* The study in [248] focuses on dynamic service provisioning in edge clouds from a theoretical perspective. The model in the study captures the limited capacity of fog nodes, the unknown arrival process of requests, the cost of forwarding requests to the remote cloud, and the cost of onloading a new service on a fog node. In a simulation study, the authors of [126] suggest a conceptual framework for fog resource provisioning. They introduce the concept of “fog cell,” which is a software component running on fog devices that controls and monitors a particular group of IoT devices. Using this and other related concepts, they model orchestration of IoT devices using a hierarchical cloud/fog resource control and provide a suitable resource provisioning solution for distributing tasks among them. Recently, researchers in [169] present the *Red Wedding Problem*, a real-world scenario motivating the need for stateful computations at the edge. They implement a prototype database for operation at the edge that addresses the issues presented in the Red Wedding Problem.

*4.8.2 Service Migration.* The authors of [232] study the service migration problem in edge clouds, in response to user movement and network performance. The solution is based on based on Markov Decision Process (MDP) that considers network state and server response time in making migration decisions. In [249], it is suggested to use multi-path TCP for live migration of VMs across edge nodes to improve VM migration time and network transparency of applications.

Farris et al. define two integer linear programming (ILP) optimization schemes to minimize QoE degradation and cost of replica deployment in service replication for MEC [234]. They distinguish classic reactive service migration from proactive migration: reactive service migration is dependent on user movement and accommodates this movement by locating the most suitable target edge and then starting the process for migration; however, proactive service migration deploys multiple replicas of the user service to neighboring nodes. As a future direction, one could study path-oriented proactive migration and trade-offs between the probability of reactive migration and the cost of service replications.

*4.8.3 Orchestration Frameworks.* To align deployed applications in distributed systems, Wen et al. underscore the importance of fog orchestration [24]. The authors develop methodologies for studying fog orchestration systems with a focus on challenges in reliability, scalability, and security for fog orchestration. Another effort for service orchestration is the work of the authors in [233] called “Foggy.” Foggy is a framework for continuous and automated application deployment in the fog. It facilitates dynamic resource provisioning and automated IoT application deployment in fog architectures. Foggy assumes that IoT nodes can host Docker containers. Developers push their containerized application package and their requirement specification to an orchestration center, which is a central authority that is in charge of monitoring each IoT node’s resources and deploying services on IoT nodes.

The authors in [239] propose a service-oriented middleware that aims to distribute services over fog nodes for scalability, and with the help of SDN, performs QoS-aware orchestration by scheduling flows between services. The architecture proposed mainly consists of two components - the service-oriented middleware and the distributed service orchestration engine. The service-oriented middleware abstracts device functionalities, allowing all the nodes to act as service hosting platforms, whereas the distributed service orchestration performs orchestration. Similarly, the authors in [132] proposed a Platform as a service (PaaS) architecture for automating application provisioning and orchestration in hybrid fog/cloud environments.

*4.8.4 Virtualization Technologies for Fog Computing.* The study in [236] explores container-oriented operational frameworks for

IoT from the perspective of how lightweight virtualization can help exploit the resources offered by IoT devices. The authors propose to use Docker-based service provisioning in wireless resource-constrained IoT environments. To analyze and identify the impact of Docker management, the authors consider two use cases, based on which they propose two distinct approaches to container-based IoT service provisioning.

In their proposed framework called FADES, researchers from TUM advertise the use of unikernels (single-purpose standalone kernels) to isolate and embed application logic into bootable images [245]. FADES is a modular orchestration architecture for handling heavy computation from IoT devices to edge nodes. Each FADES unit supervises a subset of IoT devices. When needed, FADES pulls the necessary application/service from the cloud to handle the requests of IoT devices. Similar to MCC, IoT devices resort to computational resources of edge nodes to handle heavy computations. In this sense, FADES is an orchestration framework for pulling required services for IoT devices from the cloud.

*4.8.5 Provisioning of Resource-limited IoT Devices.* To satisfy QoS requirements in an edge cloud built by single-board devices (e.g., Raspberry Pi), the University of Cambridge researchers in [240] present a platform, PiCasso[4], for service deployment based on the service specifications and the status of the resources at the hosting devices. The architecture of PiCasso consists of edge nodes and a service orchestrator. Similar to PiCasso, an edge node resource management framework with provisioning and deployment capability is introduced in [242]. The framework integrates a fog node with a cloud server and supports auto-scaling to manage edge resources dynamically. The framework is developed for resource-limited environments and hence is shown to be simple. The authors then validate the proposed framework on a location-aware and latency-sensitive online game use case (PokeMon Go).

*4.8.6 Handover.* Handover issues of mobile IoT devices between access points should be considered in designing orchestration or migration frameworks. In a recent study, it is shown that VM handoff in the edge across cloudlets is more than an order of magnitude faster than live migration methods currently used in data centers, for typical WAN bandwidths [250]. The authors in [243] observe that traditional mobile network handover mechanisms cannot handle the demands of fog computation resources and the low-latency requirements of mobile IoT applications. The authors propose *Follow Me Fog* framework to guarantee service continuity and reduce latency during handovers. The key idea proposed is to continuously monitor the received signal strength of the fog nodes at the mobile IoT device, and to trigger pre-migration of computation jobs before disconnecting the IoT device from the existing fog node. Comparably, the authors in [244] use Docker container migration between edge nodes for service handoff.

## 4.9 Resource Management and Provisioning: Placement (VM and Service)

In this subsection, we survey research articles that address placement problems (e.g., service placement, VM placement, content placement, and caching) in fog networks.

[4]Available at https://github.com/AdL1398/PiCasso

*4.9.1 Application (Service) Placement.* Gu et al. developed a linear programming-based two-phase heuristic algorithm to compute an optimal solution for service placement in fog computing medical cyber-physical systems [255]. The authors formulate service placement problem based on user association, task distribution, VM placement, and QoS constraints. Souza et al. also formulate service allocation in a combined fog-cloud ecosystem in [256]. The authors in [274] address the problem of multi-component application placement on fog nodes. Each application could be modeled as a graph, where each node is a component of the application, and the edges indicate the communication between them.

The researchers in [258] investigate the problem of optimal resource provisioning and fog service placement while taking into account their QoS requirements. The authors define the fog service placement problem and formulate it as an integer linear programming problem with the objective of maximizing the utilization of the fog landscape. As a research direction, we motivate the reader to consider other constraints such as the availability of resources, the reliability of services, and the cost of resources. Similarly, an uncoordinated strategy for service placement in edge-clouds is studied in [273].

Virtual network functions (VNFs) are network services that provide some network functionality and provide flexible ways to deploy network services [438]. The study in [257] focuses on the QoS-aware VNF placement and provisioning problem over an edge-cloud infrastructure. They propose a strategy to determine the required resources and placement of VNFs in the two-tier carrier cloud infrastructure while considering SLA requirements. They formulate the VNF placement problem as Mixed Integer Linear Program (MILP).

The paper [260] addresses the problem of allocating computing resources (specifically, containers) in edge networks. The authors introduce a contract model between cloud service providers and edge infrastructure providers for resource sharing. Based on the contract, service providers can provision their services on the micro data centers owned by edge infrastructure provides. This essentially decouples the management of the edge infrastructures with that of the service placement performed by service providers.

*4.9.2 VM Placement.* The paper [261] analyzes the VM placement problem in fog radio access networks (F-RANs) with the objective to minimize the overall back-haul traffic. The back-haul traffic is incurred in two ways: the VM replication and data transmission to the cloud. When a user connects to a fog node and requests an application service, there is no back-haul bandwidth consumption if the fog node has the application VM. Otherwise, the VM has to be replicated on the fog node, or the request is forwarded to the cloud. They formulate the replica-based VM placement problem by considering the computing and storage of fog nodes, the user service constraint, as well as the edge bandwidth constraint. Similarly, the study [265] addresses the issues of launching VM replicas and migrating them for latency-sensitive, computation and memory intensive applications in a MEC environment.

Guaranteeing availability in a fog network needs a careful design, as fog nodes are presumed to be less reliable than always-provisioned cloud data centers. The authors in [268] study VM placement in MEC with respect to availability. The goal is to find placement strategies for different types of multi-access edge applications with low cost while satisfying the availability requirements of an application. Comparably, the traffic-aware VM placement problem in the cloudlet mesh with the objective of minimizing the total inter-cloudlet communication traffic is studied in [272]. The study in [264] proposes a messaging method with low overhead that notifies fog nodes about nearby replica nodes so that the replica nodes can be used for handling requests instead of depending on cloud storage.

*4.9.3 Caching.* Analogous to service placement, caching data on fog nodes can also considerably reduces the data retrieval delay compared to solely relying on a central data store. In [267], the authors propose a cache grouping mechanism for managing content in edge clouds. They also present a cache coherency mechanism to ensure data consistency within a cache group. Another study on cache placement is discussed in [149], where cache placement is done via the MEC paradigm in wireless networks to satisfy the demands of automated driving services. The service requests submitted by autonomous vehicles are first processed by the edge server to determine if they can be processed locally or need to be handed to the remote cloud.

In their recent study, the authors of [269] study the cache placement problem in F-RANs and consider flexible physical-layer transmission schemes. They develop centralized and distributed cache placement strategies to minimize users' average download delay while meeting the fog storage capacity constraints. Similar to cache placement, the cloudlet placement problem is of prominent importance. [270] introduces the concept of movable cloudlets and explores the problem of how to cost-effectively deploy these movable cloudlets to enhance cloud services for dynamic context-aware mobile applications. To this end, the authors propose an adaptive cloudlet placement method for mobile applications. Another use of edge caching may be for 360-degree vedeo streaming. Streaming high-quality 360-degree video is challenging due to high bandwidth and low latency requirements [439]. In [440] a probabilistic model of common field of view (FoV) for 360-degree video based on previous users' viewing histories to improve caching performance at the edge is introduced.

## 4.10 Resource Management and Provisioning: Control and Monitoring

*4.10.1 Control and Monitoring of Fog-based Networks.* The traditional network architecture was not designed with high levels of scalability of IoT devices in mind. Tomovic et al. [127] and Xu et al. [281] propose a control and monitoring framework for the fog-based IoT network that utilizes SDN. Using SDN allows a system to know the requirements of the network and all of the resources it has, giving it the ability to handle large waves of data [443]. In the proposed model, a global view of the network allows for easy fog orchestration. The SDN controller controls fog orchestration as it can see all the available fog nodes and their resources such as RAM, storage, and software applications. The authors of [281] use the message queuing telemetry transport (MQTT) protocol in addition to SDN to enable effective and reliable delivery of IoT data. The SDN controller is a fog node that is the broker for MQTT clients.

The study [137] explores the idea of implementing a wide area SDN controller for fog and cloud while also taking into consideration the problem of minimizing the overall carbon footprint of data centers. The paper [280] proposed an SDN-enabled wireless fog architecture by combining Openflow and distributed wireless protocols. The architecture deploys a hybrid SDN control plane to address the limitations of both the centralized and distributed SDN controllers. In case of a controller failure or overhead, additional controllers are added at runtime as required to balance the network performance. Likewise, the study [9] motivates how edge computing can benefit from SDN, proposes a collaboration model between SDN and edge computing through practical architectures, and shows that SDN can feasibly operate within the edge computing infrastructures.

*4.10.2 Virtualization for Control and Monitoring.* Some studies suggest bringing virtualization of network services to the edge devices. The authors in [238] argue that the heavy footprint of today's virtualization platforms is responsible for preventing them from being utilized at the network edge. They present Glasgow Network Functions (GNF), a container-based NFV platform that runs and orchestrates lightweight VNFs. They show that the presented framework has low VNF instantiation time and memory requirements as compared with other existing virtualization technologies, making it suited to run on a variety of edge devices. On the other hand, the TelcoFog controller [135] extends the functionalities of an NFV orchestrator for dynamic deployment of virtualized functions. The main component in the TelcoFog controller is the Resource Orchestration, which defines and enforces orchestration logic based on fog node status provided by the Resource monitoring module.

*4.10.3 Control and Monitoring Other Networks via Fog.* Vehicular ad hoc networks (VANETs) face many issues such as unreliable connectivity, delay constraints, and poor scalability. The authors of [279] suggest that using principles of fog computing along with SDN could solve many of the current problems with VANETs. In their architecture, the vehicles, which act as end-users, are SDN wireless nodes. These wireless nodes send their data to Road Side Units (RSUs) which are installed alongside road systems. Once the data is sent to an RSU, it is then sent to an RSU controller (RSUC) which is a cluster of RSUs connected by broadband. The RSUC is capable of data storage and processing. Finally, the RSUCs communicate with the SDN controller, which has global knowledge of the VANET system.

## 4.11 Operation: Scheduling, offloading, and Load Balancing

In this subsection, we survey the studies that address job scheduling, job dispatching, task offloading, and load balancing in fog networks.

*4.11.1 Offline vs. Online.* It is worth mentioning that we discuss several candidate papers among all of the papers in this area since there are a large number of papers in this area and most of them study "centralized" (referred to as *offline*) scheduling, dispatching, offloading, and load balancing. In centralized settings, either full information about the tasks, network, or nodes is known; or a centralized entity decides where each task could be sent. Even though there is a large body of research in this area, a more challenging

