Computer Networks 130 (2018) 94–120

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers

Kashif Bilal[a,b,*], Osman Khalid[b], Aiman Erbad[a], Samee U. Khan[c]

[a] Qatar University, Doha, Qatar
[b] COMSATS Institute of Information Technology, Pakistan
[c] North Dakota State University, USA

ARTICLE INFO

Article history:
Received 21 April 2017
Revised 26 September 2017
Accepted 9 October 2017
Available online 18 October 2017

Keywords:
Edge computing
Fog computing
Internet of Things

ABSTRACT

Advancements in smart devices, wearable gadgets, sensors, and communication paradigm have enabled the vision of smart cities, pervasive healthcare, augmented reality and interactive multimedia, Internet of Every Thing (IoE), and cognitive assistance, to name a few. All of these visions have one thing in common, i.e., delay sensitivity and instant response. Various new technologies designed to work at the edge of the network, such as fog computing, cloudlets, mobile edge computing, and micro data centers have emerged in the near past. *We use the name "edge computing" for this set of emerging technologies.* Edge computing is a promising paradigm to offer the required computation and storage resources with minimal delays because of "being near" to the users or terminal devices. Edge computing aims to bring cloud resources and services at the edge of the network, as a middle layer between end user and cloud data centers, to offer prompt service response with minimal delay. Two major aims of edge computing can be denoted as: (a) minimize response delay by servicing the users' request at the network edge instead of servicing it at far located cloud data centers, and (b) minimize downward and upward traffic volumes in the network core. Minimization of network core traffic inherently brings energy efficiency and data cost reductions. Downward network traffic can be minimized by servicing set of users at network edge instead of service provider's data centers (e.g., multimedia and shared data) Content Delivery Networks (CDNs), and upward traffic can be minimized by processing and filtering raw data (e.g., sensors monitored data) and uploading the processed information to cloud. This survey presents a detailed overview of potentials, trends, and challenges of edge computing. The survey illustrates a list of most significant applications and potentials in the area of edge computing. State of the art literature on edge computing domain is included in the survey to guide readers towards the current trends and future opportunities in the area of edge computing.

© 2017 Published by Elsevier B.V.

1. Introduction

Cloud computing brought a technological revolution and paradigm shift in the Information and Communication Technology (ICT) sector in the last decade. Cloud computing experienced a massive adoption in almost every domain of human life [1–4]. Data centers, the backbone and underlying resource architecture of cloud computing are constantly growing in size and number to meet the increasing resource demands [2]. Technological advances in personal gadgets and wearable computing are enabling a new stream of real-time and pervasive applications, such as cognitive assistance, augmented reality, traffic monitoring, vehicular tracking, and interactive video streaming [5]. Such applications demand real-time response, which is one of the major constraints in the cloud paradigm because of the delays from distant cloud data centers. As indicated in Fig. 1, a user's request to the cloud has to traverse multiple hops before reaching the cloud servers, thus increasing the response time.

The proliferation of mobile devices, which are predicted to be more than 50 billion devices by the year 2020, will produce massive amounts of data [6]. Moreover, the ever increasing data rates from the Internet of Things (IoT) devices will impose further challenges on the cloud computing infrastructure. IoT is an emerging technology that extends Internet connection to devices embedded with sensors, actuators, and RFID tags [7]. IoT devices collect sensory data from the surrounding environment with a requirement to provide scalable infrastructure to communicate, process, and store

* Corresponding author at: Qatar University, Computer Science and Engineering, Doha, Qatar.
E-mail addresses: kashif@qu.edu.qa (K. Bilal), osman@ciit.net.pk (O. Khalid), aerbad@qu.edu.qa (A. Erbad), samee.khan@ndsu.edu (S.U. Khan).

https://doi.org/10.1016/j.comnet.2017.10.002
1389-1286/© 2017 Published by Elsevier B.V.