## Table 6: Features and objectives of the cited papers in this survey

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Deng et al. [124] | ✓ | | ✓ | | | | | | ✓ | ✓ | | |
| Zhao et al. [288] | ✓ | | | ✓ | | | | | | | | |
| Vaquero and Rodero-Merino [71] | | | | | | ✓ | | | | | ✓ | |
| Dastjerdi and Buyya [441] | | | | | | ✓ | | | | | | |
| Streiffer et al. [397] | ✓ | | ✓ | | | | | | ✓ | | | |
| Garcia Lopez et al. [43] | | | | | | ✓ | | | | | | |
| Aazam and Huh [221] | | ✓ | | | | | | | | | ✓ | |
| Gu et al. [255] | ✓ | ✓ | | | | | | | | | ✓ | |
| Urgaonkar et al. [231] | ✓ | ✓ | | | | | | ✓ | | | | |
| Guo et al. [289] | ✓ | ✓ | | | | | | | | | | |
| Souza et al. [256] | ✓ | | | | | | | | | ✓ | | |
| Stojmenovic and Wen [82] | | | ✓ | | | ✓ | | | | | | |
| Yannuzzi et al. [97] | ✓ | | | | | ✓ | | ✓ | ✓ | | | |
| Truong et al. [279] | | | | | | | | ✓ | | | ✓ | |
| Gao et al. [83] | | | | | | | | | | ✓ | ✓ | |
| Sarkar and Misra [72] | ✓ | | ✓ | | | | | | | ✓ | | |
| Li et al. [290] | ✓ | ✓ | | ✓ | | | | | | | | |
| Sarkar et al. [89] | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | | |
| Jemaa et al. [257] | ✓ | ✓ | | | | | | | | | ✓ | |
| Alippi et al. [125] | | | | | | ✓ | | | | | | |
| Skarlat et al. [126] | ✓ | ✓ | | | | ✓ | | | ✓ | ✓ | ✓ | |
| Sun and Ansari [106] | | | | | | ✓ | | | | ✓ | ✓ | |
| Tomovic et al. [127] | | | | | | | | | ✓ | ✓ | ✓ | |
| Skarlat et al. [258] | ✓ | ✓ | | | | | | | | ✓ | ✓ | |
| Shi et al. [80] | | | | | | ✓ | | | | | | |
| Chiang and Zhang [84] | | | | | | ✓ | | | | | | |
| Villari et al. [67] | | | | | | ✓ | | | | | | |
| Brogi and Forti [107] | ✓ | | | | | | | | | ✓ | | ✓ |
| OpenFogConsortium [6] | | | | | | ✓ | | | | | | |
| Zhang et al. [194] | | | | | | | | | | | | ✓ |
| Li et al. [330] | | | | ✓ | | | | | | | | |
| Dantu et al. [195] | | | | | | | ✓ | | | | | |
| Ku et al. [128] | ✓ | | | | | | | | | ✓ | | |
| Gupta et al. [380] | ✓ | | ✓ | | | | | | ✓ | ✓ | | ✓ |
| Dsouza et al. [108] | | | | | ✓ | | | | | | ✓ | |
| Yi et al. [109] | | | | | | ✓ | | | | | | |
| Hong et al. [259] | ✓ | | | | | | | | | ✓ | ✓ | |
| Hao et al. [98] | ✓ | | | | | ✓ | | | | | | ✓ |
| Cirani et al. [171] | | | | | | | | | ✓ | ✓ | ✓ | |
| Malensek et al. [331] | | | | ✓ | | | | | ✓ | | | |
| Moreno-Vozmediano et al. [129] | | | | ✓ | | | | | ✓ | ✓ | ✓ | |
| Sill [73] | | | | | | ✓ | | | | | | |
| Bittencourt et al. [291] | ✓ | | | | | | | ✓ | | | | |
| Esposito et al. [419] | | | | | ✓ | ✓ | | | | | | |
| Liang et al. [130] | ✓ | | | ✓ | | | | ✓ | ✓ | | ✓ | |
| Kapsalis et al. [131] | ✓ | | | | | | | | | | | |
| Yu et al. [420] | | | | | ✓ | | | | | | | |
| Kattepur et al. [292] | ✓ | | | | | | | | | ✓ | | |
| Kattepur et al. [293] | ✓ | | ✓ | | | | | | | ✓ | | |
| Basudan et al. [421] | | | | | ✓ | | | | | | | |
| Yangui et al. [132] | | | | | | | | | ✓ | ✓ | ✓ | |
| Xu et al. [260] | ✓ | ✓ | | | | | | | | | ✓ | |
| Kaur et al. [196] | ✓ | | ✓ | | | | | | | | ✓ | |
| Saurez et al. [197] | ✓ | | | | | | | ✓ | | | ✓ | ✓ |
| Nan et al. [294] | ✓ | ✓ | ✓ | | | | | | | | | |
| Xiao and Krunz [295] | ✓ | | ✓ | | | | | | | | | |
| Byers [96] | | | | | | ✓ | | | | | | |
| Yang [332] | | | ✓ | | | ✓ | | | | | | |
| Zhang et al. [232] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | |
| Klas [78] | | | | | | ✓ | | | | | | |
| Mouradian et al. [8] | | | | | | ✓ | | | | | | |
| Wen et al. [24] | ✓ | | | | | | | | | | ✓ | |
| Jayaraman et al. [333] | | ✓ | ✓ | | | | | | ✓ | | ✓ | |
| Zao et al. [334] | | | | | | | | | | | | ✓ |
| Li et al. [133] | | | | ✓ | | | | | | | | |
| Hou et al. [172] | | | | | | ✓ | | ✓ | | | ✓ | |
| Jalali et al. [23] | | | ✓ | ✓ | | | | | | | | |
| Jalali et al. [222] | | | ✓ | | | | | | | | | |
| Peng et al. [27] | | | ✓ | | | | | ✓ | | | | |
| Bruneo et al. [381] | | | | | | | | ✓ | | | | |
| Prazeres and Serrano [173] | | | | | | | | | ✓ | ✓ | ✓ | |
| Jain and Tata [198] | | | | | | | | | | | | ✓ |
| Xu et al. [382] | | | | | | | | | ✓ | | ✓ | |
| Yu et al. [261] | | | | ✓ | | | | ✓ | | | | |
| Lee et al. [296] | ✓ | | | | | | | | | ✓ | | |
| Chen and Zhang [297] | ✓ | | ✓ | | | | | | | | | |
| Hakiri et al. [280] | | | | ✓ | | | | | | | ✓ | |
| Baktir et al. [9] | | | | | | ✓ | | | | | ✓ | |
| Xu et al. [281] | | | | | | | | | | | ✓ | |
| Chang et al. [95] | | | | | | | | | | ✓ | ✓ | |
| Liu et al. [383] | ✓ | | | ✓ | | | | | ✓ | | ✓ | ✓ |
| Yigitoglu et al. [233] | | | | | | | | | | ✓ | ✓ | ✓ |
| Hong et al. [199] | ✓ | | | | | | | | ✓ | | | ✓ |
| Yin et al. [298] | | ✓ | | ✓ | | | | | ✓ | | ✓ | |
| Ananthanarayanan et al. [335] | ✓ | | | | | | | | ✓ | | | |
| Ni et al. [13] | | | | | ✓ | ✓ | | | | | | |
| Cheng et al. [336] | | | | | | ✓ | | | | | | |
| Fan and Ansari [210] | ✓ | ✓ | | | | | | ✓ | | | | |
| Wang and Zhou [299] | ✓ | | | | | | | | | | | |
| Sun et al. [300] | | ✓ | ✓ | | | | | ✓ | | | | |
| Ti and Le [301] | ✓ | | ✓ | ✓ | | | | | | | | |
| Farris et al. [234] | | | | | | | | ✓ | | | | |
| Zhang et al. [302] | ✓ | ✓ | | | | | | | | | | |
| Xiao et al. [211] | ✓ | ✓ | | ✓ | | | | | | | | |
| Abdelwahab et al. [61] | | | | | | ✓ | | | ✓ | ✓ | | |
| Abdelwahab and Hamdaoui [235] | ✓ | | ✓ | ✓ | | | | | ✓ | ✓ | | |
| van Lingen et al. [134] | | | | | | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| Vilalta et al. [135] | ✓ | | | | | ✓ | | | | ✓ | ✓ | ✓ |
| Choi et al. [200] | | | | | | | | | ✓ | ✓ | ✓ | |
| Zhang et al. [136] | | | | | | ✓ | | | | | | |
| Kaminski et al. [174] | ✓ | ✓ | | | | | | | | ✓ | | |
| Farris et al. [175] | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | | |
| Borylo et al. [137] | ✓ | | ✓ | | | | | | | ✓ | ✓ | |
| Morabito et al. [236] | | | | | | | | | | ✓ | ✓ | |
| Amento et al. [138] | | | ✓ | | | | | | ✓ | ✓ | ✓ | |
| Ottenwälder et al. [237] | ✓ | ✓ | | | | | | ✓ | | | | |
| Giang et al. [201] | | | | | | | | ✓ | ✓ | ✓ | | ✓ |
| Giang et al. [238] | ✓ | | | | ✓ | | | ✓ | | | ✓ | |
| Zeng et al. [262] | ✓ | | | | | | ✓ | | | | | |
| Gupta et al. [239] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |
| Yannuzzi et al. [139] | | | | | | ✓ | | | ✓ | ✓ | ✓ | |
| Elkhatib et al. [398] | ✓ | | | | | | | | ✓ | | | |
| Alrawais et al. [422] | | | | | ✓ | | | | | | | |
| Muñoz et al. [399] | | | | | | | | | | | ✓ | |
| Lertsinsrubtavee et al. [240] | ✓ | | | | | | | | | | ✓ | ✓ |
| Hong et al. [263] | ✓ | | | ✓ | | | | | | ✓ | ✓ | |
| Rodrigues et al. [241] | ✓ | | | ✓ | | | | | | | | |
| Tong et al. [92] | ✓ | | | | | ✓ | | | ✓ | | ✓ | |
| Aral and Ovatman [264] | ✓ | ✓ | | ✓ | | | | | ✓ | ✓ | | |
| Tan et al. [303] | ✓ | | | | | | | | | ✓ | | |
| Sookhak et al. [176] | | | | | | ✓ | | | ✓ | ✓ | | |
| El Kafhali and Salah [223] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hu et al. [337] | | | | | ✓ | | | | | ✓ | | |
| Lyu et al. [338] | ✓ | | ✓ | | | | | | ✓ | | | |
| Huo et al. [339] | | | | | ✓ | | | | | ✓ | | |
| Al Faruque and Vatanparvar [384] | | | ✓ | | | | | | ✓ | ✓ | | |
| Li et al. [282] | | | | | ✓ | | ✓ | | | | ✓ | |
| Wang et al. [242] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |
| Tao et al. [177] | ✓ | | ✓ | | | | | ✓ | ✓ | | | |
| Nan et al. [140] | ✓ | ✓ | | | | | | | | ✓ | | |
| Azimi et al. [340] | ✓ | | | | | | ✓ | | ✓ | | ✓ | |
| He et al. [304] | ✓ | ✓ | ✓ | | ✓ | | | | | | | |
| Zhao et al. [265] | | ✓ | | ✓ | | | | | ✓ | | ✓ | |
| Chang et al. [305] | ✓ | | ✓ | | | | | | | | | |
| Tinini et al. [266] | | ✓ | ✓ | ✓ | | | | | | | ✓ | |
| Shen et al. [341] | | ✓ | | | ✓ | | | | ✓ | | | |
| Iotti et al. [400] | | | | ✓ | | | | | ✓ | ✓ | | |
| Yi et al. [99] | | | | | ✓ | | | | | | | |
| Hu et al. [342] | ✓ | | | | | | | | ✓ | ✓ | | |
| Wang et al. [141] | | | | | | ✓ | | | | | | |
| Bao et al. [243] | ✓ | | | | | | | ✓ | | ✓ | | |
| Yi et al. [306] | ✓ | | | ✓ | | | | | | ✓ | | |
| Ma et al. [244] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | |
| Ficco et al. [142] | | | | | | | ✓ | | ✓ | ✓ | | |
| Huang et al. [178] | | | | | ✓ | ✓ | | | | | | |
| Ahmed et al. [79] | | | | | | ✓ | | | | | | |
| Markakis et al. [93] | ✓ | | | | | ✓ | | | ✓ | ✓ | ✓ | |
| Markakis et al. [94] | | | ✓ | | | ✓ | | | ✓ | ✓ | ✓ | |
| Li et al. [143] | | | | | | ✓ | | ✓ | ✓ | ✓ | | |
| Yu et al. [144] | | | | ✓ | | | | | | ✓ | | |
| Basudan et al. [423] | | ✓ | | | ✓ | | | | ✓ | | | |
| Nikoloudakis et al. [343] | ✓ | | | | | | | | ✓ | | | |
| Lu et al. [424] | | ✓ | | ✓ | ✓ | | | | | ✓ | | |
| Wang et al. [425] | | | | | ✓ | | | | | ✓ | | |
| Guo et al. [212] | ✓ | | ✓ | | | | | | | | | |
| Masip-Bruin et al. [91] | ✓ | | | | | ✓ | | | ✓ | ✓ | ✓ | |
| An et al. [426] | | | | | ✓ | | | | | ✓ | | |
| Guevara et al. [74] | | | | | | ✓ | | | | | | |
| Desikan et al. [307] | ✓ | | | | | | | | ✓ | ✓ | | |
| Shih et al. [213] | ✓ | ✓ | | ✓ | | ✓ | | | ✓ | | | |
| Zhang et al. [145] | ✓ | | ✓ | | | ✓ | | ✓ | ✓ | | ✓ | |
| Ren et al. [179] | ✓ | | ✓ | | | | | | ✓ | ✓ | | |
| Zhang et al. [146] | | | | | | ✓ | | ✓ | ✓ | | | |
| Lorenzo et al. [147] | ✓ | | ✓ | | | | ✓ | | ✓ | ✓ | | |
| Chen et al. [434] | | | | | | | | | | ✓ | | |
| Tang and He [308] | ✓ | | ✓ | | | | | | | | | |
| Grewe et al. [100] | ✓ | | | | | ✓ | | | | | | |
| Mohan et al. [267] | ✓ | | | | | | | | ✓ | ✓ | | |
| Cozzolino et al. [245] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |
| Giri et al. [427] | | | | | ✓ | | | | | | ✓ | |
| Mondal et al. [214] | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | | |
| Zhu and Huang [268] | | ✓ | | ✓ | | | ✓ | | ✓ | | | |
| Malandrino et al. [224] | ✓ | | | | | ✓ | | | | | | |
| Lyu et al. [90] | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | | |
| Suganuma et al. [148] | | | | | | | | | ✓ | | ✓ | |
| Yuan et al. [149] | | | | | | | | ✓ | ✓ | ✓ | ✓ | |
| Cho et al. [322] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | |
| Liu et al. [269] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |
| Kiani and Ansari [110] | ✓ | ✓ | ✓ | ✓ | | | | | | | ✓ | |
| Sun and Ansari [246] | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | | ✓ | |
| Xiang et al. [270] | | | | | | | | ✓ | ✓ | | | |
| Kamiyama et al. [225] | ✓ | | | | | ✓ | | | | | | |
| Perera et al. [101] | | | | | | ✓ | | | | | | |
| Pahl et al. [401] | | | | | | | | | ✓ | | ✓ | |
| Lin and Shen [323] | ✓ | ✓ | | ✓ | | | | | ✓ | ✓ | | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Schäfer et al. [324] | | | | | | | | | ✓ | ✓ | | |
| Teerapittayanon et al. [344] | | | | ✓ | | | | | ✓ | | | |
| Li et al. [111] | ✓ | | ✓ | | | ✓ | | | | ✓ | | |
| Abdallah et al. [402] | | | | | | | | | | ✓ | ✓ | ✓ |
| Hoque et al. [403] | ✓ | | | | | | | | | | | |
| He et al. [428] | | | | | ✓ | | | | | | | |
| Drolia et al. [271] | ✓ | | | ✓ | | | | | ✓ | ✓ | | |
| Gedeon et al. [325] | ✓ | | | | | | | ✓ | | ✓ | | |
| Yousefpour et al. [88] | ✓ | | | | | | | | ✓ | ✓ | | |
| Vallati et al. [215] | ✓ | | ✓ | | | | | | | | | |
| Farris et al. [112] | ✓ | | | | | | | ✓ | ✓ | ✓ | | |
| Kamath et al. [345] | | ✓ | | | | | | | ✓ | | | |
| Lebre et al. [150] | ✓ | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| Bellavista et al. [87] | | | | | | ✓ | | | | ✓ | | |
| Lopes et al. [385] | ✓ | | | | | | | ✓ | | ✓ | | ✓ |
| Ismail et al. [404] | | | | | | | | | | ✓ | ✓ | ✓ |
| Confais et al. [386] | ✓ | | | | | | | ✓ | ✓ | | | |
| Hong et al. [346] | ✓ | | | | | | | | ✓ | ✓ | | |
| Santoro et al. [405] | | | | ✓ | | | | | | | ✓ | |
| Li and Nabrzyski [272] | | | | ✓ | | | | | ✓ | | ✓ | |
| Khan and Freitag [387] | | | | | | | | | | | | ✓ |
| Ascigil et al. [273] | ✓ | | | | | | | | ✓ | ✓ | | |
| Abderrahim et al. [151] | | | | | | ✓ | | | ✓ | ✓ | ✓ | |
| Mehta et al. [216] | | ✓ | | ✓ | | | | | ✓ | ✓ | | |
| Garcia-Perez and Merino [152] | ✓ | | ✓ | | | ✓ | | ✓ | | ✓ | ✓ | |
| Dhakal and Ramakrishnan [347] | | | | | | | | | ✓ | | | |
| Le Tan et al. [226] | | | | ✓ | | | | ✓ | | ✓ | | |
| Sonmez et al. [388] | | | | | | | | ✓ | | ✓ | | ✓ |
| Alonso-Monsalve et al. [180] | | | | ✓ | | ✓ | ✓ | | ✓ | ✓ | | |
| Beraldi et al. [309] | ✓ | | | | | | | | | ✓ | | |
| Le et al. [153] | ✓ | | ✓ | | | | ✓ | | | ✓ | | |
| Baresi et al. [406] | ✓ | | | | | | | | ✓ | | | |
| Lujic et al. [154] | | | | | | | | | ✓ | ✓ | ✓ | |
| Li et al. [181] | ✓ | | | ✓ | | | | | | ✓ | | |
| Mangiante et al. [348] | ✓ | | | ✓ | | | | | ✓ | | | |
| Tärneberg et al. [155] | ✓ | ✓ | | | | | | | ✓ | ✓ | ✓ | |
| Shekhar et al. [156] | ✓ | ✓ | | | | | | | | ✓ | ✓ | |
| Jonathan et al. [310] | ✓ | | | | | | ✓ | | ✓ | ✓ | | |
| Gosain et al. [407] | ✓ | | | | | ✓ | | | ✓ | ✓ | | ✓ |
| Bhardwaj et al. [66] | ✓ | | | ✓ | ✓ | | | | ✓ | ✓ | ✓ | |
| Renart et al. [202] | ✓ | | | | | | | | ✓ | ✓ | | ✓ |
| Echeverría et al. [389] | | | | | ✓ | | | | | | ✓ | ✓ |
| Ni et al. [349] | | | | ✓ | ✓ | | | | ✓ | ✓ | | |
| Helmer et al. [408] | | ✓ | ✓ | ✓ | | | ✓ | | ✓ | | | |
| Morshed et al. [68] | | | | | | ✓ | | | | ✓ | | |
| Prasad et al. [113] | | ✓ | | | | | | | | ✓ | | |
| Li et al. [350] | ✓ | | ✓ | | | | | | ✓ | | | |
| Yoon et al. [351] | ✓ | | | ✓ | | | | | ✓ | ✓ | | |
| Zhang et al. [247] | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | | ✓ | |
| Manzoni et al. [75] | ✓ | | | | | ✓ | | | ✓ | ✓ | | |
| Bhardwaj et al. [352] | ✓ | ✓ | | ✓ | | | | | ✓ | | | |
| Jang et al. [353] | ✓ | | ✓ | | | | | | ✓ | | | |
| Naas et al. [157] | ✓ | | | | | | | | | ✓ | ✓ | |
| Qi et al. [354] | | | | | | | | ✓ | ✓ | | | |
| Sajjad et al. [355] | ✓ | | | ✓ | | | | | ✓ | | | ✓ |
| Jonathan et al. [76] | | | | ✓ | | | ✓ | | | ✓ | | |
| Lee et al. [356] | ✓ | | | ✓ | | | | | ✓ | | | |
| Singh et al. [311] | ✓ | ✓ | | | ✓ | | | | ✓ | ✓ | | |
| Mei et al. [357] | | | | | | | | | | ✓ | | |
| Fricker et al. [312] | ✓ | | | | | | | | | ✓ | | |
| Varshney and Simmhan [85] | | | | | | ✓ | | | | | | |
| Hao et al. [51] | ✓ | | ✓ | ✓ | | | | | ✓ | | | |
| Chen and Xu [313] | ✓ | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Weinman [77] | | | | | | ✓ | | | | | | |
| Hou et al. [248] | | ✓ | | | | | | | | ✓ | ✓ | |
| Yang et al. [158] | ✓ | | | | | ✓ | | | | ✓ | ✓ | |
| Habak et al. [64] | ✓ | | | | | | | | ✓ | ✓ | ✓ | |
| Bahreini and Grosu [274] | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | |
| Ryden et al. [182] | ✓ | | | ✓ | | | ✓ | | ✓ | ✓ | | |
| Hadžić et al. [409] | ✓ | | | | | | | | | | | |
| Hagenauer et al. [183] | ✓ | | | | | ✓ | | ✓ | ✓ | ✓ | | |
| Ni et al. [102] | | | | | ✓ | ✓ | | | | | | |
| Drolia et al. [358] | ✓ | | ✓ | | | | | | ✓ | | | |
| Gia et al. [359] | ✓ | | | ✓ | | | | | ✓ | ✓ | | |
| Marín-Tordera et al. [70] | | | | | | ✓ | | | | ✓ | | |
| Chaufournier et al. [249] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | |
| Mortazavi et al. [184] | ✓ | | | | | | ✓ | | ✓ | ✓ | | ✓ |
| Chen et al. [410] | ✓ | | | ✓ | | | | | | | | |
| Chen et al. [360] | ✓ | | | | | | | | | | | |
| Tang et al. [159] | ✓ | | | | | ✓ | | | ✓ | ✓ | | |
| Ha et al. [250] | ✓ | | | ✓ | | | | | ✓ | | ✓ | |
| Sha et al. [429] | | | | | ✓ | | | | | ✓ | ✓ | |
| Kar et al. [361] | ✓ | | | | | | | | ✓ | ✓ | | |
| Yu et al. [314] | | | | ✓ | | | | | ✓ | ✓ | ✓ | |
| Mahmud et al. [103] | | | | | | ✓ | | | | | | |
| Hu et al. [86] | | | | | | ✓ | | | | | | |
| Kimovski et al. [442] | ✓ | | | ✓ | | ✓ | | | | | | |
| Naas et al. [390] | ✓ | | | | | | | | ✓ | | ✓ | ✓ |
| Lujic et al. [362] | ✓ | | | | | | | | ✓ | | | |
| Su et al. [203] | ✓ | | | | | | | | ✓ | ✓ | | ✓ |
| Buzachis et al. [411] | ✓ | | | | | | | | ✓ | | | |
| Nguyen et al. [227] | | ✓ | | | | | | | | ✓ | | |
| Bailas et al. [412] | ✓ | | | | | | | | | | | |
| Machen et al. [251] | ✓ | | | | | | ✓ | | | ✓ | ✓ | |
| Bellavista et al. [185] | | | | | | | | ✓ | ✓ | | | |
| He et al. [160] | ✓ | ✓ | | | | | | ✓ | ✓ | | ✓ | |
| Cheng et al. [204] | ✓ | | | ✓ | | | | | ✓ | ✓ | | ✓ |
| Zhang et al. [104] | | | | | ✓ | ✓ | | | | | | |
| Abbas et al. [81] | | | | | | ✓ | | | | | | |
| Ali et al. [363] | ✓ | | | | | | | | ✓ | | | |
| Abeshu and Chilamkurti [364] | ✓ | | | | ✓ | | | | ✓ | | | |
| Ceselli et al. [217] | ✓ | ✓ | | | | | | ✓ | | | ✓ | |
| Amadeo et al. [326] | | | | | ✓ | | | | | ✓ | | |
| Cao et al. [365] | ✓ | | ✓ | | | | | ✓ | | | | |
| Shirazi et al. [105] | | | | | ✓ | ✓ | | | | | | |
| Plumb and Stutsman [413] | ✓ | | | | | | | | ✓ | | | |
| Oteafy and Hassanein [161] | | | | | | ✓ | | | | | | |
| Muhammad et al. [366] | ✓ | | | ✓ | | | | | | | | |
| Tanganelli et al. [327] | ✓ | | | | | | | | ✓ | ✓ | | |
| Li et al. [162] | ✓ | ✓ | | | | | | | ✓ | ✓ | ✓ | |
| Sharma et al. [283] | | | ✓ | ✓ | ✓ | | | | ✓ | | ✓ | |
| Jia et al. [218] | ✓ | | | | | | | | ✓ | | ✓ | |
| Cui et al. [54] | ✓ | | ✓ | ✓ | | | | ✓ | ✓ | | ✓ | |
| Shen et al. [430] | | | | | ✓ | | | | | | | |
| Rausch et al. [284] | ✓ | | | ✓ | | | | ✓ | | | ✓ | |
| Bi et al. [285] | ✓ | ✓ | | | | | | ✓ | | | ✓ | |
| Yaghmaee and Leon-Garcia [367] | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | |
| Rimal et al. [219] | ✓ | | | ✓ | | | ✓ | | ✓ | | ✓ | |
| Puthal et al. [163] | ✓ | | | | ✓ | | | | ✓ | | ✓ | |
| Lyu et al. [315] | ✓ | | ✓ | ✓ | | | | | | | | |
| Bavier et al. [414] | | | | ✓ | | ✓ | | | ✓ | | | |
| Lyu et al. [316] | ✓ | ✓ | | ✓ | | | | | ✓ | ✓ | | |
| He et al. [431] | | | | | ✓ | | | | | | | |
| Jeong et al. [186] | ✓ | | ✓ | | | | | ✓ | | | | |
| Li et al. [317] | | ✓ | ✓ | | | | | | ✓ | | ✓ | |
| Carrega et al. [415] | ✓ | | ✓ | ✓ | | | | | | | ✓ | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rafetseder et al. [391] | | | | | | | | | | ✓ | ✓ | ✓ |
| Jeong et al. [205] | | | | | | | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Varadi et al. [114] | | | | | ✓ | ✓ | | | | | | |
| Chen and Hao [286] | ✓ | | ✓ | | | | | | ✓ | | ✓ | |
| Acharya and Gaur [20] | | | | | | | | | ✓ | | | |
| Raagaard et al. [164] | ✓ | | | | | | | | ✓ | | ✓ | |
| Anglano et al. [228] | | ✓ | ✓ | | | | | | | | ✓ | |
| Yannuzzi et al. [115] | | | | | | ✓ | | | | | | |
| Nasr et al. [187] | ✓ | | | | | | ✓ | | | | | |
| Moustafa et al. [188] | | | | | | ✓ | | | | | | |
| Lera et al. [275] | | | | ✓ | | | | | | ✓ | | |
| Gupta and Ramachandran [206] | ✓ | | | | | | ✓ | | ✓ | ✓ | | ✓ |
| Mayer et al. [392] | | | | | | | | | ✓ | | | ✓ |
| Pahl et al. [165] | | | | | ✓ | | | | | | | |
| Ahn et al. [368] | ✓ | | | | ✓ | | | | | | | |
| Harchol et al. [116] | | | | | | | ✓ | ✓ | | ✓ | | |
| Mohan et al. [166] | ✓ | ✓ | | | | | | | ✓ | ✓ | ✓ | |
| Xu et al. [189] | | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | | |
| Wang et al. [369] | ✓ | | | | | | | | ✓ | | | |
| Kastanakis et al. [276] | ✓ | | | | | | | | | ✓ | ✓ | |
| Zhang et al. [416] | ✓ | | ✓ | | | | | | | | | |
| Silvestro et al. [167] | ✓ | ✓ | | ✓ | | | | | | ✓ | ✓ | |
| Németh et al. [252] | ✓ | ✓ | | | | | | | | | ✓ | |
| Körner et al. [393] | ✓ | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| Zhao et al. [207] | ✓ | | | | | | | | ✓ | | | ✓ |
| Wang et al. [277] | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | |
| Hao et al. [318] | ✓ | | | | | | | | ✓ | ✓ | | |
| Corneo and Gunningberg [370] | ✓ | | ✓ | | | | | | | | | |
| Wen et al. [190] | ✓ | | | ✓ | | | | | ✓ | ✓ | | |
| Noghabi et al. [208] | | ✓ | | | | | | | ✓ | | ✓ | ✓ |
| Bhardwaj et al. [371] | | | | | ✓ | | ✓ | | | | | |
| Ge and Wang [372] | ✓ | | | | | | | | ✓ | | | |
| Okada et al. [328] | ✓ | | | | | | | ✓ | ✓ | | | |
| Giang et al. [394] | | | | | | | | | | | ✓ | |
| Mao et al. [373] | ✓ | | | | ✓ | | | | | | | |
| Nguyen et al. [319] | ✓ | | | | ✓ | | | | ✓ | | ✓ | |
| Biookaghazadeh et al. [435] | | | ✓ | ✓ | | | | | ✓ | ✓ | | |
| Garg et al. [168] | ✓ | ✓ | | | | | ✓ | | | ✓ | ✓ | |
| Li et al. [374] | ✓ | | | ✓ | | | | | | ✓ | | |
| Coutinho et al. [395] | | | | | | | | | | ✓ | ✓ | ✓ |
| Mortazavi et al. [253] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | |
| Elias et al. [375] | ✓ | | | ✓ | | | | | | | | |
| Sathiaseelan and Secker [191] | | | | ✓ | | | ✓ | | ✓ | ✓ | | |
| Zang et al. [117] | | ✓ | | | | | | | | ✓ | | |
| Hung et al. [278] | ✓ | ✓ | | | | | | | | ✓ | ✓ | |
| Deyannis et al. [320] | | | | ✓ | ✓ | | | | ✓ | | | |
| Jeong et al. [321] | ✓ | | | | | | | | ✓ | | | |
| Psaras et al. [118] | | | | | | ✓ | | | ✓ | | | |
| Kim et al. [119] | | ✓ | | | | ✓ | | | | ✓ | | |
| Yekta and Lu [432] | ✓ | ✓ | | | ✓ | | | | | | | |
| Cherrueau et al. [120] | | | | | | | | | | | ✓ | |
| Venanzi et al. [329] | | | ✓ | | | | | | | | | |
| Anglano et al. [229] | ✓ | ✓ | ✓ | | | | | | | ✓ | ✓ | |
| Akrivopoulos et al. [417] | ✓ | | | ✓ | | | | | | ✓ | | |
| Meiklejohn et al. [169] | ✓ | ✓ | | | | | | | ✓ | | ✓ | |
| Ran et al. [376] | ✓ | | ✓ | ✓ | | | | | ✓ | ✓ | | |
| Li et al. [170] | ✓ | | | | | | | | ✓ | | | |
| Kim et al. [433] | | | | | ✓ | | | | | | | |
| Gedeon et al. [220] | ✓ | | | | | | | | ✓ | | | |
| Talagala et al. [287] | | | | | | | | | ✓ | | ✓ | |
| Liu et al. [377] | ✓ | | | | | | | | | ✓ | ✓ | |
| Zeng et al. [230] | ✓ | ✓ | ✓ | | | | | | | ✓ | | |
| Skarin et al. [418] | ✓ | | | | | | | | ✓ | ✓ | | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Syamkumar et al. [121] | | | | | | ✓ | | | | | ✓ | |
| Benson et al. [192] | ✓ | | | | | | ✓ | | ✓ | ✓ | | |
| Liu et al. [396] | ✓ | | | ✓ | | | | | ✓ | | | ✓ |
| Gupta et al. [209] | ✓ | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| Bruschi et al. [254] | ✓ | | | | | | | ✓ | ✓ | | ✓ | |

| Paper | QoS | Cost | Energy | Bandwidth | Security | Foundation | RAS | Mobility | Scalability | Heterogeneity | Management | Programmability |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Tao and Li [378] | | | | | ✓ | | | | ✓ | ✓ | | |
| Chang et al. [122] | ✓ | | | | | ✓ | | | | | | |
| Carmo et al. [193] | | | | ✓ | | | | | ✓ | | ✓ | |
| Xu et al. [379] | ✓ | | | ✓ | | | | ✓ | ✓ | ✓ | | |
| Abdelwahab et al. [123] | ✓ | | | | | | | ✓ | ✓ | ✓ | | |

setting is the "distributed" version of the problem (referred to as *online*) where there is no central authority knowledgeable of the tasks, nor is the full information about the tasks, network, or nodes known. Note that, in the online scenario, information on each job is unknown before its release.

#### 4.11.2 *Cooperative Offloading and Load Sharing.*

The authors in [88] propose a delay-minimizing collaboration and offloading policy for fog-capable devices that aims to reduce the service delay for IoT applications. They also develop an analytical model to evaluate the policy and show how the proposed framework helps to reduce IoT service delay. In [310], the authors proposed a locality-aware workload sharing scheme for mobile edge computing environments. In the proposed scheme, each node is aware of its neighboring nodes and their current workloads and utilizes such information for workload sharing in case of high workload. Another distributed latency-aware task processing and offloading model is proposed in the study [307]. In this model, gateways, which act as fog nodes, dynamically exchange their processing and storage capability information. Based on this information, fog nodes probabilistically forward their task to their neighboring fog nodes or the cloud, when there is a limit in local processing or storage.

The authors in [358] argue that offloading computation-intensive tasks, such as those of image recognition, to fog nodes is not always a right decision. The last-mile network latency due to the wireless communication may not be tolerable for some applications; also, the fog nodes themselves may become the bottleneck of processing delay, if many tasks are offloaded to them. Hence, they propose to use the available resources on edge devices and propose Precog, a collaborative scheme between edge devices and fog nodes to prefetch parts of the trained models of image recognition onto the device. Comparably, the authors in [313] introduce the notion of edge computing coalition, which is a collaborative edge-based resource pool of small cell base stations with cloud computing capabilities to serve computation requests. The collaborative edge computing scheme accommodates more computation workloads by offloading workload from overloaded nodes to lightly-loaded nodes. The coalition framework is based on coalition game theory, which follows a payment-based incentive mechanism to form stable groups, and which builds a social trust network for managing risks among edge nodes.

The authors in [309] propose and formulate a cooperative offloading policy between two edge data centers for load balancing. The model is based on a simple rule: if a service request arrives at one data center when its buffer is full, the request is offloaded to the other cooperating data center and served by that data center. On the other hand, the study in [312] analyzes an offloading policy between multiple fog data centers installed at the edge of the network in a ring topology. The study also quantitatively models and estimates the gain achieved via cooperation between neighboring fog data centers in a ring topology.

#### 4.11.3 *Offloading in Dynamic/Uncertain Fog Networks.*

Lee et al. study the problem of fog network formation and task distribution in a hybrid cloud-fog architecture [296]. Their framework differs from other studies on task allocation for fog nodes by accounting for the dynamic formation of a fog network. Since the locations fog node neighbors is an uncertainty, the authors use an online approach for quickly obtaining information of the fog network and minimizing computational latency accordingly. Their online k-secretary algorithm allows a given fog node to observe its unknown environment and determine how to offload computational tasks. A recent study that investigates the computation offloading in an uncertain wireless environment is the work of authors in [308]. The authors study the computation offloading when the mobile users' behavior is subjective and may deviate from rationality. In this framework, which is modeled as a non-cooperative game, users compete for limited communication resources.

#### 4.11.4 *Fog Offloading for Robotics.*

The study [292] proposes a scheme to estimate the processing time of robotic tasks in fog networks. They use OS profiling tools to roughly measure the execution times of runtime task on heterogeneous hardware. Based on the estimated computation time, the robot can decide (using a basic offloading condition) whether to do the computation task locally or offload to a nearby fog node. The authors also profile conventional robotic runtime algorithms to estimate computational times more accurately. To do so, they estimate processing time of multiple image, video, and map processing algorithms using OpenCV. In their other study, the authors investigate the problem of computation offloading in fog computing [293]. An optimization formulation is presented conserving the computation, energy, and communication overheads when making an offloading decision. Networked robotics is proposed as the motivating example, where the objective is to perform a computationally intensive task over the collected data by robots, to actuate in a short time.

#### 4.11.5 *Privacy-aware Offloading and Scheduling.*

Privacy protection is also a challenge when offloading tasks in fog networks. The authors in [304] consider a system in which the mobile device generates tasks and the objective of the mobile device is to find a scheduling policy that minimizes expected long-term cost. The authors observe that in their formulated problem, the privacy issues of location and usage pattern are ignored. To this end, they propose a heuristic privacy metric that jointly quantifies location and usage pattern.

*4.11.6 Energy-aware Offloading and Scheduling.* Deng et al. focused on investigating power consumption and network delay tradeoff in cloud-fog services by developing a computation-based framework and workload scheduling [124]. Similarly, Xiao et al. study the workload offloading problem in fog networks [295]. In their scheme, fog nodes in their scheme can process or offload to other fog nodes part of the workload that was initially sent to the cloud. Fog nodes decide to either offload the workload to neighbors or locally process it, under a given power constraint. The authors study the tradeoff between quality of service and power efficiency when considering fog offloading. They propose a distributed optimization algorithm that solves the problem of optimal workload allocation to maximize QoS in terms of response time.

*4.11.7 Quality of Results in Offloading.* The authors of [111] propose a systematic optimization framework with the key idea that relaxing Quality of Results (QoR) in applications where a perfect result is not always necessary. Relaxing QoR alleviates the required computation workload and enables a significant reduction of response time and energy consumption. For the proposed framework, the authors consider a mobile edge environment where the computing tasks can be divided, offloaded, and processed in parallel by distributed edge nodes. Thus, the goal of the framework is to minimize both response time and energy consumption by jointly optimizing the selection of edge nodes' QoR levels and task assignments to all edges.

*4.11.8 Scalability Issues in Task Scheduling and Dispatching.* The problem of task scheduling and dispatching in edge clouds has been studied extensively by several researchers in the area [231, 262, 288, 303, 311]. One example is the work of Tan et al. [303], where they formulate the problem of online job scheduling and dispatching in edge clouds. According to this study, the job scheduling problem is to determine which task should be served first, and the job dispatching problem is to determine where to send the job, based on latency, required resources, etc. As a future direction for task scheduling and dispatching, it is suggested to investigate the scalability of the proposed frameworks. Often, the framework fails to scale to the large magnitude of edge cloud networks, since, for example, when an IoT device generates a job, it has to be sent to all edge cloud nodes, or all edge cloud nodes need to calculate some function and inform the IoT device about the suitable edge cloud. Often, this has to be done for every job of every IoT device; hence the scalability and communication overhead are the two major issues associated with the proposed algorithms.

## 4.12 Operation: Resource Discovery

In this subsection, we summarize the studies that have investigated the problem of resource discovery or selection in fog computing. By "resources," we mean resources in the fog networks such as IoT nodes, fog nodes, nearby devices, fog services, etc.

*4.12.1 IoT Resource Discovery and Selection.* The goal in IoT resource discovery and selection is to provide applications with global discovery and access of IoT resources irrespective of their location [327]. In order to preserve the distributed nature of the federation of IoT devices, in [327] the service is realized by IoT gateways (fog nodes) through a P2P overlay. The service is implemented using a distributed hash table (DHT), where information about all available IoT resources is stored for global lookup store.

ACACIA [322] uses context awareness and employs LTE-direct for service discovery, which is a proximity service discovery technique using D2D communication in IoT. In [61], edge computing nodes are used as cloud agents near the edge to discover, virtualize, and form a cloud network of IoT devices, named Cloud of Things. This network is a geographically distributed infrastructure, in which cloud agents continuously discover resources of IoT devices and pool them as cloud resources. Similarly, in FocusStack [138], IoT devices can be selected and orchestrated using their geolocation information. Comparably, in [444] a semantic-based and space-efficient routing protocol for IoT service discovery is proposed.

*4.12.2 Fog Resource Discovery and Selection.* The paper [325] examines the problem of discovering surrogates, which are micro-clouds, fog nodes, or cloudlets, used by client devices to offload computation tasks in a fog computing environment. In order to enable the discovery and selection of available surrogates, the authors propose a brokering mechanism in which available surrogates advertise themselves to the broker. The broker receives client requests and considers a number of attributes such as network information, hardware capabilities, and distance to find the best available surrogate for the client. The proposed mechanism is implemented on off-the-shelf home routers. The authors in [196] discuss a comprehensive architecture for resource (container) selection in fog nano data centers. They introduced a 5-layered framework for task scheduling over containers, which selects a container based on energy-efficiency to meet the users' SLA requirements. Container selection and task scheduling occur through a cooperative game between special middle entities called brokers. For container scheduling and migration, Docker is used, which also helps schedule tasks over containers.

## 4.13 Operation: Applications

In this subsection, we survey the papers related to fog computing that have used the concept of fog to develop and propose new applications in other domains.

*4.13.1 Data Stream Processing.* Cloud-based data stream processing applications are not able to keep up with geo-distributed IoT systems. Cheng et al. designed GeeLytics, an edge analytics platform, to process real-time data streams from network edges and in the cloud [336]. To process IoT data streams, the authors design their platform to account for unstructured stream data that is constantly generated, mobility and colocation of sensors, low latency, heterogeneity, and ubiquity. Jayaraman et al. propose a context-aware real-time data analytics platform, CARDAP, to enable energy-efficient data delivery strategies in mobile crowdsourcing applications [333].

Microsoft Research in their paper [335] has developed a real-time video analytics system which relies on edge computing. The proposed system processes live camera feeds from the traffic intersections in the city of Bellevue, Washington, and raises alerts on anomalous traffic patterns. The system could be expanded to operate in other cities and can identify dangerous traffic patterns to reduce traffic casualties eventually. Similarly, Yang in the article [332] describes general model and architecture of fog data stream processing and analytics. The framework in [331] enables fog nodes to support query evaluations, specifically with weather data, in a federated environment in which high data volume is expected. The authors in their study [361] propose the use of vehicles as fog nodes, to process live video streams from the in-vehicle dashboard-mounted cameras; unlike smartphones that are constrained in compute resources, vehicles can support efficient computing platforms.

*4.13.2 Bandwidth Savings.* The authors of [349] propose a fog-based crowdsourcing framework for precise task allocation and secure data deduplication. Fog nodes in their scheme can detect and erase the repeated data in crowdsourced reports without deducing any information. Fog helps in data size reduction by erasing duplicate data. The researchers from Georgia Institute of Technology built AppFlux, a novel mobile app streaming system based on edge-clouds, for fast and efficient delivery of mobile apps and their updates [352]. This approach also relieves users from having to deal with app updates and could potentially save bytes on their mobile plans. EdgeCourier proposed in [51] uses edge computing to address the bandwidth issues caused by cloud-based office applications. In a measurement study, they show that contemporary cloud storage office services (e.g., word editor, or spreadsheet) consume unnecessary bandwidth since they transmit the whole file when an update happens. They design an effective office-document-aware incremental sync approach, and EdgeCourier, that uses edge-hosted unikernels for low-bandwidth mobile document synchronization in cloud storage services.

*4.13.3 Data Analytics.* Another study to make use of edge computing is the work in [354] that introduces a Wi-Fi-based in-bus monitoring and tracking system that observes mobile devices and provides analytics about people both within and outside the vehicle. The system can further use the data that is collected by the vehicle-mounted wireless device to track passenger movements, detect pedestrian flows, and evaluate how external factors impact human mobility, which provides useful analytics to transit operators. Another use of resources near the edge and edge computing is in vehicular applications [356], shown by the researchers of the University of Michigan. Edge computing is used to do some analysis on the user's interactions with the car's application, to determine what priority the current interaction should have and how much of the driver's attention should be demanded.

*4.13.4 Healthcare.* Several studies have considered the use of fog in healthcare [340, 359]. In [359] the authors introduce processing ECG features using fog nodes, which results in low-bandwidth and low-latency data processing. In [340], the authors present a hierarchical fog-assisted computing architecture for remote IoT-based patient monitoring systems. The hierarchical computing scheme enables partitioning of analytics and decision making between the fog and the cloud and deploys. The idea is based on mapping the heavy training procedures in the cloud while outsourcing the trained hypothesis to the fog nodes periodically, and exploiting the knowledge at the edge.

Through the utilization of the MQTT publish/subscribe protocol and fog computing paradigm, Zao et al. implemented a pervasive neuroimaging system to demonstrate the benefits of fog computing [334]. They used mobile devices to act as an interface, fog servers for brain state classification, and cloud servers for further brain state analysis and archiving. Thus, the authors were able to take advantage of fog computing to classify brain states in real-time. Other researchers introduced the use of fog computing in measuring ultraviolet (UV) radiation via smartphones [357]. Due to the sensitivity of CMOS sensors in mobile phone cameras to UV, the researchers have found that mobile phones have potential to measure UV radiation. Through fog servers, UV measurement results can be gathered and improved to provide more accurate UV measurements.

*4.13.5 Video and Game Analytics.* Video analytics are either computation-intensive or bandwidth hungry. Even with mobile cloud computing, there are still issues of unpredictable latency, unexpected service outage, and limited bandwidth. To address this, the authors in [306] present an edge computing platform called Latency-Aware Video Edge Analytics (LAVEA) to provide low-latency video analytics at places closer to the users. Cloud gaming comes with disadvantages such as long response latency, user coverage, QoS, and bandwidth cost. The authors in [323] explore approaches to deal with the challenges of thin-client massively multiplayer online gaming and propose a lightweight fog-based system called CloudFog. Fog is formed by idle machines that are close to the end-users and connect to the cloud. In CloudFog, the intensive computation of the new game state of the virtual world is conducted in the cloud which then sends update messages to fog nodes. The fog nodes update the virtual world, render game videos, and stream videos to the players.

*4.13.6 Image and Face Recognition.* The authors of [271] focus on image recognition based mobile applications that are latency sensitive and are soft real-time in nature. They present the idea of using an edge server as a cache with computing resources. The authors show that using an edge server as a typical web cache does not reduce latencies much, and therefore propose Cachier. Cachier is an image recognition cache that leverages the spatiotemporal locality of requests by storing appropriate requests locally and minimizes expected latency by dynamically adjusting its cache size.

Traditionally, face recognition includes face identification and resolution, and requires performing computation-intensive tasks in the cloud and transmitting raw facial images to the cloud, which is bandwidth intensive. The authors in [342] observe that migrating part of the resolution tasks to fog nodes and transmitting only the feature value to the cloud can significantly reduce network traffic. To this end, they propose fog-based face identification and face resolution frameworks.

*4.13.7 Artificial Intelligence and Machine Learning.* [345] presents edge stochastic gradient descent (EdgeSGD), a decentralized SGD algorithm for solving linear regression problem with the objective of estimating the feature vector on the edge node. EdgeSGD is used to predict subsurface seismic anomaly via real-time imaging. The edge nodes form a mesh network, and the algorithm obtains the image by collaboratively optimizing the objective function over the edge network. The proposed algorithm is entirely decentralized and does not require synchronization. Comparably, an edge-assisted adaptive deep learning framework for mobile object recognition is introduced in [350].

## 4.14 Software and Tools

In this subsection, we describe the software and tools that are developed for fog computing.

*4.14.1 Simulation and Emulation.* Gupta et al. focused on implementing resource management techniques in fog computing to measure latency and throughput, and implemented iFogSim[5], a Java-based tool for simulation of fog networks [380]. iFogSim relies on CloudSim, a cloud simulation tool that enables modeling and simulation of cloud systems and application provisioning environments. iFogSim supports cloud-only placement and edge-ward placement to demonstrate the scalability of fog-based applications. An extension[6] of iFogSim to support mobility through migration of VMs between cloudlets is implemented in [385]. Another extension is proposed in [390] to simulate scenarios with strategies aiming to optimize data placement[7].

Similarly, the authors in [388] propose another edge computing simulation environment, EdgeCloudSim[8], that considers both network and computational resources and covers all aspects of edge computing simulation modeling, including network and computational modeling. Similar to iFogSim, EdgeCloudSim relies on CloudSim as well. Additionally, EdgeCloudSim provides a modular architecture to provide support for a variety of critical functionality and supports simulating multi-tier scenarios where multiple edge servers are running in coordination with upper layer cloud solutions.

Nevertheless, compared to simulation, emulation supports both repeatable and controllable experiments with real applications. The authors in [392] propose their software implementation of EmuFog[9], an extensible emulation framework for fog computing. EmuFog enables the user to design network topologies, embed fog nodes in the topology, and run Docker-based applications on those nodes connected by an emulated network. The modules of EmuFog are easily extensible, although EmuFog provides a default implementation for them [392]. Another emulation environment for fog computing is FogBed[10] that is based on Mininet and Docker [395].

*4.14.2 Edge Computing Middleware.* One of the software implementations for fog is developed by the researchers of the University of Wisconsin, named ParaDrop [383]. ParaDrop is an edge computing platform that runs on WiFi access points to enable edge computing at the extreme edge of the network. Developers can use this edge computing platform to deploy services, which should be based on Docker containers, on the WiFi access points. ParaDrop is available as an open source project[11], along with the documents and tutorials. ParaDrop has three components: ParaDrop access points, the ParaDrop controller, and the ParaDrop API. Using ParaDrop API, cloud services could be dynamically deployed on access points using the ParaDrop controller, which is a back-end controller that developers interact with to develop their desired services.

Analogous to offloading computation and task to either cloud or fog, a group of researchers from Georgia Institute of Technology argue the utility of "onloading" cloud services to the edge of the network to address the bandwidth and latency challenges of IoT networks [66]. They view the cyber-foraging (e.g., code/task offloading) research domain as client-based methodology, while their proposed approach is "backend-based" cyber foraging or onloading. They define onloading specific services (e.g., caching, or aggregating traffic) near end-users, with a goal of minimizing user-perceived latency. They propose AirBox, a software platform based on containers for fast, scalable and secure onloading of edge services.

Bruneo et al. designed Stack4Things[12], a framework based on OpenStack IaaS middleware that adopts a cloud-oriented model for IoT resource provisioning [381]. Their framework allows injected code at runtime through the cloud, which they define as "contextualization." Similarly, the authors in [150] revise OpenStack Nova service for compatibility with fog/edge systems, by leveraging a distributed key/value store instead of the centralized SQL backend[13].

One commercial edge computing platform for IoT gateways is the *Everyware Software Framework* recently developed by Eurotech[14]. Similarly, EdgeX Foundry[15] is a vendor-neutral open source project building a common open framework for edge computing. UC Berkeley researchers also implemented a generic and platform-agnostic open source edge computing framework[16] called Open Carrier Interface (OCI) [393].

*4.14.3 Edge-based Data Analytics.* Xu et al. proposed edge analytics as a service (EAaaS) to promote a lightweight, scalable, and low-latency service model for IoT devices [382]. The primary motivations for EAaaS are the lack of desired real-time responsiveness, a pricy pay-as-you-go model, and data privacy concerns associated with cloud-centered IoT analytic services. EAaaS provides RESTful interfaces for outside applications, an edge analytics agent on the gateway side, and an edge analytics SDK to allow users to develop methods for integrating with devices. EAaaS is provided as a part of IBM Watson IoT platform on IBM Bluemix Cloud. In their work, the authors plan to provide software upgrade capabilities as part of the existing RESTful services and utilize machine learning for existing analytic models. Comparably, SpanEdge[17] provides a programming environment that allows programmers to specify parts of their applications that need to be close to the data source, without knowledge of the number of data sources and their geographical distributions [355].

*4.14.4 Distributed Deep Learning.* Harvard University researchers recently proposed a framework and its software implementation[18] for distributed deep neural networks (DDNN), that can span over

[5]Available at https://github.com/cloudslab/ifogsim
[6]Available at http://www.lrc.ic.unicamp.br/fogcomputing/
[7]Available at https://github.com/medislam/iFogSimWithDataPlacement
[8]Available at https://github.com/CagataySonmez/EdgeCloudSim
[9]Available at https://github.com/emufog/emufog
[10]Available at https://github.com/fogbed/fogbed
[11]Available at https://paradrop.org
[12]Available at http://stack4things.unime.it
[13]More information and code available at http://beyondtheclouds.github.io
[14]Available at https://esf.eurotech.com
[15]Available at https://www.edgexfoundry.org
[16]Available at https://github.com/marckoerner/oci
[17]Available at https://github.com/Telolets/StormOnEdge
[18]Available at https://github.com/kunglab/ddnn

cloud, fog, and end devices [344]. The framework map sections of a deep neural network (DNN) onto a distributed computing graph. While the resulting DDNN allows for deep inferences in the cloud, it accommodates fast and localized inference via some shallow layers of the DNN at the edge and end devices. DDNN inherently enhances sensor fusion, data privacy, and fault tolerance for DNN applications. According to this study, DDNN can reduce the communication overhead by a factor of 20x, compared with the traditional method of sending raw sensor data to the cloud.

*4.14.5 Trust Establishment.* A traditional solution for establishing trust between two entities is to create and share credentials in advance, and then to use a third-party to validate the credentials of the nodes. Nevertheless, the characteristics of some environments (e.g., environments with intermittent or no Internet connectivity), do not consistently provide access to third-party authority. A team of researchers from CMU designed and implemented a system for establishing a trusted identity solution based on cloudlets [389]. The authors discuss a software implementation[19] of an in-the-field solution for establishing trusted identities in disconnected edge environments.

## 4.15 Testbeds and Experiments

Several studies in the fog computing research area implement testbeds or conduct experiments to verify the concepts and ideas experimentally. In this subsection, we look at such studies.

*4.15.1 Where is The Edge?* One might ask “Where is the edge of the network?” Edge computing or MEC as general concepts do not restrict the specific location of the edge compute and storage nodes, and most researchers assume they are “at the edge of the network” or “in the users’ proximity.” A recent measurement study by Bell Labs researchers [409] shed light on actual values of delay and provided a realistic picture of LTE deployments for edge computing. They found that the first hop (UE to base station) imposes latency of 10-12 ms and adds a sawtooth pattern with an amplitude of about 40 ms. They also found that in some cases the latency of the first aggregation stage dominates the end-to-end latency.

*4.15.2 Experiments with Single-board Computers as Enablers of Fog Computing.* In [401], the authors proposed a container-based architecture for edge-based PaaS, in which applications are orchestrated between nodes at the edge. To show that the proposed architecture can meet the requirements of edge computing, such as cost-efficiency, and low power consumption, the researchers implemented their solution on a cluster of Raspberry Pi (RPi) devices, which are resource-limited devices. Researchers in [408] have looked at the problem of bringing cloud services to rural and remote areas in developing countries, where building large and expensive data centers may not be feasible. They propose a hardware platform based on a cluster of single-board computer (e.g., RPi) for making cloudlet nodes in rural and remote areas that offer cloud services.

Elkhatib et al. [398] propose the use of micro-clouds, which are small deployable computational infrastructures, to deploy fog networks. A device as small as an RPi can be used to create a micro-cloud. The authors note that, unlike mini data centers, micro-clouds are portable, easy to set up, low cost, and can be deployed in rougher environments. They run tests to compare the traditional cloud architecture to the micro-clouds and concluded that micro-clouds are more suitable for scenarios where the cloud server is far away, suffering from high latency. Besides, the boot time of micro-clouds is significantly faster than cloud. New RPi models are capable of booting up 40% faster than an Amazon EC2.

*4.15.3 Experiments with Lightweight Virtualization Technologies as Enablers of Fog Computing.* In [403], the authors evaluated three container orchestration tools, namely Google Kubernetes, Docker Swarm and Apache Marathon, to study their applicability for IoT networks. To do so, they defined three requirements that an effective container orchestration solution for such an environment must meet. First, adding/removing a new fog node to a cluster must be seamless, requiring a minimal software package installation on that node. Second, scheduling an application to a specific fog node must be possible. Third, all available hardware resources on a fog node must be accessible by the container on top of it. Therefore, they proposed a new container orchestration framework based on Docker Swarm for fog computing environment that can meet all of the requirements above.

Similarly, The paper [404] presents an evaluation of Docker as a container to host applications at the edge for enabling edge computing. The evaluation is based on four fundamental requirements for edge computing, namely, deployment and termination, resource and service management, fault tolerance, and caching. To evaluate and examine Docker as a candidate for edge computing, a testbed with a data center and three edge sites is simulated, and the four requirements of edge are evaluated in this testbed. Authors in [445] study how lightweight virtualization technologies, such as containers and unikernels, can be integrated with edge architectures and be suitable for IoT pervasive environments. They present three IoT use cases, in which lightweight virtualization solutions can bring benefits and desirable design flexibility.

*4.15.4 Experiments with High Computation Applications.* Interactive wearable cognitive assistance application is often considered a killer app for edge computing and cloudlets [410]. In an empirical study on latency [410], the performance of several such applications is evaluated. The authors in [406] proposed a new serverless architecture for MEC environments, in which mobile app computation is offloaded among edge nodes to reach high throughput while keeping the latency low. To evaluate their architecture, they studied a mobile augmented reality application, in which captured frames from camera must be analyzed to detect the point of interest of objects. The authors in their recent study [413] investigate the benefits game developers may obtain by exploring emerging edge cloud technology, specifically use of Google’s Edge Network. They demonstrate how massively multiplayer online games benefit from this new system through simulations. Another recent experimental study is the work in [416] where authors compare the performance of machine learning packages on the edges, including TensorFlow, Caffe2, MXNet, PyTorch, and TensorFlow Lite.

[19] Available at https://github.com/SEI-TTG/pycloud/wiki

## 4.16 Hardware and Protocol Stack

In this subsection, we summarize the literature that introduced specific hardware for implementation of fog computing, cloudlet, F-RAN, etc., or that proposed a protocol stack for fog.

*4.16.1 Hardware.* Intel recently has released a documentation for Fog Reference Unit, a reference design in a self-contained enclosed chassis for testing and demonstration of fog use cases [446]. Intel's Fog Reference Unit can be seen as a generic fog node. The authors of [212] propose to use optical fiber networks with MEC and present a generic fiber-wireless (FiWi) architecture for IoT. The network architecture consists of two parts: the backbone network for connecting to centralized cloud servers, and the FiWi network for providing MEC services for IoT devices. The authors in [214] propose the use of time division multiplexed passive optical networks (TDM-PON) and optical network units (ONUs) for designing a cloudlet-based network. The architecture for fog computing in [128] is based on the general-purpose processor (GPP) platform, which allows the architecture for processing general and shared resources.

In [434], the authors focus on the requirements of an industrial IoT (IIoT) gateway with respect to heterogeneous network communication, management, big data, and other services. They describe the advantages of using a multi-Microcontroller architecture. Their architecture incorporates a high-speed parallel bridge controller using a reconfigurable field programmable gate array (FPGA) to overcome the serial communication bottlenecks in a traditional MCU architecture. The proposed architecture of the IIoT multi-Microcontroller gateway consists of three major modules: a master controller for the cloud, a high-speed bridge controller for data and instruction exchange, and slave controllers for IoT management and database operations. The high-speed bridge controller module is the core of the IIoT gateway and is responsible for packaging tasks and controlling communication with the other modules.