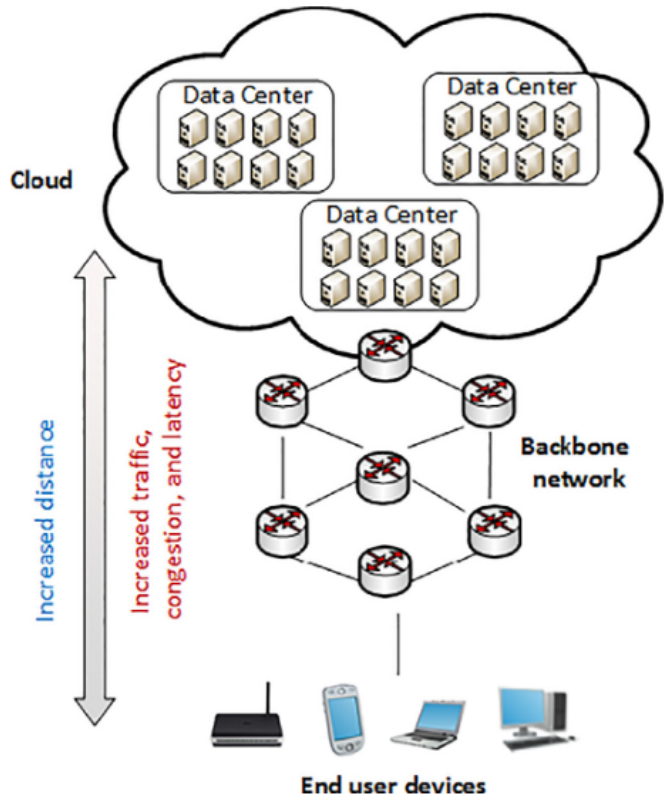

Fig. 1. Multiple hops between end user/devices and cloud data centers result in delayed response.

the data [8,9]. The number of such devices will reach billions in the coming years, with a large number of sensors monitoring and flooding the network with dynamic real-time data. According to Cisco Global Cloud Index [10], by the year 2019, 500 ZB of data will be produced by people, machines, and things, and 2.3 trillion GBs of data will be produced every day in the year 2020 [11]. IoT platforms demand low latency communication, need support for high degree of mobility, and real-time data analytics. Although cloud computing provides many benefits, the latency sensitive and data intensive IoT applications appear to be a challenge for current cloud computing system. The needs for real-time response and ever increasing data demands novel solutions. Edge computing (fogs, cloudlets, micro datacenters, and mobile edge computing) is emerging as a viable solution to these challenges, offering real-time response and near to end cloud services. Edge computing augments cloud computing by bringing networking and computational resources on edge devices near to the end user. An edge device can be a router, gateway, switch, or a base station, that provides an entry point into the service provider's core network. Edge devices are proposed to have sufficient computational and storage resources to meet real-time and resource intensive demands of end user. Generally, the edge computing platform comprises of a heterogeneous infrastructure of access points, switches, edge routers, servers, and end user devices. Compared to cloud computing, the edge provides low latency and reduced data traffic, as the applications are localized to the region where the edge is deployed.

We use the term *"Edge Computing Technologies" to encompass different emerging technologies situated at the edge of the network to provide computational and storage resources to deliver real-time communication with minimum latency*. Examples of such technologies include Fog computing, Mobile Edge Computing (MEC), Micro Data Centers (MDC), Cloudlet, and related technologies. The term edge computing or edge technologies used in this article refers to the set of these emerging technologies. Fog computing represents a platform that brings cloud computing to the proximity of end users [12,13]. The term Fog computing was coined initially by Cisco [13,14]. The main focus of fog computing is to equip the network edge and network devices with virtualized services, in terms of processing and storage along with offering network services. MEC is the edge technology initiated by European Telecommunications Standards Institute (ETSI) [15,16]. The major focus of MEC is Radio Access Networks (RANs) in 4G and 5G cellular networks. MEC offers edge computing by proposing a collocation of computation and processing resources at base stations. MDCs, initiated by Microsoft are small scaled version of data centers to extend the hyperspace cloud data centers [17,18]. MDCs aim to provide small size data centers extending the offered services of the cloud near to the end users. Concept of cloudlet, initiated by Carnegie Mellon University (CMU) is similar to MDC, as small scaled virtualized data center to serve users near the edge in a distributed fashion [19,20]. Some similar terms, such as Nano-data centers are also used in literature for similar concepts and objects [21,51].