The paper [266] proposed a Cloud-Fog Radio Access Network (CF-RAN) by jointly considering the fog computing paradigm over a Time and Wavelength-Division Multiplexing Passive Optical Networks (TWDM-PON). The proposed CF-RAN can place services provided by the cloud onto the fog nodes by adopting NFV to process the baseband signals sent from the RRHs. Similarly, [146] focuses on interference mitigation, resource optimization, and mobility management in F-RAN. The authors first present the system architecture that illustrates how the various components in F-RAN, such as macro RRHs (MRRHs), small RRHs (SRRHs), fog computing access points (F-APs), and smart user equipment (F-UE), work together for the successful implementation of F-RAN. The MRRHs, SRRHs and the F-APs connect to the BBU pool which supports resource optimization and provides centralized storage and communications in F-RAN.

*4.16.2 Protocol Stack.* The IoT hub proposed in [171] introduces a protocol stack for an IoT gateway (e.g., fog node) that interacts with the smart object devices using a variety of network protocols such as IEEE 802.15.4, IEEE 802.11, or Bluetooth Low-Energy. The job of the application layer includes service and resource discovery, maintaining a resource directory, acting as an origin server, and providing a CoAP-to-CoAP/HTTP-to-CoAP Proxy and cache. The network and physical/link layer both work together to provide border router functionality, allowing the IoT Hub to act as a gateway/bridge between multiple constrained networks.

Another protocol stack for fog is introduced as FogOS [200]. The authors view the entire IoT ecosystem as a computer and utilize operating system concepts to create FogOS to manage this abstract computer. FogOS is composed of four main layers: service abstraction layer, application manager layer, resource manager layer, and device abstraction layer. The role of the service and device abstraction component is to provide a service APIs and device data model, respectively. Resource management then pools or slices resources of the fog and devices as needed. The application manager manages IoT applications, finds the proper edge resource for a service request, and resources of currently running applications.

## 4.17 Security and Privacy

In this subsection, we discuss the studies that consider and address security and privacy issues in fog computing or utilize fog computing to improve current security systems and protocols. In [99], the authors discussed various privacy and security challenges in fog computing and surveyed the existing literature addressing these problems. Alrawis et al. [422] explain security concerns with current IoT environments. They proposed a scheme that utilizes fog computing to address the security issues faced in IoT environments, specifically the distribution of certificate revocation information. Security and privacy issues of vehicular crowdsensing (VFCS) are discussed in [102].

*4.17.1 Location Privacy.* Ting et al. [428] proposed several strategies that can be utilized to prevent cyber eavesdroppers from tracking a user's location by observing the user's service migrations across edge clouds. The underlying idea is to use chaff services, or fake services, in addition to the real services and move them between the edge clouds to confuse eavesdroppers. The authors showed that by carefully moving the chaff services in conjunction with the real ones, tracking accuracy could be close to zero when the entropy of the user movement between edge clouds is sufficiently high.

To protect the privacy of users using applications with location-based services, several private proximity detection algorithms exist. However, these algorithms fall short in the vicinity range and come with high communication and computation costs. [339] presents a secure homomorphic protocol for private proximity detection in applications of location-based services that addresses the drawbacks of existing approaches. To achieve privacy in the data transmission process, the authors propose to use a homomorphic encryption scheme that provides fast encryption and decryption of data.

*4.17.2 Data Privacy.* The authors in [421] propose a privacy-preserving protocol for vehicular road surface condition monitoring. They propose a certificate-less signcryption scheme and a data transmission protocol for road surface condition monitoring that provides confidentiality, mutual authenticity, integrity, privacy, and anonymity. In [425], the authors proposed a new privacy-preserving scheme based on differential privacy technology for fog computing. The primary goal in this scheme is to prevent colluded nodes in a fog

**Table 7: Summary of challenges and future research directions**

| Challenge | Current Limitation | Research Direction or Potential Solution | Related Features or Objectives | Related Categories |
|---|---|---|---|---|
| Fog service SLA | SLAs are not defined for fog systems. Current SLAs are defined for cloud services or network infrastructure. | • Define new and compatible SLA for fog systems. • Design SLA management techniques and framework for fog computing. • Support for multi-vendor or provider SLA for fog systems | QoS, Cost | Architectures and Frameworks for Fog; Control and Monitoring |
| Multi-objective fog system design | Many schemes (e.g., offloading, load balancing) consider few objectives and ignore other objectives. | • Propose schemes that consider multiple objectives (e.g., latency, bandwidth, energy) simultaneously (e.g., an efficient task offloading scheme that considers bandwidth, waiting time, availability, security, and energy). | QoS, Cost, Energy, Bandwidth | Resource Analysis and Estimation; Scheduling, offloading, and Load Balancing; Testbeds and Experiments |
| Bandwidth-aware fog system design | Few works consider bandwidth savings through the use of fog computing, even though one of the promising features of fog computing is to reduce bandwidth usage of the core. | • Need more studies on bandwidth savings through the use of fog computing. • Perform measurement studies to capture the actual bandwidth usage in the presence of fog. | Bandwidth | Testbeds and Experiments; Scheduling, offloading, and Load Balancing; Control and Monitoring; Resource Analysis and Estimation; Infrastructure Design |
| Scalable design of fog schemes | Many of the existing schemes and algorithms for fog do not scale to the magnitude of IoT networks. | • Design scalable algorithms and schemes for fog systems, e.g., online task offloading scheme that does not consider individual IoT nodes for decision making • Verify scalability of the algorithm and schemes by actual implementation. | Scalability | Service Provisioning; Placement; Scheduling, offloading, and Load Balancing; Applications |
| Mobile fog computing | Most of the existing literature assumes fog nodes are fixed, or focus on the mobility of IoT devices. If fog nodes are mobile, resource availability, offloading, and resources provisioning will be more challenging. | • Propose mobile fog computing, where fog nodes can move. • Scheme for management or federation of mobile fog nodes. • Provisioning method for mobile fog services to keep the service always-available for IoT nodes. • Design of mobility-aware task offloading and scheduling schemes when fog nodes are mobile. | Mobility, Management | Resource Discovery; Concepts and Frameworks using Fog; Programming Models and Data Modeling; Service Provisioning; Security and Privacy; Scheduling, offloading, and Load Balancing |
| Fog resource monitoring | Few studies address monitoring of fog resources. Monitoring is more challenging if multiple operators use a fog node. | • Multi-operator fog resource monitoring techniques. • SDN-based monitoring software for resource monitoring and resource advertisement. | Management, programmability | Control and Monitoring; Software and Tools |
| Green fog computing | Improving the overall energy consumption of fog has not been well studied (literature considered energy-aware computation offloading, energy-aware mobility management, and federation of IoT devices to improve energy consumption). | • Use of energy harvesters and battery storages for IoT devices and sensors. • Energy-aware fog node placement, e.g., close to renewable energy resources (solar, wind, or vibration) | Energy | Infrastructure Design; Resource Analysis and Estimation |
| Support of high-speed users | Current communication protocols do not support high-speed users. | • Develop fast or stateless handshake protocols for high-speed users, e.g., users in vehicles or for automotive communication. • Develop machine-learning-based mobility prediction algorithms. | Mobility, RAS | Architectures and Frameworks for Fog; Service Provisioning; Resource Discovery |
| Fog node security | Fog nodes normally are located close to users, e.g., at the base stations, routers, or even at extreme network edge such as WiFi access points. This makes their security challenging. | • Design of physically secure fog nodes against site attacks • Design secure hardware, safe against physical damage, jamming, etc. • Design strong access-control policies for fog nodes. | Security, Heterogeneity | Security and Privacy; Infrastructure Design; Hardware and Protocol Stack |
| SDN support for fog | SDN does not have native support for fog computing. | • Enhancing and standardizing SDN (e.g., OpenFlow northbound, southbound, east-west bound interface) for fog use cases. | Foundation, programmability | Definition and Standards; Software and Tools |
| Fog node site selection | Few studies address the fog node site selection problem, which is a design problem for finding appropriate locations for deploying for nodes. | • Developing fog node site selection strategies that considers communication, storage, and computing (a communication hotspot may not be a storage or computing hotspot). • Considering cost in fog node site selection strategies (e.g., deploying fog nodes in Manhattan may be a good decision concerning latency and bandwidth, but may not be a good decision with respect to rental costs). | Cost, RAS, QoS, Energy | Infrastructure Design; Resource Analysis and Estimation |
| Resilient fog system design | Current fog networks do not consider failure or fault in the network. Also, denial of service (DoS) attacks are more possible on fog nodes, since they are more resource-constrained than cloud servers. | • Fault detection, fault prevention, and fault recovery in fog-based networks • DoS-resilient fog system design • Design a coordinated protection mechanism that considers fog and cloud to guarantee availability. | RAS, Security | Control and Monitoring; Infrastructure Design; Service Provisioning; Security and Privacy |
| Fog federation | There is no fog federation framework or software similar to that of hybrid cloud federation schemes. | • Design new schemes for the federation of fog nodes, across different operating domains. • Design resource sharing models for fog nodes from different vendors/operators • Define new pricing models for federated fog resource sharing schemes | Management, programmability | Placement; Software and Tools; Resource Discovery. Service Provisioning |

| Challenge | Current Limitation | Research Direction or Potential Solution | Related Features or Objectives | Related Categories |
|---|---|---|---|---|
| Trust and authentication in heterogeneous fog systems | Heterogeneity of fog nodes and IoT nodes makes the conventional trust and authentication protocols unsuitable for fog systems. | • Design new authentication and trust mechanisms that could cope with heterogeneity of fog nodes and IoT nodes. • Design authentication protocols for fog nodes of different vendors/operators. | Heterogeneity, Security | Definition and Standards; Security and Privacy; Hardware and Protocol Stack |
| Secure fog offloading | Offloading tasks to fog nodes might incur some security and privacy risks. | • Design secure and private offloading and load balancing schemes. • A mechanism for receivers to verify the correctness and integrity of the offloaded task. | Security, QoS | Scheduling, offloading, and Load Balancing; Security and Privacy |
| PaaS for fog computing | Lack of a PaaS for fog systems, where developers can easily develop software across fog, IoT, and cloud. | • Developing a PaaS for fog computing, which is transparent to users and supports different communication- and application-level protocols and APIs. • Developing plugins for PaaS for different fog computing applications | programmability, Management | Software and Tools; Service Provisioning; Programming Models and Data Modeling |
| Standardizing fog computing | Many independent definitions for fog (and fog-related computing paradigms) are being proposed. | • Unanimous and universally-agreed on the definition of fog computing. | Foundation | Definition and Standards |
| Hardware technologies for fog | Most studies do not use new available hardware or communication technologies. | • Use of new hardware and communication technologies, such as non-volatile storage technologies, optical networks, and FPGAs. | Scalability | Hardware and Protocol Stack |

network from learning the data shared by the IoT devices, thus preserving their privacy. The underlying idea is to introduce artificial noise to the data before outsourcing, such that the colluded node cannot infer the original data by conducting statistical analysis.

A lightweight privacy-preserving data aggregation scheme for IoT networks is proposed in [424]. The scheme enables a control center to compute the average and variance of data collected by various types of IoT sensors while preserving their privacy; fog nodes and control centers cannot learn the original data collected by devices. In this scheme, IoT devices are authenticated by the fog nodes. Also, the scheme is fault tolerant in the sense that if some IoT devices malfunction, the control center still can calculate the correct average and variance of the remaining devices without breaching their privacy. Similarly, for information sharing, a lightweight privacy-preserving fog-assisted information sharing scheme for health data collected by medical IoT devices is proposed in [341].

*4.17.3 Intrusion Detection.* Although utilizing SDN switches as fog nodes seems promising (as discussed in Section 4.10), it can increase the security risks in SDN networks. The attacker who compromises a fog node can also attempt to take advantage of the SDN switch to control the network. In [282], the authors described a man-in-the-middle attack on SDN networks relying only on the TLS protocol to secure their control channels between SDN switches and controllers. They also proposed a lightweight countermeasure utilizing Bloom Filters to detect such an attempt. Fog can be used to improve the current security system. In [426], the authors propose a new fog-based intrusion system.

*4.17.4 Secure Protocols and Secure Data Transfer.* A recent study in [427] investigates secure data transmission between IoT nodes and fog nodes. The study analytically models a threat model and some possible attacks; it then provides the necessary proofs to show that the framework is secure against the discussed attacks. The authors of [420] propose constructing leakage-resilient functional encryption schemes for the fog that provide privacy, fine-grained access control, and security against side channel attacks.

Identification and resolution of human subjects are crucial in cyber-physical systems where humans are involved. Hu et al. [337] proposed a new face identification and resolution framework that utilizes fog computing to address bandwidth issues that arise in conventional cloud-based schemes. In [337], they provided the details of the security protocols that they devised to address the security and privacy concerns that arise in their system, such as confidentiality and integrity.

## 5 CHALLENGES AND FUTURE RESEARCH DIRECTIONS

In this section, we discuss the current challenges and limitations of the research in the fog computing area, and we provide future directions and potential starting points for those challenges. The summary of the challenges and future research directions are included in Table 7. Fig. 10 shows the number of research articles in this survey that fall under each category of our taxonomy. Fig. 11 illustrates the number of research articles in this survey paper that address/support a particular objective/feature. The number of research articles may be an indicator of potential future research directions.

*5.0.1 Fog Service SLA.* Service level agreements (SLAs) are not currently defined for fog systems. Current SLAs that are used for fog systems are defined for cloud services (e.g., 99.99% availability guarantee for cloud services) or network infrastructure. Moreover, a fog system may have multiple providers/operators and span across multiple operating domains. A potential research direction is defining new and compatible SLA for fog systems (e.g., guaranteeing latency and bandwidth). Additionally, designing SLA management techniques and framework for fog computing that supports for multi-vendor or provider is another potential direction.

*5.0.2 Multi-objective Fog System Design.* Most of the existing schemes that are proposed for fog systems, such as offloading, load balancing, or service provisioning, only consider few objectives (e.g., QoS, cost) and assume other objectives do not affect the problem (e.g., [107, 239, 257, 295, 305]). A new research direction will be to design schemes that consider many objectives (e.g., QoS, bandwidth, energy, cost) simultaneously. For instance, developing an efficient task offloading scheme that considers bandwidth, waiting time, availability, security, and energy at the same time is a promising direction.

*5.0.3 Bandwidth-aware Fog System Design.* Few studies consider bandwidth savings through the use of fog computing (e.g., [51]), even though one of the promising features of fog computing is to reduce bandwidth usage in the core of the Internet. There is a need for more research on bandwidth savings through the use of fog computing. These studies could be measurement studies that capture the actual bandwidth usage in the presence of fog computing.

*5.0.4 Scalable Design of Fog Schemes.* Many of the existing schemes and algorithms for fog do not scale to the magnitude of IoT networks, since the authors neglect scalability in their fundamental design. We believe scalability is critical in designing fog systems; fog systems should be scalable so that they could be implemented in IoT networks. For instance, a scalable algorithm for fog offloading is an online offloading scheme that does not need information of individual IoT nodes for decision making (e.g., [88, 312, 313]). We encourage researchers in the fog computing area to verify the scalability of their proposed algorithms and schemes (e.g., by an actual implementation).

*5.0.5 Mobile Fog Computing.* Most of the existing literature assumes that the fog nodes are fixed, or only considers the mobility of IoT devices (e.g., [146, 291]). Less attention has been paid to mobile fog computing and how the mobile fog nodes can improve the QoS, cost, and energy consumption. When fog nodes are mobile, fog resource availability, resource discovery, task offloading, and resources provisioning will be more challenging. Mobile fog computing, where fog nodes can move and form new networks, is an interesting and challenging research direction. Moreover, designing a scheme for management or federation of mobile fog nodes is another possible direction. Along with mobile fog computing, there needs to exist new provisioning methods for mobile fog services, such that fog services are available for IoT nodes and users. Similarly, one could design a task offloading and scheduling scheme when fog nodes are mobile. An early effort for mobile fog nodes can be found in [160].

*5.0.6 Fog Resource Monitoring.* Few studies in the literature propose monitoring schemes for fog resources [151]. Monitoring is useful when multiple operators use a fog node, or when a fog node is located in a location where many users use the fog node. A possible direction is developing fog resource monitoring techniques that support multi-operator access. Use of SDN-based monitoring software for fog resource monitoring and fog resource advertisement is also a promising approach.

*5.0.7 Green Fog Computing.* Few studies in the reviewed literature have addressed the energy criterion in their system design (e.g. [23, 196, 294, 295, 333, 384]). Most of the studies on energy are about energy-aware computation offloading, energy-aware mobility management, and federation of IoT devices to improve the energy consumption of fog systems. However, improving the overall energy consumption of fog has not been well studied. Energy consumption of a fog network includes three major portions: (1) energy consumption of IoT devices sending data to the fog, (2) energy consumption of the network interconnecting IoT devices and the fog nodes, and (3) energy consumption of the fog nodes.

To reduce the energy consumption of IoT devices, use of energy harvesters and battery storage for IoT devices and sensors are potential research directions. Energy harvesters can improve energy consumptions while bringing new challenges to the system, such as uncertainty and unpredictability. In order to reduce the energy consumption of the network interconnecting IoT devices and the fog nodes, one of the potential research directions is to identify where to put fog nodes and how close they should be to the end users. Mobile fog nodes is also a compelling use case for energy consumption. To reduce the energy consumption of fog nodes, one potential research direction is to reduce the distance between fog servers and local renewable energy sources (such as solar, wind, or vibration). This problem can be addressed in different ways: the traffic from IoT devices can be rerouted to the nearest fog node that is powered by renewable energy. The other way is for telecommunication companies to identify the location of fog nodes that need the high amount of power to serve the traffic, and to encourage people to use their local renewable energy for their local micro-grid to power up their local fog nodes.

*5.0.8 SDN Support for Fog.* SDN software does not have native support for fog computing. SDN is mostly commercially viable inside large data centers or campus networks [9]. Enhancing and standardizing SDN software (e.g., OpenFlow northbound, southbound, east-west bound interface) for fog use cases is a direction we believe will ease the development of fog computing software. Moreover, with multiple vendors/operators in fog systems, there will be a need for new SDN architectures with multiple domains and hierarchies of SDN controllers.

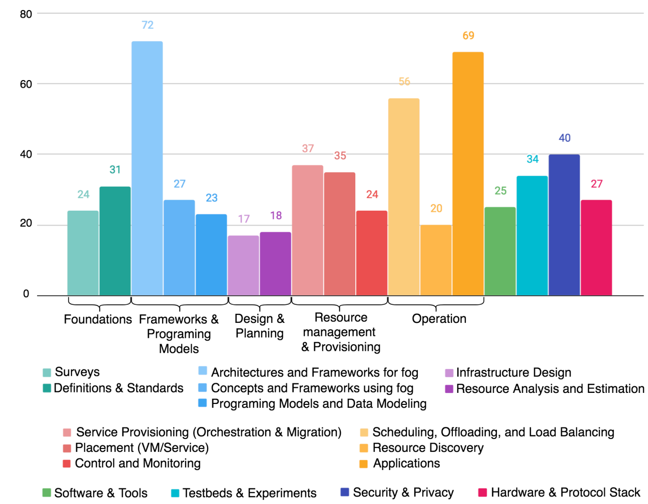


**Figure 10: A bar chart showing the number of research articles in this survey paper under each category of our proposed taxonomy.**

*5.0.9 Support of High-speed Users.* Current communication protocols that are proposed for fog computing environments do not support high-speed users, such as users in cars, users on trains, and vehicular computing. A research direction is to develop quick or stateless handshake and authentication protocols for high-speed users and automotive communication. Note that there have been already some early efforts in this field, such as the articles [100, 145, 172]; nevertheless, we are still far from having a working and resilient communication protocol for high-speed users and automotive communication for fog computing. One can use machine-learning-based mobility prediction algorithms in the design of the handshake and authentication protocols, to predict the location of the high-speed users and analyze their mobility patterns for fog computing.

In addition to handshake and authentication protocols for high-speed users, fog service provisioning for IoT applications is required to be dynamic and proactive due to the rapid changes (such as connectivity, bandwidth fluctuations, or failure) in mobile and high-speed IoT environments. To address dynamic and proactive fog service provisioning, predicting the behavior and location of IoT devices and high-speed users based on historical data or machine learning methods is another potential solution that requires further investigation.

*5.0.10 Fog Node Security.* Fog nodes are going to be placed near users, in locations such as on the base stations, or routers, or even at the extreme network edge on WiFi access points. This makes it challenging to provide security for fog nodes. Site attacks are more possible on the fog nodes than cloud data centers. A research direction may be designing secure fog node sites, safe against physical damage, jamming, etc. Also, another direction may be to designing strong access-control policies for fog nodes, so that they are secure in the presence of malicious users in the vicinity. A potential starting point for access control in fog computing can be found in [104, 420].

*5.0.11 Fog Node Site Selection.* Few studies address the fog node site selection problem, which is a design problem for finding appropriate locations for deploying nodes. Fog node site selection strategies should consider communication, storage, and computing at the same time for finding an appropriate location (a communication hotspot may not necessarily be a storage or computing hotspot). Moreover, cost should also be a deciding factor in fog node site selection strategies; deploying fog nodes in Manhattan may be a good decision concerning reduction in latency and bandwidth, but may not be a good decision with respect to rental costs. Furthermore, fog node security considerations that are discussed in "Fog Node Security" previously could also affect the fog node site selection decision. We refer the interested reader to the recent article [447] that describes the security requirements and approaches of an open fog architecture.

*5.0.12 Resilient Fog System Design.* From the reliability and availability perspective, fog services and fog networks bring new challenges to the current network and service provisioning methods. To guarantee the availability and reliability of the fog services, a coordinated service provisioning mechanism that considers both fog and cloud computing is needed. For example, if a fog service needs some functions to process a stream of data, providing extra replicas of those functions can improve the availability of the service. On the other hand, due to the limited computing resources of the fog nodes compared to the cloud data centers, allocation of the function replicas to provide availability and reliability is not a straightforward decision. As a future direction, availability may be considered in addition to constraints, such as latency, throughput, and security when designing provisioning methods for fog services.

Most of the articles in the literature about fog computing do not consider failure or fault in the fog network. Another research direction is to provide different protection and restoration mechanism across different layers. In addition, failure detection, prevention, and recovery are efficient ways to improve the availability of the fog services. Additionally, fog nodes are more prone to denial of service (DoS) attacks, since they are more resource-constrained than adequately-secure cloud data centers; also, recently compromising IoT nodes and embedded systems are becoming new sources of distributed DoS attacks [448]. Novel classes of proactive defense techniques based on moving target defense paradigm (sometimes referred to as *address mutation/randomization*) could be used to thwart DoS attacks [449, 450]. Researchers of UC Berkeley recently have proposed a resilient edge computing framework [116], which is a good starting point in the direction of resilient fog system design.

*5.0.13 Fog Federation.* Currently, there is no fog federation framework or software (similar to that of hybrid cloud federation schemes), which controls and federates fog resources across multiple operating domains. There is a need for new schemes for the federation of fog nodes, especially when they belong to different operating domains. The federation scheme should account for resource sharing models for fog nodes from different vendors/operators. Similarly, one can define new pricing models for federated fog resources. Finally, one can propose policies for new fog resource sharing schemes (e.g., P2P fog computing resource sharing model) under the federation framework. The recent article in [228] might be a good start for potential research about fog federation.

*5.0.14 Trust and Authentication in Heterogenous Fog Systems.* In addition to mobility that is discussed in "Support of High-speed Users," heterogeneity of fog nodes and IoT devices makes the conventional trust and authentication protocols unsuitable for fog systems. Also, fog nodes may belong to different vendors and operators. Hence, we need to design new authentication and trust mechanisms for new fog systems that could cope with the heterogeneity of fog nodes and IoT devices. Some researchers have started to address heterogeneity in their fog system design, such as the articles [171, 324].

*5.0.15 Secure Fog Offloading.* Offloading tasks among fog nodes might incur some security and privacy risks. The risk is when tasks containing security- and privacy-critical information is offloaded. Also, a security risk might be when a fog node becomes over-loaded (e.g., by the requests sent by a malicious user) and starts offloading security- and privacy-critical information to other fog nodes (which may be accessible to the malicious user). A research direction, hence, is to design and implement secure offloading and load balancing schemes. An early effort for privacy-aware offloading in MEC is included in [304]. Furthermore, designing a lightweight and efficient

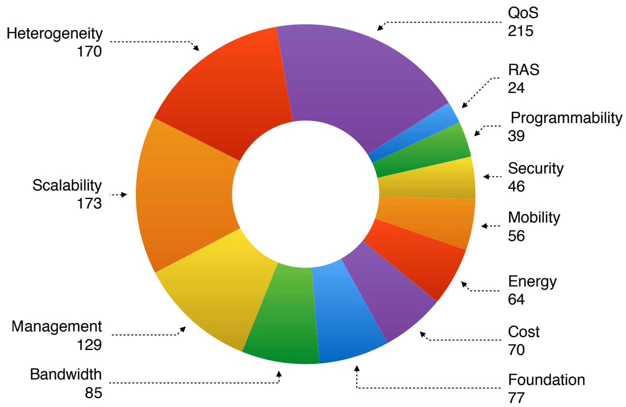


**Figure 11: A pie chart showing number of research articles in this survey paper that address/support a particular objective/feature.**

mechanism for IoT receivers to verify the correctness and integrity of the offloaded tasks.

*5.0.16 PaaS for Fog Computing.* There are not any solid implementations of a PaaS for fog systems, where developers can easily develop software across fog, IoT, and cloud. [132, 401] are some of the very few efforts in this direction. Developing a PaaS for fog computing can ease the general development and acceptability of fog computing. The future PaaS for fog computing should hide the fog configuration specification (e.g., location of fog nodes, their interconnection, their capacity) for users, provision applications, and services proactively and automatically with minimal effort from developers, and support different communication- and application-level protocols and APIs. Once this PaaS is available, its modular design could also be taken into account so that various plugins for different fog computing applications and services could be easily integrated with the PaaS.

*5.0.17 Standardizing Fog Computing.* Different research teams are proposing many independent definitions of fog (and fog-related computing paradigms, such as edge computing). In Section 3 we saw that the definitions of fog computing and its related computing paradigms are not completely standardized. We believe there is a research gap in the definitions and standards for fog computing and other fog-related computing paradigms that needs to be filled by standards and universally-agreed definitions. Once the definitions are agreed upon, researchers become more clear when defining problems, and there will be more agreement among researchers and industry about these paradigms. Organizations such as OpenFog Consortium and OpenEdge Computing are already developing standards and definitions for fog computing and edge computing.

*5.0.18 Hardware Technologies for Fog.* Most studies in the area of fog computing or edge computing do not make use of new hardware or communication technologies, such as non-volatile storage technologies, optical networks, fiber-wireless (FiWi), or FPGAs. Use of new hardware and communication technologies for the design of fog networks, (e.g., fog-to-cloud interconnection) is a direction worth exploring.

## 6 CONCLUSION

The Internet of Things accelerates digital transformation and provides benefits to many industries, including manufacturing, energy, transportation, smart cities, education, retail, healthcare, and government. Due to IoT's fundamental benefits, the number of connected devices and the IoT networks is on the rise, as individuals and companies deploy more and more IoT devices. IoT is expected to connect billions of devices and humans to bring promising advantages for us. Fog computing is one of the promising solutions for handling the big data that is being produced by the IoT, which is often security-critical and time-sensitive. In this paper, we provided a tutorial on what fog computing is and how it relates to or differs from other computing paradigms, such as cloudlets, MEC, and edge computing. Next, we provided a taxonomy of research topics in fog computing and summarized the relevant papers on fog computing and its related computing paradigms. Finally, we provided challenges and future directions for research in fog computing.

## REFERENCES

[1] John Gantz and David Reinsel. Extracting value from chaos. *IDC iview*, 1142 (2011):1–12, 2011.
[2] Andrew McAfee, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. Big data: the management revolution. *Harvard business review*, 90(10): 60–68, 2012.
[3] Dave Evans. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011):1–11, 2011.
[4] Babak Ravandi and Ioannis Papapanagiotou. A self-learning scheduling in cloud software defined block storage. In *Cloud Computing (CLOUD), 2017 IEEE 10th International Conference on*, pages 415–422. IEEE, 2017.
[5] Ben Zhang, Nitesh Mor, John Kolb, Douglas S Chan, Ken Lutz, Eric Allman, John Wawrzynek, Edward A Lee, and John Kubiatowicz. The cloud is not enough: Saving iot from the cloud. In *HotStorage*, 2015.
[6] OpenFogConsortium. Openfog reference architecture for fog computing, 2017. [Online]. Available: https://www.openfogconsortium.org/ra/, February 2017.
[7] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.
[8] Carla Mouradian, Diala Naboulsi, Sami Yangui, Roch H Glitho, Monique J Morrow, and Paul A Polakos. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials*, 20 (1):416–464, Firstquarter 2018.
[9] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Communications Surveys & Tutorials*, 19(4):2359–2391, 2017.
[10] Mithun Mukherjee, Lei Shu, and Di Wang. Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys & Tutorials*, 20(3):1826–1857, 2018.
[11] Charith Perera, Yongrui Qin, Julio C Estrella, Stephan Reiff-Marganiec, and Athanasios V Vasilakos. Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)*, 50(3):32:1–32:43, June 2017. ISSN 0360-0300.
[12] Chao Li, Yushu Xue, Jing Wang, Weigong Zhang, and Tao Li. Edge-oriented computing paradigms: A survey on architecture design and system management. *ACM Computing Surveys (CSUR)*, 51(2):39:1–39:34, April 2018. ISSN 0360-0300.
[13] Jianbing Ni, Kuan Zhang, Xiaodong Lin, and Xuemin Shen. Securing fog computing for internet of things applications: Challenges and solutions. *IEEE Communications Surveys & Tutorials*, 20(1):601–628, Firstquarter 2018.
[14] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3):1657–1681, 2017.
[15] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.
[16] Peter Mell, Tim Grance, et al. The nist definition of cloud computing, 2011.
[17] Luis M Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008.

[18] Tharam Dillon, Chen Wu, and Elizabeth Chang. Cloud computing: issues and challenges. In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pages 27–33. IEEE, 2010.

[19] Borja Sotomayor, Rubén S Montero, Ignacio M Llorente, and Ian Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet computing*, 13(5), 2009.

[20] Joydeep Acharya and Sudhanshu Gaur. Edge compression of gps data for mobile iot. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[21] Tao Zhang. Fog computing brings new business opportunities and disruptions. [Online]. Available: http://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Fog-computing-brings-new-business-opportunities-and-disruptions, Blog, TechTarget.

[22] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[23] Fatemeh Jalali, Kerry Hinton, Robert Ayre, Tansu Alpcan, and Rodney S Tucker. Fog computing may help to save energy in cloud computing. *IEEE Journal on Selected Areas in Communications*, 34(5):1728–1739, 2016.

[24] Zhenyu Wen, Renyu Yang, Peter Garraghan, Tao Lin, Jie Xu, and Michael Rovatsos. Fog orchestration for internet of things services. *IEEE Internet Computing*, 21(2):16–24, 2017.

[25] Shao-Chou Hung, Hsiang Hsu, Shao-Yu Lien, and Kwang-Cheng Chen. Architecture harmonization between cloud radio access networks and fog networks. *IEEE Access*, 3:3019–3034, 2015.

[26] Aleksandra Checko, Henrik L Christiansen, Ying Yan, Lara Scolari, Georgios Kardaras, Michael S Berger, and Lars Dittmann. Cloud ran for mobile networksâĂŤa technology overview. *IEEE Communications surveys & tutorials*, 17(1): 405–426, 2015.

[27] Mugen Peng, Shi Yan, Kecheng Zhang, and Chonggang Wang. Fog-computing-based radio access networks: issues and challenges. *IEEE Network*, 30(4):46–53, 2016.

[28] George H. Forman and John Zahorjan. The challenges of mobile computing. *Computer*, 27(4):38–47, 1994.

[29] Mahadev Satyanarayanan. Fundamental challenges in mobile computing. In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, pages 1–7. Acm, 1996.

[30] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.

[31] NIST. Mobile cloud computing. [Online]. Available: https://www.nist.gov/programs-projects/mobile-cloud-computing, Technical Report, National Institute of Standards and Technology.

[32] Ju Ren, Yaoxue Zhang, Kuan Zhang, and Xuemin Shen. Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions. *IEEE Communications Magazine*, 53(3):98–105, 2015.

[33] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya. Heterogeneity in mobile cloud computing: taxonomy and open challenges. *IEEE Communications Surveys & Tutorials*, 16(1):369–392, 2014.

[34] Muhammad Shiraz, Abdullah Gani, Rashid Hafeez Khokhar, and Rajkumar Buyya. A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Communications Surveys & Tutorials*, 15(3):1294–1313, 2013.

[35] Giulio Grassi, Kyle Jamieson, Paramvir Bahl, and Giovanni Pau. Parkmaster: An in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 16. ACM, 2017.

[36] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, Ma Yunsheng, Songqing Chen, and Peng Hou. A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. *IEEE Transactions on Services Computing*, 11(2):249–261, 2018.

[37] J-P Hubaux, Thomas Gross, J-Y Le Boudec, and Martin Vetterli. Toward self-organized mobile ad hoc networks: the terminodes project. *IEEE Communications Magazine*, 39(1):118–124, 2001.

[38] Utsav Drolia, Rolando Martins, Jiaqi Tan, Ankit Chheda, Monil Sanghavi, Rajeev Gandhi, and Priya Narasimhan. The case for mobile edge-clouds. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, pages 209–215. IEEE, 2013.

[39] Gonzalo Huerta-Canepa and Dongman Lee. A virtual cloud computing provider for mobile devices. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, page 6. ACM, 2010.

[40] Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless networks*, 8(2-3):153–167, 2002.

[41] OpenEdgeConsortium. About - the who, what, and how. [Online]. Available: http://openedgecomputing.org/about.html, Technical Report, OpenEdge Computing.

[42] Andrea Reale. A guide to edge iot analytics. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/edge-iot-analytics, Blog, International Business Machines.

[43] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. Edge-centric computing: Vision and challenges. *ACM SIGCOMM Computer Communication Review*, 45(5):37–42, 2015.

[44] Mung Chiang, Sangtae Ha, I Chih-Lin, Fulvio Risso, and Tao Zhang. Clarifying fog computing and networking: 10 questions and answers. *IEEE Communications Magazine*, 55(4):18–20, 2017.

[45] What is edge computing? [Online]. Available: https://www.ge.com/digital/blog/what-edge-computing, Blog, General Electric.

[46] Fabio Giust, Gianluca Verin, Kiril Antevski, Chou Joey, Yonggang Fang, Walter Featherstone, Francisco Fontes, et al. Mec deployments in 4g and evolution towards 5g. *ETSI White Paper*, 24:1–24, 2018.

[47] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computingâĂŤa key technology towards 5g. *ETSI white paper*, 11 (11):1–16, 2015.

[48] Krishna P Kadiyala and Jorge A Cobb. Inter-as traffic engineering with sdn. In *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*, pages 1–7. IEEE, 2017.

[49] Behzad Mirkhanzadeh, Ali Shakeri, Chencheng Shao, Miguel Razo, Marco Tacca, Gabriele Maria Galimberti, Giovanni Martinelli, Marco Cardani, and Andrea Fumagalli. An sdn-enabled multi-layer protection and restoration mechanism. *Optical Switching and Networking*, 2018.

[50] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4), 2009.

[51] Pengzhan Hao, Yongshu Bai, Xin Zhang, and Yifan Zhang. Edgecourier: an edge-hosted personal service for low-bandwidth document synchronization in mobile cloud storage services. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 7. ACM, 2017.

[52] Yujin Li and Wenye Wang. Can mobile cloudlets support mobile applications? In *Infocom, 2014 proceedings ieee*, pages 1060–1068. IEEE, 2014.

[53] Yaser Jararweh, Loai Tawalbeh, Fadi Ababneh, and Fahd Dosari. Resource efficient mobile computing using cloudlet infrastructure. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*, pages 373–377. IEEE, 2013.

[54] Yong Cui, Jian Song, Kui Ren, Minming Li, Zongpeng Li, Qingmei Ren, and Yangjun Zhang. Software defined cooperative offloading for mobile cloudlets. *IEEE/ACM Transactions on Networking*, 25(3):1746–1760, 2017.

[55] Alex Davies. Cisco pushes iot analytics to the extreme edge with mist computing. [Online]. Available: http://rethinkresearch.biz/articles/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2, Blog, Rethink Research.

[56] Jürgo S Preden, Kalle Tammemäe, Axel Jantsch, Mairo Leier, Andri Riid, and Emine Calis. The benefits of self-awareness and attention in fog and mist computing. *Computer*, 48(7):37–45, 2015.

[57] Pedro M Pinto Silva, Joao Rodrigues, Joaquim Silva, Rolando Martins, Luís Lopes, and Fernando Silva. Using edge-clouds to reduce load on traditional wifi infrastructures and improve quality of experience. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 61–67. IEEE, 2017.

[58] Ahmed Salem and Tamer Nadeem. Lamen: leveraging resources on anonymous mobile edge nodes. In *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop*, pages 15–17. ACM, 2016.

[59] Roberto Morabito. Virtualization on internet of things edge devices with container technologies: a performance evaluation. *IEEE Access*, 5:8835–8850, 2017.

[60] V Bahl. The emergence of micro datacenters (cloudlets) for mobile computing, 2015.

[61] Sherif Abdelwahab, Bechir Hamdaoui, Mohsen Guizani, and Taieb Znati. Cloud of things for sensing-as-a-service: Architecture, algorithms, and use case. *IEEE Internet of Things Journal*, 3(6):1099–1112, 2016.

[62] Minsung Jang, Karsten Schwan, Ketan Bhardwaj, Ada Gavrilovska, and Adhyas Avasthi. Personal clouds: Sharing and integrating networked resources to enhance end user experiences. In *INFOCOM, 2014 Proceedings IEEE*, pages 2220–2228. IEEE, 2014.

[63] Arjuna Sathiaseelan, Adisorn Lertsinsrubtavee, Adarsh Jagan, Prakash Baskaran, and Jon Crowcroft. Cloudrone: Micro clouds in the sky. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 41–44. ACM, 2016.

[64] Karim Habak, Ellen W Zegura, Mostafa Ammar, and Khaled A Harras. Workload management for dynamic mobile device clusters in edge femtoclouds. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 6. ACM, 2017.

[65] Hyunseok Chang, Adiseshu Hari, Sarit Mukherjee, and TV Lakshman. Bringing the cloud to the edge. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 346–351. IEEE, 2014.

[66] Ketan Bhardwaj, Ming-Wei Shih, Pragya Agarwal, Ada Gavrilovska, Taesoo Kim, and Karsten Schwan. Fast, scalable and secure onloading of edge functions using airbox. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 14–27. IEEE, 2016.

[67] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, and Rajiv Ranjan. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing*, 3(6):76–83, 2016.

[68] Ahsan Morshed, Prem Prakash Jayaraman, Timos Sellis, Dimitrios Georgakopoulos, Massimo Villari, and Rajiv Ranjan. Deep osmosis: Holistic distributed deep learning in osmotic computing. *IEEE Cloud Computing*, 4(6):22–32, 2018.

[69] IEEE Standard Association. Ieee 1934-2018 - ieee standard for adoption of openfog reference architecture for fog computing. [Online]. Available: https://standards.ieee.org/standard/1934-2018.html.

[70] Eva Marín-Tordera, Xavi Masip-Bruin, Jordi García-Almiñana, Admela Jukan, Guang-Jie Ren, and Jiafeng Zhu. Do we all really know what a fog node is? current trends towards an open definition. *Computer Communications*, 109: 117–130, 2017.

[71] Luis M Vaquero and Luis Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, 44(5):27–32, 2014.

[72] Subhadeep Sarkar and Sudip Misra. Theoretical modelling of fog computing: a green computing paradigm to support iot applications. *Iet Networks*, 5(2):23–29, 2016.

[73] Alan Sill. Standards at the edge of the cloud. *IEEE Cloud Computing*, 4(2):63–67, 2017.

[74] Judy C Guevara, Luiz F Bittencourt, and Nelson LS da Fonseca. Class of service in fog computing. In *Communications (LATINCOM), 2017 IEEE 9th Latin-American Conference on*, pages 1–6. IEEE, 2017.

[75] Pietro Manzoni, Enrique Hernández-Orallo, Carlos T Calafate, and Juan-Carlos Cano. A proposal for a publish/subscribe, disruption tolerant content island for fog computing. In *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*, pages 47–52. ACM, 2017.

[76] Albert Jonathan, Muhammed Uluyol, Abhishek Chandra, and Jon Weissman. Ensuring reliability in geo-distributed edge cloud. In *Resilience Week (RWS), 2017*, pages 127–132. IEEE, 2017.

[77] Joe Weinman. The 10 laws of fogonomics. *IEEE Cloud Computing*, 4(6):8–14, 2018.

[78] Guenter Klas. Edge computing and the role of cellular networks. *Computer*, 50 (10):40–49, 2017.

[79] Ejaz Ahmed, Arif Ahmed, Ibrar Yaqoob, Junaid Shuja, Abdullah Gani, Muhammad Imran, and Muhammad Shoaib. Bringing computation closer toward the user network: Is edge computing the solution? *IEEE Communications Magazine*, 55(11):138–144, 2017.

[80] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[81] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, Feb 2018.

[82] Ivan Stojmenovic and Sheng Wen. The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1–8. IEEE, 2014.

[83] Longxiang Gao, Tom H Luan, Bo Liu, Wanlei Zhou, and Shui Yu. Fog computing and its applications in 5g. In *5G Mobile Communications*, pages 571–593. Springer, 2017.

[84] Mung Chiang and Tao Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, 2016.

[85] Prateeksha Varshney and Yogesh Simmhan. Demystifying fog computing: Characterizing architectures, applications and abstractions. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 115–124. IEEE, 2017.

[86] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, and Tie Qiu. Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*, 2017.

[87] Paolo Bellavista, Luca Foschini, and Domenico Scotece. Converging mobile edge computing, fog computing, and iot quality requirements. In *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 313–320. IEEE, 2017.

[88] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue. On reducing iot service delay via fog offloading. *IEEE Internet of Things Journal*, 5(2):998–1010, April 2018.

[89] Subhadeep Sarkar, Subarna Chatterjee, and Sudip Misra. Assessment of the suitability of fog computing in the context of internet of things. *IEEE Transactions on Cloud Computing*, 6(1):46–59, 2018.

[90] Xinchen Lyu, Hui Tian, Li Jiang, Alexey Vinel, Sabita Maharjan, Stein Gjessing, and Yan Zhang. Selective offloading in mobile edge computing for the green internet of things. *IEEE Network*, 32(1):54–60, 2018.

[91] Xavi Masip-Bruin, Eva Marín-Tordera, Ghazal Tashakor, Admela Jukan, and Guang-Jie Ren. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications*, 23(5):120–128, 2016.

[92] Liang Tong, Yong Li, and Wei Gao. A hierarchical edge cloud architecture for mobile computing. In *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2016.

[93] Evangelos K Markakis, Kimon Karras, Anargyros Sideris, George Alexiou, and Evangelos Pallis. Computing, caching, and communication at the edge: The cornerstone for building a versatile 5g ecosystem. *IEEE Communications Magazine*, 55(11):152–157, 2017.

[94] Evangelos K Markakis, Kimon Karras, Nikolaos Zotos, Anargyros Sideris, Theoharris Moysiadis, Angelo Corsaro, George Alexiou, Charalabos Skianis, George Mastorakis, Constandinos X Mavromoustakis, et al. Exegesis: Extreme edge resource harvesting for a virtualized fog environment. *IEEE Communications Magazine*, 55(7):173–179, 2017.

[95] Chii Chang, Satish Narayana Srirama, and Rajkumar Buyya. Indie fog: An efficient fog-computing infrastructure for the internet of things. *Computer*, 50 (9):92–98, 2017.

[96] Charles C Byers. Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled iot networks. *IEEE Communications Magazine*, 55(8):14–20, 2017.

[97] Marcelo Yannuzzi, R Milito, René Serral-Gracià, D Montero, and Mario Nemirovsky. Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on*, pages 325–329. IEEE, 2014.

[98] Zijiang Hao, Ed Novak, Shanhe Yi, and Qun Li. Challenges and software architecture for fog computing. *IEEE Internet Computing*, 21(2):44–53, 2017.

[99] Shanhe Yi, Zhengrui Qin, and Qun Li. Security and privacy issues of fog computing: A survey. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 685–695. Springer, 2015.

[100] Dennis Grewe, Marco Wagner, Mayutan Arumaithurai, Ioannis Psaras, and Dirk Kutscher. Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions. In *Proceedings of the Workshop on Mobile Edge Communications*, pages 7–12. ACM, 2017.

[101] Charith Perera, Yongrui Qin, Julio C Estrella, Stephan Reiff-Marganiec, and Athanasios V Vasilakos. Fog computing for sustainable smart cities: A survey. *ACM Computing Surveys (CSUR)*, 50(3):32, 2017.

[102] Jianbing Ni, Aiqing Zhang, Xiaodong Lin, and Xuemin Sherman Shen. Security, privacy, and fairness in fog-based vehicular crowdsensing. *IEEE Communications Magazine*, 55(6):146–152, 2017.

[103] Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. Fog computing: A taxonomy, survey and future directions. In *Internet of Everything*, pages 103–130. Springer, 2018.

[104] Peng Zhang, Joseph K Liu, F Richard Yu, Mehdi Sookhak, Man Ho Au, and Xiapu Luo. A survey on access control in fog computing. *IEEE Communications Magazine*, 56(2):144–149, 2018.

[105] Syed Noorulhassan Shirazi, Antonios Gouglidis, Arsham Farshad, and David Hutchison. The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective. *IEEE Journal on Selected Areas in Communications*, 35(11):2586–2595, 2017.

[106] Xiang Sun and Nirwan Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12):22–29, 2016.

[107] Antonio Brogi and Stefano Forti. Qos-aware deployment of iot applications through the fog. *IEEE Internet of Things Journal*, 4(5):1185–1192, 2017.

[108] Clinton Dsouza, Gail-Joon Ahn, and Marthony Taguinod. Policy-driven security management for fog computing: Preliminary framework and a case study. In *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, pages 16–23. IEEE, 2014.

[109] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing: Platform and applications. In *Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on*, pages 73–78. IEEE, 2015.

[110] Abbas Kiani and Nirwan Ansari. Toward hierarchical mobile edge computing: An auction-based profit maximization approach. *IEEE Internet of Things Journal*, 4(6):2082–2091, 2017.

[111] Yongbo Li, Yurong Chen, Tian Lan, and Guru Venkataramani. Mobiqor: Pushing the envelope of mobile edge computing via quality-of-result optimization. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 1261–1270. IEEE, 2017.

[112] Ivan Farris, Roberto Girau, Leonardo Militano, Michele Nitti, Luigi Atzori, Antonio Iera, and Giacomo Morabito. Social virtual objects in the edge cloud. *IEEE Cloud Computing*, 2(6):20–28, 2015.

[113] Abhinandan S Prasad, Mayutan Arumaithurai, David Koll, and Xiaoming Fu. Raera: A robust auctioning approach for edge resource allocation. In *Proceedings of the Workshop on Mobile Edge Communications*, pages 49–54. ACM, 2017.

[114] Sz Varadi, G Gultekin Varkonyi, and A Kertesz. Law and iot: How to see things clearly in the fog. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 233–238. IEEE, 2018.

[115] M Yannuzzi, R Irons-Mclean, F van Lingen, S Raghav, A Somaraju, C Byers, T Zhang, A Jain, J Curado, D Carrera, et al. Toward a converged openfog and etsi mano architecture. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–6. IEEE,

2017.

[116] Yotam Harchol, Aisha Mushtaq, James McCauley, Aurojit Panda, and Scott Shenker. Cessna: Resilient edge-computing. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pages 1–6. ACM, 2018.