Different edge technologies are defined independently; however, these technologies can cooperate and work together [12]. Considering the futuristic aspects of the Internet of Everything (IoE) [143] and recent trends in technology cooperation, such as Content Distribution Network Interconnection [22] and Heterogeneous Networks (HetNets) [144], it can be foreseen that various edge technologies will work in cooperation to support the overall vision of the IoE. Edge computing enables a large number of applications including vehicular communications, smart cities, smart grid, wireless sensor networks embedded with actuators, road traffic monitoring, pipe line monitoring, wind farms, smart traffic light system, railway monitoring, industrial control systems, and the applications in oil and gas explorations. IDC reported that by the year 2019, 45% of the data generated by IoT will be processed, stored, and analyzed on the edge [8]. Fig. 2 shows the some of the potential application areas of IoT and edge computing.

Edge computing technologies are in their infancy, with no standardized definitions, architectures, and protocols. Various researchers define edge technologies from their own perspective and models, which is expected for non-standardized technologies. A similar trend was observed in cloud computing as well before standardization of an official definition of cloud computing by National Institute of Science and Technology (NIST) in 2011 [23]. The lack of a standard definition leads to misconceptions in the relation among edge technologies, IoT, and cloud. Examples of such misconception mentioned in the literature, where authors claim that edge computing technologies will "move" or "replace" cloud with fog or decentralize the cloud paradigm to edges. For instance, [24] mentions that "Cloud is migrating to the edge of the network and the traditional Cloud Computing paradigm is not enough for the storage of Big Data produced by IoT". It needs to be clearly understood that edge computing technologies should not be considered as a substitute of cloud paradigm, rather, as shown in Fig. 3, these technologies will complement cloud and extend cloud services to the edges, so that the needs of applications with real-time requirements are satisfied [25]. For the big data analytics, and lengthy, resource intensive batch jobs, the cloud is a must. Similarly, there is also a confusion in understanding and perceiving the architecture of edge technologies, for instance, some authors treat fog computing as micro datacenters [26,27], while others focus mainly on the idea of strengthening and equipping networking components with extra processing and storage capabilities [25].

In this survey, we discuss various edge computing technologies, their potentials, applications, and challenges. Specifically, we provide a list of some potential areas in the field of edge computing (please see Fig. 4 for the taxonomy and topics discussed in this survey). The state of the art in various edge computing

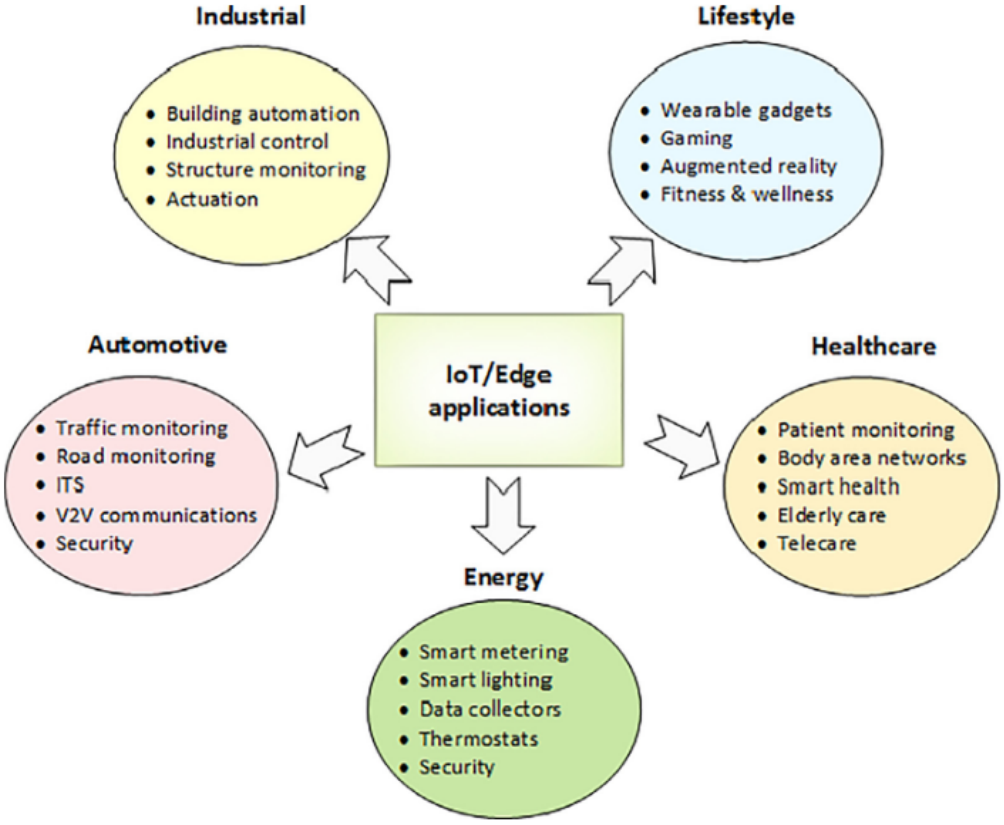

Fig. 2. Potential application areas of edge computing.

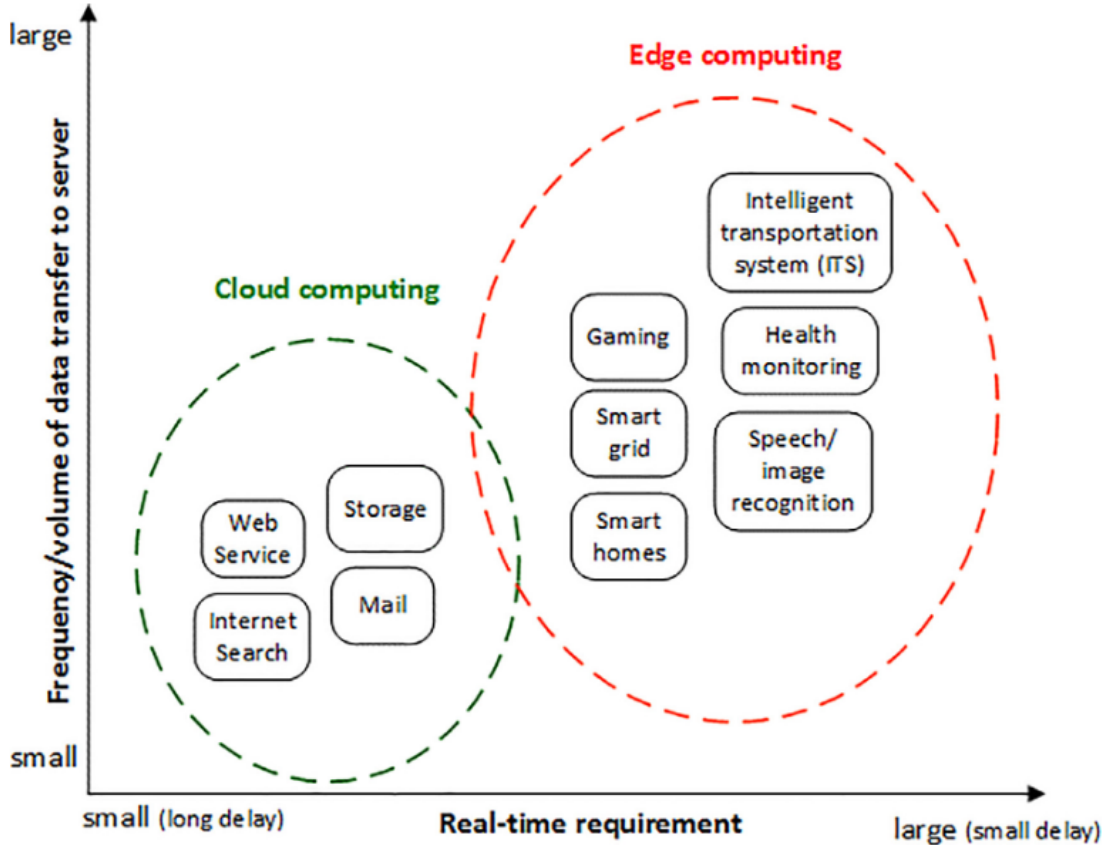

Fig. 3. The applications designed for traditional cloud computing have usually less frequent data transfer to cloud and can afford some slow response. However, the edge specific applications have more frequent interactions with edge servers and require a quicker response.