[117] Shizhe Zang, Wei Bao, Phee Lep Yeoh, Branka Vucetic, and Yonghui Li. Paying less for more? combo plans for edge-computing services. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[118] Ioannis Psaras, Onur Ascigil, Sergi Rene, George Pavlou, Alex Afanasyev, and Lixia Zhang. Mobile data repositories at the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[119] Daewoo Kim, Hyojung Lee, Hyungseok Song, Nakjung Choi, and Yung Yi. On the economics of fog computing: Inter-play among infrastructure and service providers, users, and edge resource owners. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[120] Ronan-Alexandre Cherrueau, Adrien Lebre, Dimitri Pertin, Fetahi Wuhib, and João Monteiro Soares. Edge computing resource management system: a critical building block! initiating the debate via openstack. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[121] Meenakshi Syamkumar, Paul Barford, and Ramakrishnan Durairajan. Deployment characteristics of "the edge" in mobile edge computing. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, MECOMM'18, pages 43–49. ACM, 2018.

[122] Chia-Yu Chang, Konstantinos Alexandris, Navid Nikaein, Kostas Katsalis, and Thrasyvoulos Spyropoulos. Mec architectural implications for lte/lte-a networks. In *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*, pages 13–18. ACM, 2016.

[123] Sherif Abdelwahab, Sophia Zhang, Ashley Greenacre, Kai Ovesen, Kevin Bergman, and Bechir Hamdaoui. When clones flock near the fog. *IEEE Internet of Things Journal*, 2018.

[124] Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H Luan, and Hao Liang. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, 3(6):1171–1181, 2016.

[125] Cesare Alippi, Romano Fantacci, Dania Marabissi, and Manuel Roveri. A cloud to the ground: The new frontier of intelligent and autonomous networks of things. *IEEE Communications Magazine*, 54(12):14–20, 2016.

[126] Olena Skarlat, Matteo Nardelli, Stefan Schulte, Michael Borkowski, and Philipp Leitner. Optimized iot service placement in the fog. *Service Oriented Computing and Applications*, 11(4):427–443, 2017.

[127] Slavica Tomovic, Kenji Yoshigoe, Ivo Maljevic, and Igor Radusinovic. Software-defined fog network architecture for iot. *Wireless Personal Communications*, 92 (1):181–196, 2017.

[128] Yu-Jen Ku, Dian-Yu Lin, Chia-Fu Lee, Ping-Jung Hsieh, Hung-Yu Wei, Chun-Ting Chou, and Ai-Chun Pang. 5g radio access network design with the fog paradigm: Confluence of communications and computing. *IEEE Communications Magazine*, 55(4):46–52, 2017.

[129] Rafael Moreno-Vozmediano, Ruben S Montero, Eduardo Huedo, and Ignacio M Llorente. Cross-site virtual network in cloud and fog computing. *IEEE Cloud Computing*, 4(2):46–53, 2017.

[130] Kai Liang, Liqiang Zhao, Xiaoli Chu, and Hsiao-Hwa Chen. An integrated architecture for software defined and virtualized radio access networks with fog computing. *IEEE Network*, 31(1):80–87, 2017.

[131] Andreas Kapsalis, Panagiotis Kasnesis, Iakovos S Venieris, Dimitra I Kaklamani, and Charalampos Z Patrikakis. A cooperative fog approach for effective workload balancing. *IEEE Cloud Computing*, 4(2):36–45, 2017.

[132] Sami Yangui, Pradeep Ravindran, Ons Bibani, Roch H Glitho, Nejib Ben Hadj-Alouane, Monique J Morrow, and Paul A Polakos. A platform as-a-service for hybrid cloud/fog environments. In *Local and Metropolitan Area Networks (LANMAN), 2016 IEEE International Symposium on*, pages 1–7. IEEE, 2016.

[133] Jianhua Li, Jiong Jin, Dong Yuan, Marimuthu Palaniswami, and Klaus Moessner. Ehopes: Data-centered fog platform for smart living. In *Telecommunication Networks and Applications Conference (ITNAC), 2015 International*, pages 308–313. IEEE, 2015.

[134] Frank van Lingen, Marcelo Yannuzzi, Anuj Jain, Rik Irons-Mclean, Oriol Lluch, David Carrera, Juan Luis Perez, Alberto Gutierrez, Diego Montero, Josep Marti, et al. The unavoidable convergence of nfv, 5g, and fog: A model-driven approach to bridge cloud and edge. *IEEE Communications Magazine*, 55(8):28–35, 2017.

[135] Ricard Vilalta, Victor López, Alessio Giorgetti, Shuping Peng, Vittorio Orsini, Luis Velasco, Rene Serral-Gracia, Donal Morris, Silvia De Fina, Filippo Cugini, et al. Telcofog: A unified flexible fog and cloud computing architecture for 5g networks. *IEEE Communications Magazine*, 55(8):36–43, 2017.

[136] Huaqing Zhang, Yanru Zhang, Yunan Gu, Dusit Niyato, and Zhu Han. A hierarchical game framework for resource management in fog computing. *IEEE Communications Magazine*, 55(8):52–57, 2017.

[137] Piotr Borylo, Artur Lason, Jacek Rzasa, Andrzej Szymanski, and Andrzej Jajszczyk. Energy-aware fog and cloud interplay supported by wide area software defined networking. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7. IEEE, 2016.

[138] Brian Amento, Robert J Hall, Kaustubh Joshi, and K Hal Purdy. Focusstack: Orchestrating edge clouds using focus of attention. *IEEE Internet Computing*, 21 (1):56–61, 2017.

[139] Marcelo Yannuzzi, Frank van Lingen, Anuj Jain, Oriol Lluch Parellada, Manel Mendoza Flores, David Carrera, Juan Luis Pérez, Diego Montero, Pablo Chacin, Angelo Corsaro, et al. A new era for cities with fog computing. *IEEE Internet Computing*, 21(2):54–67, 2017.

[140] Yucen Nan, Wei Li, Wei Bao, Flavia C Delicato, Paulo F Pires, and Albert Y Zomaya. A dynamic tradeoff data processing framework for delay-sensitive applications in cloud of things systems. *Journal of Parallel and Distributed Computing*, 112:53–66, 2018.

[141] Xiaokang Wang, Laurence T Yang, Xia Xie, Jirong Jin, and M Jamal Deen. A cloud-edge computing framework for cyber-physical-social services. *IEEE Communications Magazine*, 55(11):80–85, 2017.

[142] Massimo Ficco, Christian Esposito, Yang Xiang, and Francesco Palmieri. Pseudo-dynamic testing of realistic edge-fog cloud ecosystems. *IEEE Communications Magazine*, 55(11):98–104, 2017.

[143] Jianhua Li, Jiong Jin, Dong Yuan, and Hongke Zhang. Virtual fog: A virtualization enabled fog computing framework for internet of things. *IEEE Internet of Things Journal*, 5(1):121–131, 2018.

[144] T. Yu, X. Wang, and A. Shami. A novel fog computing enabled temporal data reduction scheme in iot systems. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–5, Dec 2017.

[145] Wenyu Zhang, Zhenjiang Zhang, and Han-Chieh Chao. Cooperative fog computing for dealing with big data in the internet of vehicles: Architecture and hierarchical resource management. *IEEE Communications Magazine*, 55(12): 60–67, 2017.

[146] H. Zhang, Y. Qiu, X. Chu, K. Long, and V. C. M. Leung. Fog radio access networks: Mobility management, interference mitigation, and resource optimization. *IEEE Wireless Communications*, 24(6):120–127, Dec 2017.

[147] B. Lorenzo, J. Garcia-Rois, X. Li, J. Gonzalez-Castano, and Y. Fang. A robust dynamic edge network architecture for the internet of things. *IEEE Network*, 32 (1):8–15, 2018.

[148] Takuo Suganuma, Takuma Oide, Shinji Kitagami, Kenji Sugawara, and Norio Shiratori. Multiagent-based flexible edge computing architecture for iot. *IEEE Network*, 32(1):16–23, 2018.

[149] Quan Yuan, Haibo Zhou, Jinglin Li, Zhihan Liu, Fangchun Yang, and Xuemin Sherman Shen. Toward efficient content delivery for automated driving services: An edge computing solution. *IEEE Network*, 32(1):80–86, 2018.

[150] Adrien Lebre, Jonathan Pastor, Anthony Simonet, and Frédéric Desprez. Revising openstack to operate fog/edge computing infrastructures. In *Cloud Engineering (IC2E), 2017 IEEE International Conference on*, pages 138–148. IEEE, 2017.

[151] Mohamed Abderrahim, Meryem Ouzzif, Karine Guillouard, Jerome Francois, and Adrien Lèbre. A holistic monitoring service for fog/edge infrastructures: a foresight study. In *The IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud 2017)*, 2017.

[152] Cesar A Garcia-Perez and Pedro Merino. Enabling low latency services on lte networks. In *Foundations and Applications of Self* Systems, IEEE International Workshops on*, pages 248–255. IEEE, 2016.

[153] Minh Le, Zheng Song, Young-Woo Kwon, and Eli Tilevich. Reliable and efficient mobile edge computing in highly dynamic and volatile environments. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 113–120. IEEE, 2017.

[154] Ivan Lujic, Vincenzo De Maio, and Ivona Brandic. Efficient edge storage management based on near real-time forecasts. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 21–30. IEEE, 2017.

[155] William Tärneberg, Alessandro Vittorio Papadopoulos, Amardeep Mehta, Johan Tordsson, and Maria Kihl. Distributed approach to the holistic resource management of a mobile cloud network. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 51–60. IEEE, 2017.

[156] Shashank Shekhar, Ajay Dev Chhokra, Anirban Bhattacharjee, Guillaume Aupy, and Aniruddha Gokhale. Indices: exploiting edge resources for performance-aware cloud-hosted services. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 75–80. IEEE, 2017.

[157] Mohammed Islam Naas, Philippe Raipin Parvedy, Jalil Boukhobza, and Laurent Lemarchand. ifogstor: an iot data placement strategy for fog infrastructure. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 97–104. IEEE, 2017.

[158] Peng Yang, Ning Zhang, Yuanguo Bi, Li Yu, and Xuemin Sherman Shen. Catalyzing cloud-fog interoperation in 5g wireless networks: An sdn approach. *IEEE Network*, 31(5):14–20, 2017.

[159] Bo Tang, Zhen Chen, Gerald Hefferman, Tao Wei, Haibo He, and Qing Yang. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015*, page 28. ACM, 2015.

[160] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang. Multitier fog computing with large-scale iot data analytics for smart cities. *IEEE Internet of Things Journal*, 5(2):677–686, April 2018.

[161] Sharief MA Oteafy and Hossam S Hassanein. Iot in the fog: A roadmap for data-centric iot development. *IEEE Communications Magazine*, 56(3):157–163, 2018.

[162] Jianhua Li, Jiong Jin, Dong Yuan, and Hongke Zhang. Virtual fog: A virtualization enabled fog computing framework for internet of things. *IEEE Internet of Things Journal*, 5(1):121–131, 2018.

[163] Deepak Puthal, Mohammad S Obaidat, Priyadarsi Nanda, Mukesh Prasad, Saraju P Mohanty, and Albert Y Zomaya. Secure and sustainable load balancing of edge data centers in fog computing. *IEEE Communications Magazine*, 56(5):60–65, 2018.

[164] Michael Lander Raagaard, Paul Pop, Marina Gutiérrez, and Wilfried Steiner. Runtime reconfiguration of time-sensitive networking (tsn) schedules for fog computing. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[165] Claus Pahl, Nabil El Ioini, Sven Helmer, and Brian Lee. An architecture pattern for trusted orchestration in iot edge clouds. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 63–70. IEEE, 2018.

[166] Nitinder Mohan, Aleksandr Zavodovski, Pengyuan Zhou, and Jussi Kangasharju. Anveshak: Placing edge servers in the wild. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pages 7–12. ACM, 2018.

[167] A Silvestro, N Mohan, J Kangasharju, F Schneider, and X Fu. Mute: Multi-tier edge networks. In *Proceedings of the 5th Workshop on CrossCloud Infrastructures & Platforms*, page 1. ACM, 2018.

[168] Sahil Garg, Amritpal Singh, Kujeet Kaur, Shalini Batra, Neeraj Kumar, and Mohammad S Obaidat. Edge-based content delivery for providing qoe in wireless networks using quotient filter. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[169] Christopher Meiklejohn, Heather Miller, and Zeeshan Lakhani. Towards a solution to the red wedding problem. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[170] Chi-Yu Li, Hsueh-Yang Liu, Po-Hao Huang, Hsu-Tung Chien, Guan-Hua Tu, Pei-Yuan Hong, and Ying-Dar Lin. Mobile edge computing platform deployment in 4g LTE networks: A middlebox approach. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[171] Simone Cirani, Gianluigi Ferrari, Nicola Iotti, and Marco Picone. The iot hub: a fog node for seamless management of heterogeneous connected smart objects. In *Sensing, Communication, and Networking-Workshops (SECON Workshops), 2015 12th Annual IEEE International Conference on*, pages 1–6. IEEE, 2015.

[172] Xueshi Hou, Yong Li, Min Chen, Di Wu, Depeng Jin, and Sheng Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, 2016.

[173] Cássio Prazeres and Martin Serrano. Soft-iot: Self-organizing fog of things. In *Advanced Information Networking and Applications Workshops (WAINA), 2016 30th International Conference on*, pages 803–808. IEEE, 2016.

[174] Nichoas Kaminski, Irene Macaluso, Emanuele Di Pascale, Avishek Nag, John Brady, Mark Kelly, Keith Nolan, Wael Guibene, and Linda Doyle. A neural-network-based realization of in-network computation for the internet of things. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[175] I Farris, A Orsino, L Militano, Michele Nitti, Giuseppe Araniti, Luigi Atzori, and Antonio Iera. Federations of connected things for delay-sensitive iot services in 5g environments. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[176] Mehdi Sookhak, F Richard Yu, Ying He, Hamid Talebian, Nader Sohrabi Safa, Nan Zhao, Muhammad Khurram Khan, and Neeraj Kumar. Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing. *IEEE Vehicular Technology Magazine*, 12(3):55–64, 2017.

[177] Ming Tao, Kaoru Ota, and Mianxiong Dong. Foud: integrating fog and cloud for 5g-enabled v2g networks. *IEEE Network*, 31(2):8–13, 2017.

[178] Cheng Huang, Rongxing Lu, and Kim-Kwang Raymond Choo. Vehicular fog computing: architecture, use case, and security and forensic challenges. *IEEE Communications Magazine*, 55(11):105–111, 2017.

[179] Ju Ren, Hui Guo, Chugui Xu, and Yaoxue Zhang. Serving at the edge: A scalable iot architecture based on transparent computing. *IEEE Network*, 31(5):96–105, 2017.

[180] Saúl Alonso-Monsalve, Félix García-Carballeira, and Alejandro Calderón. Fog computing through public-resource computing and storage. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 81–87. IEEE, 2017.

[181] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE Network*, 32(1):96–101, 2018.

[182] Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman. Nebula: Distributed edge cloud for data intensive computing. In *Cloud Engineering (IC2E), 2014 IEEE International Conference on*, pages 57–66. IEEE, 2014.

[183] Florian Hagenauer, Christoph Sommer, Takamasa Higuchi, Onur Altintas, and Falko Dressler. Vehicular micro clouds as virtual edge servers for efficient data collection. In *23rd ACM International Conference on Mobile Computing and Networking (MobiCom 2017), 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services (CarSys 2017)*, 2017.

[184] Seyed Hossein Mortazavi, Mohammad Salehe, Carolina Simoes Gomes, Caleb Phillips, and Eyal de Lara. Cloudpath: A multi-tier cloud computing framework. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 20:1–20:13. ACM, 2017.

[185] Paolo Bellavista, Stefano Chessa, Luca Foschini, Leo Gioia, and Michele Girolami. Human-enabled edge computing: Exploiting the crowd as a dynamic extension of mobile edge computing. *IEEE Communications Magazine*, 56(1):145–155, 2018.

[186] Seongah Jeong, Osvaldo Simeone, and Joonhyuk Kang. Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology*, 67(3):2049–2063, 2018.

[187] Omar A Nasr, Yasser Amer, and Mohammed AboBakr. The âĂIJdropletâĂİ: A new personal device to enable fog computing. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 93–99. IEEE, 2018.

[188] Hassnaa Moustafa, Eve M Schooler, and Jessica McCarthy. Reverse cdn in fog computing: The lifecycle of video data in connected and autonomous vehicles. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–5. IEEE, 2017.

[189] Chenhan Xu, Kun Wang, Guoliang Xu, Peng Li, Song Guo, and Jiangtao Luo. Making big data open in collaborative edges: a blockchain-based framework with reduced resource requirements. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[190] Zhenyu Wen, Pramod Bhatotia, Ruichuan Chen, Myungjin Lee, et al. Approxiot: Approximate analytics for edge computing. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 411–421. IEEE, 2018.

[191] Aravindh Raman Nishanth Sastry Arjuna Sathiaseelan and Jigna Chandria Andrew Secker. Wi-stitch: Content delivery in converged edge networks. *ACM SIGCOMM Computer Communication Review*, 47(5), 2017.

[192] Kyle E Benson, Guoxi Wang, Nalini Venkatasubramanian, and Young-Jin Kim. Ride: A resilient iot data exchange middleware leveraging sdn and edge cloud resources. In *Internet-of-Things Design and Implementation (IoTDI), 2018 IEEE/ACM Third International Conference on*, pages 72–83. IEEE, 2018.

[193] Maxweel Carmo, Sandino Jardim, Augusto Neto, Rui Aguiar, Daniel Corujo, and Joel JPC Rodrigues. Slicing wifi wlan-sharing access infrastructures to enhance ultra-dense 5g networking. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[194] Quan Zhang, Xiaohong Zhang, Qingyang Zhang, Weisong Shi, and Hong Zhong. Firework: Big data sharing and processing in collaborative edge environment. In *Hot Topics in Web Systems and Technologies (HotWeb), 2016 Fourth IEEE Workshop on*, pages 20–25. IEEE, 2016.

[195] Karthik Dantu, Steven Y Ko, and Lukasz Ziarek. Raina: Reliability and adaptability in android for fog computing. *IEEE Communications Magazine*, 55(4): 41–45, 2017.

[196] Kuljeet Kaur, Tanya Dhand, Neeraj Kumar, and Sherali Zeadally. Container-as-a-service at the edge: Trade-off between energy efficiency and service availability at fog nano data centers. *IEEE wireless communications*, 24(3):48–56, 2017.

[197] Enrique Saurez, Kirak Hong, Dave Lillethun, Umakishore Ramachandran, and Beate Ottenwälder. Incremental deployment and migration of geo-distributed situation awareness applications in the fog. In *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems*, pages 258–269. ACM, 2016.

[198] Rakesh Jain and Samir Tata. Cloud to edge: distributed deployment of process-aware iot applications. In *Edge Computing (EDGE), 2017 IEEE International Conference on*, pages 182–189. IEEE, 2017.

[199] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. Mobile fog: A programming model for large-scale applications on the internet of things. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pages 15–20. ACM, 2013.

[200] Nakjung Choi, Daewoo Kim, Sung-Ju Lee, and Yung Yi. A fog operating system for user-oriented iot services: Challenges and research directions. *IEEE Communications Magazine*, 55(8):44–51, 2017.

[201] Nam Ky Giang, Michael Blackstock, Rodger Lea, and Victor CM Leung. Developing iot applications in the fog: a distributed dataflow approach. In *Internet of Things (IOT), 2015 5th International Conference on the*, pages 155–162. IEEE, 2015.

[202] Eduard Gibert Renart, Javier Diaz-Montes, and Manish Parashar. Data-driven stream processing at the edge. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 31–40. IEEE, 2017.

[203] Xiang Su, Pingjiang Li, Jukka Riekki, Xiaoli Liu, Jussi Kiljander, Juha-Pekka Soininen, Christian Prehofer, Huber Flores, and Yuhong Li. Adaptive recovery of incomplete datasets for edge analytics. In *Pervasive Computing and Communications (PerCom), 2018 IEEE International Conference on*, pages 1–9. IEEE, 2018.

[204] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa. Fogflow: Easy programming of iot services over cloud and edges for smart cities. *IEEE Internet of Things Journal*, 5(2):696–707, April 2018.

[205] Taeyeol Jeong, Jaeyoon Chung, James Won-Ki Hong, and Sangtae Ha. Towards a distributed computing framework for fog. In *Fog World Congress (FWC), 2017*

*IEEE*, pages 1–6. IEEE, 2017.
[206] Harshit Gupta and Umakishore Ramachandran. Fogstore: A geo-distributed key-value store guaranteeing low latency for strongly consistent access. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pages 148–159. ACM, 2018.
[207] Jianxin Zhao, Tudor Tiplea, Richard Mortier, Jon Crowcroft, and Liang Wang. Data analytics service composition and deployment on edge devices. In *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, Big-DAMA '18, pages 27–32. ACM, 2018.
[208] Shadi A. Noghabi, John Kolb, Peter Bodik, and Eduardo Cuervo. Steel: Simplified development and deployment of edge-cloud applications. In *10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18)*, Boston, MA, 2018. USENIX Association.
[209] Harshit Gupta, Zhuangdi Xu, and Umakishore Ramachandran. Datafog: Towards a holistic data management platform for the iot age at the network edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.
[210] Qiang Fan and Nirwan Ansari. Cost aware cloudlet placement for big data processing at the edge. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[211] Yu Xiao, Marius Noreikis, and Antti Ylä-Jaäiski. Qos-oriented capacity planning for edge computing. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[212] Hongzhi Guo, Jiajia Liu, and Huiling Qin. Collaborative mobile edge computation offloading for iot over fiber-wireless networks. *IEEE Network*, 32(1):66–71, 2018.
[213] Yuan-Yao Shih, Wei-Ho Chung, Ai-Chun Pang, Te-Chuan Chiu, and Hung-Yu Wei. Enabling low-latency applications in fog-radio access networks. *IEEE Network*, 31(1):52–58, 2017.
[214] S. Mondal, G. Das, and E. Wong. A novel cost optimization framework for multi-cloudlet environment over optical access networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–7, Dec 2017.
[215] Carlo Vallati, Antonio Virdis, Enzo Mingozzi, and Giovanni Stea. Exploiting lte d2d communications in m2m fog platforms: Deployment and practical issues. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 585–590. IEEE, 2015.
[216] Amardeep Mehta, William Tärneberg, Cristian Klein, Johan Tordsson, Maria Kihl, and Erik Elmroth. How beneficial are intermediate layer data centers in mobile edge networks? In *Foundations and Applications of Self* Systems, IEEE International Workshops on*, pages 222–229. IEEE, 2016.
[217] Alberto Ceselli, Marco Premoli, and Stefano Secci. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking*, 25(3):1818–1831, 2017.
[218] M. Jia, J. Cao, and W. Liang. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, 5(4):725–737, 2017.
[219] Bhaskar Prasad Rimal, Dung Pham Van, and Martin Maier. Mobile-edge computing versus centralized cloud computing over a converged fiwi access network. *IEEE Transactions on Network and Service Management*, 14(3):498–513, 2017.
[220] Julien Gedeon, Jeff Krisztinkovics, Christian Meurisch, Michael Stein, Lin Wang, and Max Mühlhäuser. A multi-cloudlet infrastructure for future smart cities: An empirical study. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 19–24. ACM, 2018.
[221] Mohammad Aazam and Eui-Nam Huh. Fog computing micro datacenter based dynamic resource estimation and pricing model for iot. In *Advanced Information Networking and Applications (AINA), 2015 IEEE 29th International Conference on*, pages 687–694. IEEE, 2015.
[222] Fatemeh Jalali, Arun Vishwanath, Julian de Hoog, and Frank Suits. Interconnecting fog computing and microgrids for greening iot. In *Innovative Smart Grid Technologies-Asia (ISGT-Asia), 2016 IEEE*, pages 693–698. IEEE, 2016.
[223] Said El Kafhali and Khaled Salah. Efficient and dynamic scaling of fog nodes for iot devices. *The Journal of Supercomputing*, 73(12):5261–5284, 2017.
[224] Francesco Malandrino, Scott Kirkpatrick, and Carla-Fabiana Chiasserini. How close to the edge?: Delay/utilization trends in mec. In *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking*, pages 37–42. ACM, 2016.
[225] Noriaki Kamiyama, Yuusuke Nakano, Kohei Shiomoto, Go Hasegawa, Masayuki Murata, and Hideo Miyahara. Analyzing effect of edge computing on reduction of web response time. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.
[226] Chanh Nguyen Le Tan, Cristian Klein, and Erik Elmroth. Location-aware load prediction in edge data centers. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 25–31. IEEE, 2017.
[227] Duong Tung Nguyen, Long Bao Le, and Vijay Bhargava. Edge computing resource procurement: An online optimization approach. In *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*, pages 807–812. IEEE, 2018.
[228] Cosimo Anglano, Massimo Canonico, Paolo Castagno, Marco Guazzone, and Matteo Sereno. A game-theoretic approach to coalition formation in fog provider federations. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 123–130. IEEE, 2018.
[229] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Profit-aware resource management for edge computing systems. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 25–30. ACM, 2018.
[230] Ming Zeng, Yong Li, Ke Zhang, Muhammad Waqas, and Depeng Jin. Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.
[231] Rahul Urgaonkar, Shiqiang Wang, Ting He, Murtaza Zafer, Kevin Chan, and Kin K Leung. Dynamic service migration and workload scheduling in edge-clouds. *Performance Evaluation*, 91:205–228, 2015.
[232] Wuyang Zhang, Yi Hu, Yanyong Zhang, and Dipankar Raychaudhuri. Segue: Quality of service aware edge cloud service migration. In *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*, pages 344–351. IEEE, 2016.
[233] Emre Yigitoglu, Mohamed Mohamed, Ling Liu, and Heiko Ludwig. Foggy: A framework for continuous automated iot application deployment in fog computing. In *AI & Mobile Services (AIMS), 2017 IEEE International Conference on*, pages 38–45. IEEE, 2017.
[234] Ivan Farris, Tarik Taleb, Miloud Bagaa, and Hannu Flick. Optimizing service replication for mobile delay-sensitive applications in 5g edge network. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[235] Sherif Abdelwahab and Bechir Hamdaoui. Flocking virtual machines in quest for responsive iot cloud services. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[236] Roberto Morabito, Ivan Farris, Antonio Iera, and Tarik Taleb. Evaluating performance of containerized iot services for clustered devices at the network edge. *IEEE Internet of Things Journal*, 4(4):1019–1030, 2017.
[237] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, and Umakishore Ramachandran. Migcep: operator migration for mobility driven distributed complex event processing. In *Proceedings of the 7th ACM international conference on Distributed event-based systems*, pages 183–194. ACM, 2013.
[238] Nam Ky Giang, Michael Blackstock, Rodger Lea, and Victor CM Leung. Developing iot applications in the fog: a distributed dataflow approach. In *Internet of Things (IOT), 2015 5th International Conference on the*, pages 155–162. IEEE, 2015.
[239] Harshit Gupta, Shubha Brata Nath, Sandip Chakraborty, and Soumya K Ghosh. Sdfog: A software defined computing architecture for qos aware service orchestration over edge devices. *arXiv preprint arXiv:1609.01190*, 2016.
[240] Adisorn Lertsinsrubtavee, Anwaar Ali, Carlos Molina-Jimenez, Arjuna Sathiaseelan, and Jon Crowcroft. Picasso: A lightweight edge computing platform. In *Cloud Networking (CloudNet), 2017 IEEE 6th International Conference on*, pages 1–7. IEEE, 2017.
[241] Tiago Gama Rodrigues, Katsuya Suto, Hiroki Nishiyama, and Nei Kato. A pso model with vm migration and transmission power control for low service delay in the multiple cloudlets ecc scenario. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[242] Nan Wang, Blesson Varghese, Michail Matthaiou, and Dimitrios S Nikolopoulos. Enorm: A framework for edge node resource management. *IEEE Transactions on Services Computing*, 2017.
[243] Wei Bao, Dong Yuan, Zhengjie Yang, Shen Wang, Wei Li, Bing Bing Zhou, and Albert Y Zomaya. Follow me fog: Toward seamless handover timing schemes in a fog computing environment. *IEEE Communications Magazine*, 55(11):72–78, 2017.
[244] Lele Ma, Shanhe Yi, and Qun Li. Efficient service handoff across edge servers via docker container migration. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 11. ACM, 2017.
[245] Vittorio Cozzolino, Aaron Yi Ding, and Jörg Ott. Fades: Fine-grained edge offloading with unikernels. In *Proceedings of the Workshop on Hot Topics in Container Networking and Networked Systems*, pages 36–41. ACM, 2017.
[246] Xiang Sun and Nirwan Ansari. Primal: Profit maximization avatar placement for mobile edge computing. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.
[247] Wuyang Zhang, Jiachen Chen, Yanyong Zhang, and Dipankar Raychaudhuri. Towards efficient edge cloud augmentation for virtual reality mmogs. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 8. ACM, 2017.
[248] I Hou, Tao Zhao, Shiqiang Wang, Kevin Chan, et al. Asymptotically optimal algorithm for online reconfiguration of edge-clouds. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 291–300. ACM, 2016.
[249] Lucas Chaufournier, Prateek Sharma, Franck Le, Erich Nahum, Prashant Shenoy, and Don Towsley. Fast transparent virtual machine migration in distributed edge clouds. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 10. ACM, 2017.

[250] Kiryong Ha, Yoshihisa Abe, Thomas Eiszler, Zhuo Chen, Wenlu Hu, Brandon Amos, Rohit Upadhyaya, Padmanabhan Pillai, and Mahadev Satyanarayanan. You can teach elephants to dance: agile vm handoff for edge computing. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 12. ACM, 2017.

[251] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis. Live service migration in mobile edge clouds. *IEEE Wireless Communications*, 25(1):140–147, February 2018.

[252] Balázs Németh, Márk Szalay, János Dóka, Matthias Rost, Stefan Schmid, László Toka, and Balázs Sonkoly. Fast and efficient network service embedding method with adaptive offloading to the edge. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018.

[253] Seyed Hossein Mortazavi, Bharath Balasubramanian, Eyal de Lara, and Shankaranarayanan Puzhavakath Narayanan. Toward session consistency for the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[254] Roberto Bruschi, Franco Davoli, Paolo Lago, and Jane Frances Pajo. Move with me: Scalably keeping virtual objects close to users on the move. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[255] Lin Gu, Deze Zeng, Song Guo, Ahmed Barnawi, and Yong Xiang. Cost efficient resource management in fog computing supported medical cyber-physical system. *IEEE Transactions on Emerging Topics in Computing*, 5(1):108–119, 2017.

[256] VBC Souza, Wilson Ramírez, Xavier Masip-Bruin, Eva Marín-Tordera, G Ren, and Ghazal Tashakor. Handling service allocation in combined fog-cloud scenarios. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–5. IEEE, 2016.

[257] Fatma Ben Jemaa, Guy Pujolle, and Michel Pariente. Qos-aware vnf placement optimization in edge-central carrier cloud architecture. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–7. IEEE, 2016.

[258] Olena Skarlat, Matteo Nardelli, Stefan Schulte, and Schahram Dustdar. Towards qos-aware fog service placement. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 89–96. IEEE, 2017.

[259] Hua-Jun Hong, Pei-Hsuan Tsai, and Cheng-Hsin Hsu. Dynamic module deployment in a fog computing platform. In *Network Operations and Management Symposium (APNOMS), 2016 18th Asia-Pacific*, pages 1–6. IEEE, 2016.

[260] Jinlai Xu, Balaji Palanisamy, Heiko Ludwig, and Qingyang Wang. Zenith: Utility-aware resource allocation for edge computing. In *Edge Computing (EDGE), 2017 IEEE International Conference on*, pages 47–54. IEEE, 2017.

[261] Ya-Ju Yu, Te-Chuan Chiu, Ai-Chun Pang, Ming-Fan Chen, and Jiajia Liu. Virtual machine placement for backhaul traffic minimization in fog radio access networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[262] Deze Zeng, Lin Gu, Song Guo, Zixue Cheng, and Shui Yu. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. *IEEE Transactions on Computers*, 65(12):3702–3712, 2016.

[263] Hua-Jun Hong, Pei-Hsuan Tsai, An-Chieh Cheng, Md Yusuf Sarwar Uddin, Nalini Venkatasubramanian, and Cheng-Hsin Hsu. Supporting internet-of-things analytics in a fog computing platform. In *Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on*, pages 138–145. IEEE, 2017.

[264] Atakan Aral and Tolga Ovatman. A decentralized replica placement algorithm for edge computing. *IEEE Transactions on Network and Service Management*, 2018.

[265] Lei Zhao, Jiajia Liu, Yongpeng Shi, Wen Sun, and Hongzhi Guo. Optimal placement of virtual machines in mobile edge computing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[266] R. I. Tinini, L. C. M. Reis, D. M. Batista, G. B. Figueiredo, M. Tornatore, and B. Mukherjee. Optimal placement of virtualized bbu processing in hybrid cloud-fog ran over twdm-pon. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.

[267] Nitinder Mohan, Pengyuan Zhou, Keerthana Govindaraj, and Jussi Kangasharju. Managing data in computational edge clouds. In *Proceedings of the Workshop on Mobile Edge Communications*, pages 19–24. ACM, 2017.

[268] He Zhu and Changcheng Huang. Availability-aware mobile edge application placement in 5g networks. *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, 2017.

[269] Juan Liu, Bo Bai, Jun Zhang, and Khaled B Letaief. Cache placement in fog-rans: From centralized to distributed algorithms. *IEEE Transactions on Wireless Communications*, 16(11):7039–7051, 2017.

[270] Haolong Xiang, Xiaolong Xu, Haoquan Zheng, Shu Li, Taotao Wu, Wanchun Dou, and Shui Yu. An adaptive cloudlet placement method for mobile applications over gps big data. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.

[271] Utsav Drolia, Katherine Guo, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan. Cachier: Edge-caching for recognition applications. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 276–286. IEEE, 2017.

[272] Kangkang Li and Jarek Nabrzyski. Traffic-aware virtual machine placement in cloudlet mesh with adaptive bandwidth. In *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 49–56. IEEE, 2017.

[273] Onur Ascigil, Truong Khoa Phan, Argyrios G Tasiopoulos, Vasilis Sourlas, Ioannis Psaras, and George Pavlou. On uncoordinated service placement in edge-clouds. In *Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on*, pages 41–48. IEEE, 2017.

[274] Tayebeh Bahreini and Daniel Grosu. Efficient placement of multi-component applications in edge computing systems. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 5. ACM, 2017.

[275] Isaac Lera, Carlos Guerrero, and Carlos Juiz. Comparing centrality indices for network usage optimization of data placement policies in fog devices. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 115–122. IEEE, 2018.

[276] Savvas Kastanakis, Pavlos Sermpezis, Vasileios Kotronis, and Xenofontas Dimitropoulos. Cabaret: Leveraging recommendation systems for mobile edge caching. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, MECOMM'18, pages 19–24. ACM, 2018.

[277] Lin Wang, Lei Jiao, Ting He, Jun Li, and Max Mühlhäuser. Service entity placement for social virtual reality applications in edge computing. In *INFOCOM 2018-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2018.

[278] Yi-Hsuan Hung, Chih-Yu Wang, and Ren-Hung Hwang. Combinatorial clock auction for live video streaming in mobile edge computing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018.

[279] Nguyen B Truong, Gyu Myoung Lee, and Yacine Ghamri-Doudane. Software defined networking-based vehicular adhoc network with fog computing. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 1202–1207. IEEE, 2015.

[280] Akram Hakiri, Bassem Sellami, Prithviraj Patil, Pascal Berthou, and Aniruddha Gokhale. Managing wireless fog networks using software-defined networking. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 1149–1156, Oct 2017.

[281] Yiming Xu, V Mahendran, and Sridhar Radhakrishnan. Towards sdn-based fog computing: Mqtt broker virtualization for effective and reliable delivery. In *Communication Systems and Networks (COMSNETS), 2016 8th International Conference on*, pages 1–6. IEEE, 2016.

[282] Cheng Li, Zhengrui Qin, Ed Novak, and Qun Li. Securing sdn infrastructure of iot–fog networks from mitm attacks. *IEEE Internet of Things Journal*, 4(5): 1156–1164, 2017.

[283] Vishal Sharma, Ilsun You, Francesco Palmieri, Dushantha Nalin K Jayakody, and Jun Li. Secure and energy-efficient handover in fog networks using blockchain-based dmm. *IEEE Communications Magazine*, 56(5):22–31, 2018.

[284] Thomas Rausch, Stefan Nastic, and Schahram Dustdar. Emma: Distributed qos-aware mqtt middleware for edge computing applications. In *Cloud Engineering (IC2E), 2018 IEEE International Conference on*, pages 191–197. IEEE, 2018.

[285] Yuanguo Bi, Guangjie Han, Chuan Lin, Qingxu Deng, Lei Guo, and Fuliang Li. Mobility support for fog computing: An sdn approach. *IEEE Communications Magazine*, 56(5):53–59, 2018.

[286] Min Chen and Yixue Hao. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 36(3):587–597, 2018.

[287] Nisha Talagala, Swaminathan Sundararaman, Vinay Sridhar, Dulcardo Arteaga, Qianmei Luo, Sriram Subramanian, Sindhu Ghanta, Lior Khermosh, and Drew Roselli. ECO: Harmonizing edge and cloud with ml/dl orchestration. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[288] Tianchu Zhao, Sheng Zhou, Xueying Guo, and Zhisheng Niu. Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[289] Xueying Guo, Rahul Singh, Tianchu Zhao, and Zhisheng Niu. An index based task assignment policy for achieving optimal power-delay tradeoff in edge cloud systems. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7. IEEE, 2016.

[290] Bo Li, Yijian Pei, Hao Wu, and Bin Shen. Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. *The Journal of Supercomputing*, 71(8):3009–3036, 2015.

[291] Luiz F Bittencourt, Javier Diaz-Montes, Rajkumar Buyya, Omer F Rana, and Manish Parashar. Mobility-aware application scheduling in fog computing. *IEEE Cloud Computing*, 4(2):26–35, 2017.

[292] Ajay Kattepur, Hemant Kumar Rath, and Anantha Simha. A-priori estimation of computation times in fog networked robotics. In *Edge Computing (EDGE), 2017 IEEE International Conference on*, pages 9–16. IEEE, 2017.

[293] Ajay Kattepur, Harshit Dohare, Visali Mushunuri, Hemant Kumar Rath, and Anantha Simha. Resource constrained offloading in fog computing. In *Proceedings of the 1st Workshop on Middleware for Edge Clouds & Cloudlets*, page 1.

ACM, 2016.
[294] Yucen Nan, Wei Li, Wei Bao, Flavia C Delicato, Paulo F Pires, Yong Dou, and Albert Y Zomaya. Adaptive energy-aware computation offloading for cloud of things systems. *IEEE Access*, 5:23947–23957, 2017.
[295] Yong Xiao and Marwan Krunz. Qoe and power efficiency tradeoff for fog computing networks with fog node cooperation. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.
[296] Gilsoo Lee, Walid Saad, and Mehdi Bennis. An online secretary framework for fog network formation with minimal latency. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[297] Xu Chen and Junshan Zhang. When d2d meets cloud: Hybrid mobile task offloadings in fog computing. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[298] Bo Yin, Wenlong Shen, Yu Cheng, Lin X Cai, and Qing Li. Distributed resource sharing in fog-assisted big data streaming. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[299] Wenjie Wang and Wei Zhou. Computational offloading with delay and capacity constraints in mobile edge. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[300] Yuxuan Sun, Sheng Zhou, and Jie Xu. Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks. *IEEE Journal on Selected Areas in Communications*, 35(11):2637–2646, 2017.
[301] Nguyen Ti Ti and Long Bao Le. Computation offloading leveraging computing resources from edge cloud and mobile peers. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[302] Ke Zhang, Yuming Mao, Supeng Leng, Sabita Maharjan, and Yan Zhang. Optimal delay constrained offloading for vehicular edge computing networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.
[303] Haisheng Tan, Zhenhua Han, Xiang-Yang Li, and Francis CM Lau. Online job dispatching and scheduling in edge-clouds. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.
[304] Xiaofan He, Juan Liu, Richeng Jin, and Huaiyu Dai. Privacy-aware offloading in mobile-edge computing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.
[305] Zheng Chang, Zhenyu Zhou, Tapani Ristaniemi, and Zhisheng Niu. Energy efficient optimization for computation offloading in fog computing system. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.
[306] Shanhe Yi, Zijiang Hao, Qingyang Zhang, Quan Zhang, Weisong Shi, and Qun Li. Lavea: latency-aware video analytics on edge computing platform. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 15. ACM, 2017.
[307] KE Desikan, Manikantan Srinivasan, and C Murthy. A novel distributed latency-aware data processing in fog computing-enabled iot networks. In *Proceedings of the ACM Workshop on Distributed Information Processing in Wireless Networks*, page 4. ACM, 2017.
[308] Ling Tang and Shibo He. Multi-user computation offloading in mobile edge computing: A behavioral perspective. *IEEE Network*, 32(1):48–53, 2018.
[309] Roberto Beraldi, Abderrahmen Mtibaa, and Hussein Alnuweiri. Cooperative load balancing scheme for edge computing resources. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 94–100. IEEE, 2017.
[310] Albert Jonathan, Abhishek Chandra, and Jon Weissman. Locality-aware load sharing in mobile cloud computing. In *Proceedings of the10th International Conference on Utility and Cloud Computing*, UCC '17, pages 141–150. ACM, 2017.
[311] Anil Singh, Nitin Auluck, Omer Rana, Andrew Jones, and Surya Nepal. Rt-sane: Real time security aware scheduling on the network edge. In *Proceedings of the10th International Conference on Utility and Cloud Computing*, pages 131–140. ACM, 2017.
[312] Christine Fricker, Fabrice Guillemin, Philippe Robert, and Guilherme Thompson. Analysis of an offloading scheme for data centers in the framework of fog computing. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 1(4):16, 2016.
[313] Lixing Chen and Jie Xu. Socially trusted collaborative edge computing in ultra dense networks. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 9. ACM, 2017.
[314] Ye Yu, Xin Li, and Chen Qian. Sdlb: A scalable and dynamic software load balancer for fog and mobile edge computing. In *Proceedings of the Workshop on Mobile Edge Communications*, pages 55–60. ACM, 2017.
[315] Xinchen Lyu, Wei Ni, Hui Tian, Ren Ping Liu, Xin Wang, Georgios B Giannakis, and Arogyaswami Paulraj. Optimal schedule of mobile edge computing for internet of things using partial information. *IEEE Journal on Selected Areas in Communications*, 35(11):2606–2615, 2017.
[316] Xinchen Lyu, Chenshan Ren, Wei Ni, Hui Tian, and Ren Ping Liu. Distributed optimization of collaborative regions in large-scale inhomogeneous fog computing. *IEEE Journal on Selected Areas in Communications*, 36(3):574–586, 2018.
[317] Wei Li, Ting Yang, Flavia C Delicato, Paulo F Pires, Zahir Tari, Samee U Khan, and Albert Y Zomaya. On enabling sustainable edge computing with renewable energy resources. *IEEE Communications Magazine*, 56(5):94–101, 2018.
[318] Zijiang Hao, Shanhe Yi, and Qun Li. Edgecons: Achieving efficient consensus in edge computing networks. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.
[319] Phuong Nguyen et al. Bracelet: Hierarchical edge-cloud microservice infrastructure for scientific instruments' lifetime connectivity. Technical report, UIUC, 2018.
[320] Dimitris Deyannis, Rafail Tsirbas, Giorgos Vasiliadis, Raffaele Montella, Sokol Kosta, and Sotiris Ioannidis. Enabling gpu-assisted antivirus protection on android devices through edge offloading. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 13–18. ACM, 2018.
[321] Hyuk-Jin Jeong, InChang Jeong, Hyeon-Jae Lee, and Soo-Mook Moon. Computation offloading for machine learning web apps in the edge server environment. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018.
[322] Junguk Cho, Karthikeyan Sundaresan, Rajesh Mahindra, Jacobus Van der Merwe, and Sampath Rangarajan. Acacia: context-aware edge computing for continuous interactive applications over mobile networks. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 375–389. ACM, 2016.
[323] Yuhua Lin and Haiying Shen. Cloudfog: Leveraging fog to extend cloud gaming for thin-client mmog with high quality of service. *IEEE Transactions on Parallel and Distributed Systems*, 28(2):431–445, 2017.
[324] Dominik Schäfer, Janick Edinger, Sebastian VanSyckel, Justin Mazzola Paluska, and Christian Becker. Tasklets: Overcoming heterogeneity in distributed computing systems. In *Distributed Computing Systems Workshops (ICDCSW), 2016 IEEE 36th International Conference on*, pages 156–161. IEEE, 2016.
[325] Julien Gedeon, Christian Meurisch, Disha Bhat, Michael Stein, Lin Wang, and Max Mühlhäuser. Router-based brokering for surrogate discovery in edge computing. In *Distributed Computing Systems Workshops (ICDCSW), 2017 IEEE 37th International Conference on*, pages 145–150. IEEE, 2017.
[326] Marica Amadeo, Claudia Campolo, Antonella Molinaro, Cristina Rottondi, Giacomo Verticale, et al. Securing the mobile edge through named data networking. In *Internet of Things (WF-IoT), 2018 IEEE 4th World Forum on*, pages 80–85, 2018.
[327] Giacomo Tanganelli, Carlo Vallati, and Enzo Mingozzi. Edge-centric distributed discovery and access in the internet of things. *IEEE Internet of Things Journal*, 5 (1):425–438, 2018.
[328] Kazuya Okada, Shigeru Kashihara, Nao Kawanishi, Nobuo Suzuki, Keizo Sugiyama, and Youki Kadobayashi. Goedge: A scalable and stateless local breakout method. In *Proceedings of the 2018 Workshop on Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms*, pages 29–34. ACM, 2018.
[329] Riccardo Venanzi, Burak Kantarci, Luca Foschini, and Paolo Bellavista. Mqtt-driven sustainable node discovery for internet of things-fog environments. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.
[330] Songze Li, Mohammad Ali Maddah-Ali, and A Salman Avestimehr. Coding for distributed fog computing. *IEEE Communications Magazine*, 55(4):34–40, 2017.
[331] Matthew Malensek, Sangmi Lee Pallickara, and Shrideep Pallickara. Hermes: Federating fog and cloud domains to support query evaluations in continuous sensing environments. *IEEE Cloud Computing*, 4(2):54–62, 2017.
[332] Shusen Yang. Iot stream processing and analytics in the fog. *IEEE Communications Magazine*, 55(8):21–27, 2017.
[333] Prem Prakash Jayaraman, Joao Bártolo Gomes, Hai Long Nguyen, Zahraa Said Abdallah, Shonali Krishnaswamy, and Arkady Zaslavsky. Cardap: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog. In *East European Conference on Advances in Databases and Information Systems*, pages 192–206. Springer, 2014.
[334] John K Zao, Tchin Tze Gan, Chun Kai You, Sergio José Rodríguez Méndez, Cheng En Chung, Yu Te Wang, Tim Mullen, and Tzyy Ping Jung. Augmented brain computer interaction based on fog computing and linked data. In *Intelligent Environments (IE), 2014 International Conference on*, pages 374–377. IEEE, 2014.
[335] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodík, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *Computer*, 50(10):58–67, 2017.
[336] Bin Cheng, Apostolos Papageorgiou, Flavio Cirillo, and Ernoe Kovacs. Geelytics: Geo-distributed edge analytics for large scale iot systems based on dynamic topology. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 565–570. IEEE, 2015.
[337] Pengfei Hu, Huansheng Ning, Tie Qiu, Houbing Song, Yanna Wang, and Xuanxia Yao. Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things. *IEEE Internet of Things Journal*, 4(5):1143–1155, 2017.
[338] Lingjuan Lyu, Jiong Jin, Sutharshan Rajasegarar, Xuanli He, and Marimuthu Palaniswami. Fog-empowered anomaly detection in iot using hyperellipsoidal clustering. *IEEE Internet of Things Journal*, 4(5):1174–1184, 2017.