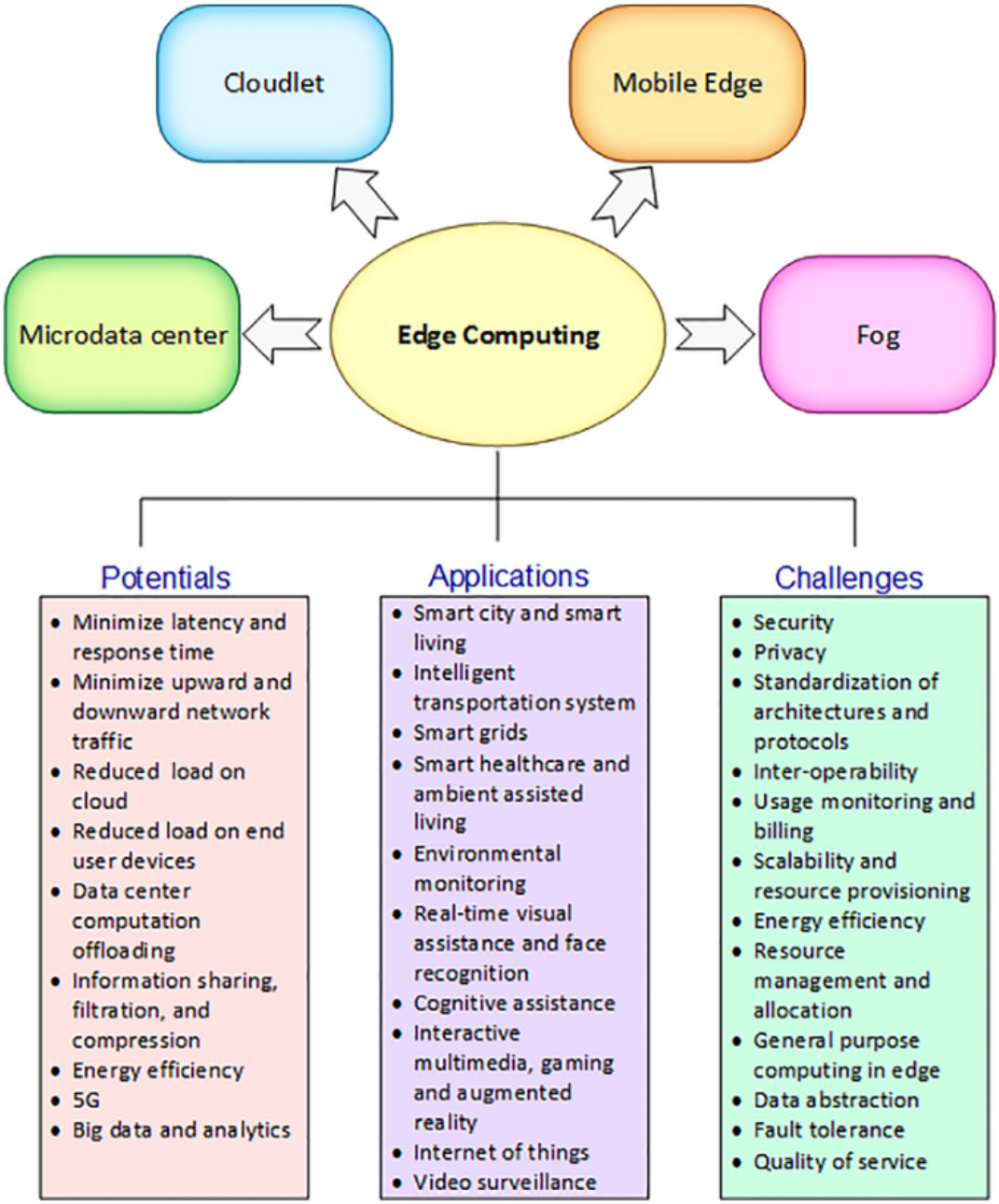

Fig. 4. Edge computing potentials, applications, and challenges.