[339] Yan Huo, Chunqiang Hu, Xiaowei Qi, and Tao Jing. Lodpd: A location difference-based proximity detection protocol for fog computing. *IEEE Internet of Things Journal*, 4(5):1117–1124, 2017.
[340] Iman Azimi, Arman Anzanpour, Amir M Rahmani, Tapio Pahikkala, Marco Levorato, Pasi Liljeberg, and Nikil Dutt. Hich: Hierarchical fog-assisted computing architecture for healthcare iot. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):174, 2017.
[341] Wenting Shen, Jia Yu, Hui Xia, Hanlin Zhang, Xiuqing Lu, and Rong Hao. Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium. *Journal of Network and Computer Applications*, 82: 56–64, 2017.
[342] Pengfei Hu, Huansheng Ning, Tie Qiu, Yanfei Zhang, and Xiong Luo. Fog computing based face identification and resolution scheme in internet of things. *IEEE Transactions on Industrial Informatics*, 13(4):1910–1920, 2017.
[343] Yannis Nikoloudakis, Spyridon Panagiotakis, Evangelos Markakis, Evangelos Pallis, George Mastorakis, Constantinos X Mavromoustakis, and Ciprian Dobre. A fog-based emergency system for smart enhanced living environments. *IEEE Cloud Computing*, 3(6):54–62, 2016.
[344] Surat Teerapittayanon, Bradley McDanel, and HT Kung. Distributed deep neural networks over the cloud, the edge and end devices. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*, pages 328–339. IEEE, 2017.
[345] Goutham Kamath, Pavan Agnihotri, Maria Valero, Krishanu Sarker, and Wen-Zhan Song. Pushing analytics to the edge. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.
[346] Hua-Jun Hong, Jo-Chi Chuang, and Cheng-Hsin Hsu. Animation rendering on multimedia fog computing platforms. In *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*, pages 336–343. IEEE, 2016.
[347] Aditya Dhakal and KK Ramakrishnan. Machine learning at the network edge for automated home intrusion monitoring. In *Network Protocols (ICNP), 2017 IEEE 25th International Conference on*, pages 1–6. IEEE, 2017.
[348] Simone Mangiante, Guenter Klas, Amit Navon, Zhuang GuanHua, Ju Ran, and Marco Dias Silva. Vr is on the edge: How to deliver 360 videos in mobile networks. In *Proceedings of the Workshop on Virtual Reality and Augmented Reality Network*, pages 30–35. ACM, 2017.
[349] Jianbing Ni, Xiaodong Lin, Kuan Zhang, and Yong Yu. Secure and deduplicated spatial crowdsourcing: A fog-based approach. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–6. IEEE, 2016.
[350] Dawei Li, Theodoros Salonidis, Nirmit V Desai, and Mooi Choo Chuah. Deepcham: Collaborative edge-mediated adaptive deep learning for mobile object recognition. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 64–76. IEEE, 2016.
[351] Jongwon Yoon, Peng Liu, and Suman Banerjee. Low-cost video transcoding at the wireless edge. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 129–141. IEEE, 2016.
[352] Ketan Bhardwaj, Pragya Agrawal, Ada Gavrilowska, Karsten Schwan, and Adam Allred. Appflux: Taming app delivery via streaming. *Proc. of the Usenix TRIOS*, 2015.
[353] Minsung Jang, Hyunjong Lee, Karsten Schwan, and Ketan Bhardwaj. Soul: an edge-cloud system for mobile applications in a sensor-rich world. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 155–167. IEEE, 2016.
[354] Bozhao Qi, Lei Kang, and Suman Banerjee. A vehicle-based edge computing platform for transit and human mobility analytics. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 1. ACM, 2017.
[355] Hooman Peiro Sajjad, Ken Danniswara, Ahmad Al-Shishtawy, and Vladimir Vlassov. Spanedge: Towards unifying stream processing over central and near-the-edge data centers. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 168–178. IEEE, 2016.
[356] Kyungmin Lee, Jason Flinn, and Brian D Noble. Gremlin: scheduling interactions in vehicular computing. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 4. ACM, 2017.
[357] Bo Mei, Ruinian Li, Wei Cheng, Jiguo Yu, and Xiuzhen Cheng. Ultraviolet radiation measurement via smart devices. *IEEE Internet of Things Journal*, 4(4): 934–944, 2017.
[358] Utsav Drolia, Katherine Guo, and Priya Narasimhan. Precog: prefetching for image recognition applications at the edge. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 17. ACM, 2017.
[359] Tuan Nguyen Gia, Mingzhe Jiang, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. Fog computing in healthcare internet of things: A case study on ecg feature extraction. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, pages 356–363. IEEE, 2015.
[360] Ning Chen, Yu Chen, Yang You, Haibin Ling, Pengpeng Liang, and Roger Zimmermann. Dynamic urban surveillance video stream processing using fog computing. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pages 105–112. IEEE, 2016.
[361] Gorkem Kar, Shubham Jain, Marco Gruteser, Fan Bai, and Ramesh Govindan. Real-time traffic estimation at vehicular edge nodes. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 3. ACM, 2017.
[362] Ivan Lujic, Vincenzo De Maio, and Ivona Brandic. Distribution of semantic reasoning on the edge of internet of things. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–10. IEEE, 2018.
[363] Muhammad Ali, Ashiq Anjum, M Usman Yaseen, A Reza Zamani, Daniel Balouek-Thomert, Omer Rana, and Manish Parashar. Edge enhanced deep learning system for large-scale video stream analytics. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–10. IEEE, 2018.
[364] Abebe Abeshu and Naveen Chilamkurti. Deep learning: the frontier for distributed attack detection in fog-to-things computing. *IEEE Communications Magazine*, 56(2):169–175, 2018.
[365] Yue Cao, Houbing Song, Omprakash Kaiwartya, Bingpeng Zhou, Yuan Zhuang, Yang Cao, and Xu Zhang. Mobile edge computing for big-data-enabled electric vehicle charging. *IEEE Communications Magazine*, 56(3):150–156, 2018.
[366] Ghulam Muhammad, Mohammed F Alhamid, Mansour Alsulaiman, and Brij Gupta. Edge computing with cloud for voice disorder assessment and treatment. *IEEE Communications Magazine*, 56(4):60–65, 2018.
[367] Mohammad Hossein Yaghmaee and Alberto Leon-Garcia. A fog-based internet of energy architecture for transactive energy management systems. *IEEE Internet of Things Journal*, 2018.
[368] Surin Ahn, Maria Gorlatova, Parinaz Naghizadeh, Mung Chiang, and Prateek Mittal. Adaptive fog-based output security for augmented reality. In *Proceedings of the 2018 Morning Workshop on Virtual Reality and Augmented Reality Network*, pages 1–6. ACM, 2018.
[369] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In *INFOCOM 2018-IEEE Conference on Computer Communications, IEEE*. IEEE, 2018.
[370] Lorenzo Corneo and Per Gunningberg. Scheduling at the edge for assisting cloud real-time systems. In *Proceedings of the 2018 Workshop on Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms*, pages 9–14. ACM, 2018.
[371] Ketan Bhardwaj, Joaquin Chung Miranda, and Ada Gavrilovska. Towards iot-ddos prevention using edge computing. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.
[372] Chang Ge and Ning Wang. Real-time qoe estimation of dash-based mobile video applications through edge computing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018.
[373] Yunlong Mao, Shanhe Yi, Qun Li, Jinghao Feng, Fengyuan Xu, and Sheng Zhong. A privacy-preserving deep learning approach for face recognition with edge computing. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.
[374] En Li, Zhi Zhou, and Xu Chen. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, MECOMM'18, pages 31–36. ACM, 2018.
[375] Andy Rosales Elias, Nevena Golubovic, Chandra Krintz, and Rich Wolski. Where's the bear?-automating wildlife image processing using iot and edge cloud systems. In *Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on*, pages 247–258. IEEE, 2017.
[376] Xukan Ran, Haoliang Chen, Xiaodan Zhu, Zhenming Liu, and Jiasi Chen. Deepdecision: A mobile deep learning framework for edge video analytics. In *INFOCOM 2018-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2018.
[377] Qiang Liu, Siqi Huang, Johnson Opadere, and Tao Han. An edge network orchestrator for mobile augmented reality. In *INFOCOM 2018-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2018.
[378] Zeyi Tao and Qun Li. esgd: Communication efficient distributed deep learning on the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.
[379] Zhuangdi Xu, Harshit Gupta, and Umakishore Ramachandran. Sttr: A system for tracking all vehicles all the time at the edge of the network. In *Proceedings of the 12th ACM International Conference on Distributed and Event-based Systems*, pages 124–135. ACM, 2018.
[380] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9):1275–1296, 2017.
[381] Dario Bruneo, Salvatore Distefano, Francesco Longo, Giovanni Merlino, Antonio Puliafito, Valeria D'Amico, Marco Sapienza, and Giovanni Torrisi. Stack4things as a fog computing platform for smart city applications. In *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pages 848–853. IEEE, 2016.

[382] Xiaomin Xu, Sheng Huang, Lance Feagan, Yaoliang Chen, Yunjie Qiu, and Yu Wang. Eaaas: Edge analytics as a service. In *Web Services (ICWS), 2017 IEEE International Conference on*, pages 349–356. IEEE, 2017.

[383] Peng Liu, Lance Hartung, and Suman Banerjee. Lightweight multitenancy at the network's extreme edge. *Computer*, 50(10):50–57, 2017.

[384] Mohammad Abdullah Al Faruque and Korosh Vatanparvar. Energy management-as-a-service over fog computing platform. *IEEE internet of things journal*, 3(2): 161–169, 2016.

[385] Márcio Moraes Lopes, Wilson A Higashino, Miriam AM Capretz, and Luiz Fernando Bittencourt. Myifogsim: A simulator for virtual machine migration in fog computing. In *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*, pages 47–52. ACM, 2017.

[386] Bastien Confais, Adrien Lebre, and Benoît Parrein. An object store service for a fog/edge computing infrastructure based on ipfs and a scale-out nas. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 41–50. IEEE, 2017.

[387] Amin M Khan and Felix Freitag. On edge cloud service provision with distributed home servers. In *Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on*, pages 223–226. IEEE, 2017.

[388] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. Edgecloudsim: An environment for performance evaluation of edge computing systems. In *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*, pages 39–44. IEEE, 2017.

[389] Sebastián Echeverría, Dan Klinedinst, Keegan Williams, and Grace A Lewis. Establishing trusted identities in disconnected edge environments. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 51–63. IEEE, 2016.

[390] Mohammed Islam Naas, Jalil Boukhobza, Philippe Raipin Parvedy, and Laurent Lemarchand. An extension to ifogsim to enable the design of data placement strategies. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–8. IEEE, 2018.

[391] Albert Rafetseder, Lukas Pühringer, and Justin Cappos. Practical fog computing with seattle. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–7. IEEE, 2017.

[392] R. Mayer, L. Graser, H. Gupta, E. Saurez, and U. Ramachandran. Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In *2017 IEEE Fog World Congress (FWC)*, pages 1–6, 2017.

[393] Marc Körner, Torsten M Runge, Aurojit Panda, Sylvia Ratnasamy, and Scott Shenker. Open carrier interface: An open source edge computing framework. In *Proceedings of the 2018 Workshop on Networking for Emerging Applications and Technologies*, pages 27–32. ACM, 2018.

[394] Nam Ky Giang, Rodger Lea, Michael Blackstock, and Victor Leung. Fog at the edge: Experiences building an edge computing platform. In *Edge Computing (EDGE), 2018 IEEE International Conference on*, pages 1–9. IEEE, 2018.

[395] Antonio Coutinho, Fabiola Greve, Cassio Prazeres, and Joao Cardoso. Fogbed: A rapid-prototyping emulation environment for fog computing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[396] Peng Liu, Bozhao Qi, and Suman Banerjee. Edgeeye: An edge service framework for real-time intelligent video analytics. In *Proceedings of the 1st International Workshop on Edge Systems, Analytics and Networking*, pages 1–6. ACM, 2018.

[397] Christopher Streiffer, Animesh Srivastava, Victor Orlikowski, Yesenia Velasco, Vincentius Martin, Nisarg Raval, Ashwin Machanavajjhala, and Landon P Cox. eprivateeye: to the edge and beyond! In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 18. ACM, 2017.

[398] Yehia Elkhatib, Barry Porter, Heverson B Ribeiro, Mohamed Faten Zhani, Junaid Qadir, and Etienne Rivière. On using micro-clouds to deliver the fog. *IEEE Internet Computing*, 21(2):8–15, 2017.

[399] Raul Muñoz, Laia Nadal, Ramon Casellas, Michela Svaluto Moreolo, Ricard Vilalta, Josep Maria Fàbrega, Ricardo Martínez, Arturo Mayoral, and Fco Javier Vílchez. The adrenaline testbed: An sdn/nfv packet/optical transport network and edge/core cloud platform for end-to-end 5g and iot services. In *Networks and Communications (EuCNC), 2017 European Conference on*, pages 1–5. IEEE, 2017.

[400] Nicola Iotti, Marco Picone, Simone Cirani, and Gianluigi Ferrari. Improving quality of experience in future wireless access networks through fog computing. *IEEE Internet Computing*, 21(2):26–33, 2017.

[401] Claus Pahl, Sven Helmer, Lorenzo Miori, Julian Sanin, and Brian Lee. A container-based edge cloud paas architecture based on raspberry pi clusters. In *Future Internet of Things and Cloud Workshops (FiCloudW), IEEE International Conference on*, pages 117–124. IEEE, 2016.

[402] Raef Abdallah, Lanyu Xu, and Weisong Shi. Lessons and experiences of a diy smart home. In *Proceedings of the Workshop on Smart Internet of Things*, page 4. ACM, 2017.

[403] Saiful Hoque, Mathias Santos de Brito, Alexander Willner, Oliver Keil, and Thomas Magedanz. Towards container orchestration in fog computing infrastructures. In *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, volume 2, pages 294–299. IEEE, 2017.

[404] Bukhary Ikhwan Ismail, Ehsan Mostajeran Goortani, Mohd Bazli Ab Karim, Wong Ming Tat, Sharipah Setapa, Jing Yuan Luke, and Ong Hong Hoe. Evaluation of docker as edge computing platform. In *Open Systems (ICOS), 2015 IEEE Confernece on*, pages 130–135. IEEE, 2015.

[405] Daniele Santoro, Daniel Zozin, Daniele Pizzolli, Francesco De Pellegrini, and Silvio Cretti. Foggy: a platform for workload orchestration in a fog computing environment. In *Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on*, pages 231–234. IEEE, 2017.

[406] Luciano Baresi, Danilo Filgueira Mendonça, and Martin Garriga. Empowering low-latency applications through a serverless edge computing architecture. In *European Conference on Service-Oriented and Cloud Computing*, pages 196–210. Springer, 2017.

[407] Abhimanyu Gosain, Mark Berman, Marshall Brinn, Thomas Mitchell, Chuan Li, Yuehua Wang, Hai Jin, Jing Hua, and Hongwei Zhang. Enabling campus edge computing using geni racks and mobile resources. In *Edge Computing (SEC), IEEE/ACM Symposium on*, pages 41–50. IEEE, 2016.

[408] Sven Helmer, Claus Pahl, Julian Sanin, Lorenzo Miori, Stefan Brocanelli, Filippo Cardano, Daniele Gadler, Daniel Morandini, Alessandro Piccoli, Saifur Salam, et al. Bringing the cloud to rural and remote areas via cloudlets. In *Proceedings of the 7th Annual Symposium on Computing for Development*, page 14. ACM, 2016.

[409] Ilija Hadžić, Yoshihisa Abe, and Hans C Woithe. Edge computing in the epc: a reality check. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, page 13. ACM, 2017.

[410] Zhuo Chen, Wenlu Hu, Junjue Wang, Siyan Zhao, Brandon Amos, Guanhang Wu, Kiryong Ha, Khalid Elgazzar, Padmanabhan Pillai, Roberta Klatzky, Daniel Siewiorek, and Mahadev Satyanarayanan. An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, pages 14:1–14:14. ACM, 2017.

[411] Alina Buzachis, Antonino Galletta, Lorenzo Carnevale, Antonio Celesti, Maria Fazio, and Massimo Villari. Towards osmotic computing: Analyzing overlay network solutions to optimize the deployment of container-based microservices in fog, edge and iot environments. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–10. IEEE, 2018.

[412] C. Bailas, M. Marsden, D. Zhang, N. E. O'Connor, and S. Little. Performance of video processing at the edge for crowd-monitoring applications. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 482–487, 2018.

[413] Jared N Plumb and Ryan Stutsman. Exploiting google's edge network for massively multiplayer online games. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–8. IEEE, 2018.

[414] Andy Bavier, Mark Berman, Marshall Brinn, Rick McGeer, Larry Peterson, and Glenn Ricart. Realizing the global edge cloud. *IEEE Communications Magazine*, 56(5):170–176, 2018.

[415] Alessandro Carrega, Matteo Repetto, Giorgio Robino, and Giancarlo Portomauro. Openstack extensions for qos and energy efficiency in edge computing. In *Fog and Mobile Edge Computing (FMEC), 2018 Third International Conference on*, pages 50–57. IEEE, 2018.

[416] Xingzhou Zhang, Yifan Wang, and Weisong Shi. pcamp: Performance comparison of machine learning packages on the edges. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[417] Orestis Akrivopoulos, Na Zhu, Dimitrios Amaxilatis, Christos Tselios, Aris Anagnostopoulos, and Ioannis Chatzigiannakis. A fog computing-oriented, highly scalable iot framework for monitoring public educational buildings. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[418] Per Skarin, William Tärneberg, Karl-Erik Årzen, and Maria Kihl. Towards mission-critical control at the edge and over 5g. In *Edge Computing (EDGE), 2018 IEEE International Conference on*, pages 1–8. IEEE, 2018.

[419] Christian Esposito, Aniello Castiglione, Florin Pop, and Kim-Kwang Raymond Choo. Challenges of connecting edge and cloud computing: A security and forensic perspective. *IEEE Cloud Computing*, 4(2):13–17, 2017.

[420] Zuoxia Yu, Man Ho Au, Qiuliang Xu, Rupeng Yang, and Jinguang Han. Towards leakage-resilient fine-grained access control in fog computing. *Future Generation Computer Systems*, 78:763–777, 2018.

[421] Sultan Basudan, Xiaodong Lin, and Karthik Sankaranarayanan. A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing. *IEEE Internet of Things Journal*, 4(3):772–782, 2017.

[422] Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu, and Xiuzhen Cheng. Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, 21(2):34–42, 2017.

[423] Sultan Basudan, Xiaodong Lin, and Karthik Sankaranarayanan. An efficient compromised node revocation scheme in fog-assisted vehicular crowdsensing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[424] Rongxing Lu, Kevin Heung, Arash Habibi Lashkari, and Ali A Ghorbani. A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot. *IEEE Access*, 5:3302–3312, 2017.

[425] Qixu Wang, Dajiang Chen, Ning Zhang, Zhe Ding, and Zhiguang Qin. Pcp: A privacy-preserving content-based publish–subscribe scheme with differential privacy in fog computing. *IEEE Access*, 5:17962–17974, 2017.

[426] Xingshuo An, Xianwei Zhou, Xing Lü, Fuhong Lin, and Lei Yang. Sample selected extreme learning machine based intrusion detection in fog computing and mec. *Wireless Communications and Mobile Computing*, 2018, 2018.

[427] Debasis Giri, Mohammad S Obaidat, and Tanmoy Maitra. Sechealth: An efficient fog based sender initiated secure data transmission of healthcare sensors for e-medical system. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[428] Ting He, Ertugrul Necdet Ciftcioglu, Shiqiang Wang, and Kevin S Chan. Location privacy in mobile edge clouds: A chaff-based approach. *IEEE Journal on Selected Areas in Communications*, 35(11):2625–2636, 2017.

[429] Kewei Sha, Ranadheer Errabelly, Wei Wei, T Andrew Yang, and Zhiwei Wang. Edgesec: Design of an edge layer security service to enhance iot security. In *Fog and Edge Computing (ICFEC), 2017 IEEE 1st International Conference on*, pages 81–88. IEEE, 2017.

[430] Shigen Shen, Longjun Huang, Haiping Zhou, Shui Yu, En Fan, and Qiying Cao. Multistage signaling game-based optimal detection strategies for suppressing malware diffusion in fog-cloud-based iot networks. *IEEE Internet of Things Journal*, 5(2):1043–1054, 2018.

[431] Daojing He, Yinrong Qiao, Sammy Chan, and Nadra Guizani. Flight security and safety of drones in airborne fog computing systems. *IEEE Communications Magazine*, 56(5):66–71, 2018.

[432] Nafiseh Izadi Yekta and Rongxing Lu. Xrquery: Achieving communication-efficient privacy-preserving query for fog-enhanced iot. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[433] Yeojin Kim, Donghyun Kim, Junggab Son, Wei Wang, and YoungTae Noh. A new fog-cloud storage framework with transparency and auditability. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[434] Ching-Han Chen, Ming-Yi Lin, and Chung-Chi Liu. Edge computing gateway of the industrial internet of things using multiple collaborative microcontrollers. *IEEE Network*, 32(1):24–32, 2018.

[435] Saman Biookaghazadeh, Ming Zhao, and Fengbo Ren. Are fpgas suitable for edge computing? In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, Boston, MA, 2018. USENIX Association.

[436] David P Anderson. Boinc: A system for public-resource computing and storage. In *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10. IEEE Computer Society, 2004.

[437] David P Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, 2002.

[438] Jian Kong, Inwoong Kim, Xi Wang, Qiong Zhang, Hakki C Cankaya, Weisheng Xie, Tadashi Ikeuchi, and Jason P Jue. Guaranteed-availability network function virtualization with network protection and vnf replication. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[439] Afshin Taghavi Nasrabadi, Anahita Mahzari, Joseph D Beshay, and Ravi Prakash. Adaptive 360-degree video streaming using scalable video coding. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1689–1697. ACM, 2017.

[440] Anahita Mahzari, Afshin Taghavi Nasrabadi, Aliehsan Samiei, and Ravi Prakash. Fov-aware edge caching for adaptive 360 video streaming. In *Proceedings of the 2018 ACM on Multimedia Conference*. ACM, 2018.

[441] Amir Vahid Dastjerdi and Rajkumar Buyya. Fog computing: Helping the internet of things realize its potential. *Computer*, 49(8):112–116, 2016.

[442] Dragi Kimovski, Humaira Ijaz, Nishant Saurabh, and Radu Prodan. Adaptive nature-inspired fog architecture. In *Fog and Edge Computing (ICFEC), 2018 IEEE 2nd International Conference on*, pages 1–8. IEEE, 2018.

[443] Behzad Mirkhanzadeh, Alessio Ferrari, Zhen Lu, Ali Shakeri, Chencheng Shao, Marco Tacca, Miguel Razo, Mattia Cantono, Vittorio Curri, Giovanni Martinelli, et al. Two-layer network solution for reliable and efficient host-to-host transfer of big data. In *Photonic Networks and Devices*, pages NeTh2F–5. Optical Society of America, 2018.

[444] Hessam Moeini, I-Ling Yen, and Farokh Bastani. Routing in iot network for dynamic service discovery. In *Parallel and Distributed Systems (ICPADS), 2017 IEEE 23rd International Conference on*, pages 360–367. IEEE, 2017.

[445] Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jorg Ott. Consolidate iot edge computing with lightweight virtualization. *IEEE Network*, 32(1):102–111, 2018.

[446] Intel. Intel's fog reference design overview. [Online]. Available: https://www.intel.com/content/www/us/en/internet-of-things/fog-reference-design-overview.html.

[447] Bridget A Martin, Frank Michaud, Don Banks, Arsalan Mosenia, Riaz Zolfonoon, Susanto Irwan, Sven Schrecker, and John K Zao. Openfog security requirements and approaches. In *Fog World Congress (FWC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[448] Ahmad Darki, Alexander Duff, Zhiyun Qian, Gaurav Naik, Spiros Mancoridis, and Michalis Faloutsos. (poster) don't trust your router: Detecting compromised router. In *the IEEE Proceedings of the 12th International Conference on Emerging Networking Experiments and Technologies CoNEXT*, volume 16, 2016.

[449] Zygmunt J Haas and Ashkan Yousefpour. A privacy scheme for monitoring devices in the internet of things. In *Pervasive Computing Paradigms for Mental Health*, pages 153–165. Springer, 2016.

[450] Jafar Haadi Jafarian, Amirreza Niakanlahiji, Ehab Al-Shaer, and Qi Duan. Multi-dimensional host identity anonymization for defeating skilled attackers. In *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, pages 47–58. ACM, 2016.