technologies are also discussed in the article. Some of the authors have presented various aspects of edge computing in the literature. Luan et al. [28] highlighted main features of fog computing including its concept, architecture, and design goals. However, the other edge technologies are not covered. In a similar study, Bonomi et al. [13] outlined key characteristics of fog computing and discussed the role of fog computing in the IoT. Some basic applications are also discussed in the survey. A report on edge technologies [16] discussed and briefly compared the three technologies of edge computing: mobile edge, cloudlets, and fog computing, with no discussion on potential areas and applications. Stojmenovic et al. [29] discussed motivation and advantages of fog computing, and considered only these application areas: smart grid, smart traffic lights, and software defined networks. Yi et al. [30] discussed basic definition of fog computing and similar concepts and discussed various application scenarios. However, in [30] the discussion on existing techniques on edge computing is missing. Bonomi et al. [13] presented a discussion on fog computing in the context of IoT. Ahmed and Ahmed [31] and Beck et al. [15] discussed the taxonomy and key attributes of mobile edge computing. Azam et al. presented an article focusing on IoT and Cloud of Things (CoTs) [32]. The authors presented some of the potentials of the fog computing specifically considering the CoTs, i.e., amalgamation of IoTs and cloud computing. The authors presented various aspects of fog in consideration of edge computing as middleware to cloud, without presenting in-depth details. Dastderji and Buyya highlighted the potentials of fog computing for IoTs [33]. The authors briefly presented how fog computing may impact IoT systems to work better in a real-time environment and how it can save unnecessary transit traffic. The authors presented generic fog computing architecture and fog based distributed data processing models, and discussed various components involved in the model.

Yi et al. presented an overview of various concepts, applications, and issues in fog computing [30]. The authors in [34] discussed motivational scenarios of fog computing and provided some simulation results. Shi et al. [7] have discussed a few case studies of edge computing along with challenges and opportunities. Varghese et al. [35] presented a limited discussion on motivation, challenges, and opportunities of edge computing, without discussing the other edge technologies.

Most of the above mentioned surveys discussed various characteristics and applications of edge computing technologies in limited and isolated way. However, detailed study on various edge computing technologies, their potentials, applications, challenges, and the state of art, still needs to be addressed. Our survey attempts to address deficiencies in the existing surveys and provides a focused study on various edge computing technologies, their challenges, potentials and applications. To the best of our knowledge this is the first survey that provides in-depth details pertaining to edge computing and its various trends and potential areas. Moreover, this survey also presents state of the art in edge computing, which is missing in most of the existing surveys. Specifically, our contributions in this survey are as follows. In Section 2, we present an introduction of edge computing technologies and some motivational scenarios, followed by details of various edge computing technologies, i.e., Fog, Cloudlets, MDCs, and MECs. Section 3 presents a detailed study on the edge computing potential and most recent works in those areas and applications. Moreover, a detailed explanation on edge computing architectures, implementations, and evaluation mechanisms is provided. Section 4 highlights the open research challenges in the edge computing technologies, followed by conclusions in Section 5.

2. Edge computing technologies

The edge computing is based on the idea of placing small servers called edge servers or resource rich networking devices in the vicinity of end users/devices (see Fig. 5). In this way, some of the computational and data storage load is transferred from cloud platform to the edge servers. The end users' devices usually consist of wireless sensor networks, smart phones, wearable gadgets, and various IoT devices that require real-time response. Deploying computation and storage resources at the edge of the network can enable a large number of applications that require real-time response. A few examples of such applications include, but not limited to: (a) traffic monitoring and navigation, that involves traffic reporting and computation of routes for a specific region near to the edge, (b) data filtering and aggregation, that performs prefiltering of content and data at edge before sending it to cloud to reduce the data volume, and (c) augmented reality, real-time interactive video streaming, and health monitoring systems that can produce fast responses using edge nodes, thereby improving user experience for time-sensitive applications. In this section, first we discuss some motivational use cases and scenarios indicating why we should use edge computing in addition to cloud. Later, we explain various technologies within the domain of the edge computing.

2.1. Edge computing motivation

2.1.1. Reduced traffic load

The traditional User-Internet interaction model involves short requests from user to Internet services and receiving response. Some of the requested services, e.g., file downloading and specifically, Video on Demand (VoD) or live video streaming are comprised of very small data requests from user to the Service Provider (SP), and large volume of data flowing from SP to users. Considering the gigantic amount of data flowing from Internet to users, various solutions have been employed, such as CDN and caching to minimize the data and delay from SP to user [145]. For instance, cacheable contents are cached at ISP caches or CDN networks to minimize transit network data flow and delay [146–148]. However, the advent of new technologies, gadget proliferations, smart environments, and IoE are changing the data flow paradigm and patterns. Futuristic vision of smart and pervasive environments is foreseen to transmit massive volumes of data to the Internet. Consider live streaming, specifically crowd-sourced live streaming as an example, significant amount of data per second now flows from the users to SP and then disseminated globally from various SPs, such as Twitch (a crowd-sourced live gaming system) [149], YouTube Live [150], Periscope [151], and YouNow [152]. Netflix hosts a huge collection of entertainment video content. If 10% of 8 million people in New York want to stream movies from Netflix at the same time, it would require an infrastructure capacity of 1.6 Tera bits per second (Tbps) to handle all requests in parallel [36]. Despite remarkable improvements in bandwidth and server- side processing, the networks may still suffer in performance with huge viewership spikes. For instance, in a recent boxing match held in Las Vegas, USA, the live video streaming pay-per-view servers crashed and network got congested due to sudden rise in viewership [156]. If CDNs are not deployed within the networks, then the centrally hosted content must travel through many networks to reach the end users. In the futuristic scenario, current CDN based content delivery model is expensive, because, data still has to travel many hops between CDN and Internet Service Provider (ISP), before reaching to viewers'. For instance, consider the scenario of European football tournament final match, where the Akamai network served 3.3 million video streams concurrently to viewers, experiencing a peak load of 7.3 Tbps [142]. If multicast is not enabled, which is the general case because of configuration and security issues, then 5.7 Tbps data passing through multiple hops between CDN and ISP results in significant energy consumptions and network cost and management. Moreover, CDNs are passive storage designs, hosting large volumes of data, with generally no or very limited processing capabilities. On the fly transcoding of the videos are not available in the current CDN designs. Edge computing technologies offer a feasible solution in terms of very small delay and data filtration to fulfill the futuristic IoT, IoE, and smart world visions. If edge locations are used as data delivery and sharing points, huge volumes of transit data between CDNs and network edge can be saved [169]. Caching at the mobile edge (base stations/eNodeBs) may save considerable amount of backhaul network traffic. It has been shown that caching at the edge of the network considerably reduce access latency and network traffic [170]. Edge locations can perform on the fly video transcoding to create required video representation versions, minimizing the storage requirements, minimizing access delays, and maximizing viewers' QoE. Moreover, edge technologies may host dedicated services at edge to provide real-time response and data filtration. For instance, Akamai network have deployed edge computing networks to provide distributed execution of Java applications [37].

2.1.2. Minimizing the latency

The inherent cloud computing delays are challenging for applications that require real-time response, e.g., intelligent transportation systems, games, live streaming applications, and other safety critical applications, where such delays are intolerable. It is studied in [41] that for real-time visual guiding services, the preferred response time is between 25 ms to 50 ms. Moreover, high processing load imposed on cloud's central servers may cause scalability problems for the compute intensive applications and increase network overhead, resulting in slow response time and excessive utilization of the Internet bandwidth [42]. Inter-network data transfer leads to increased latency and congestion. Generally, Internet comprises

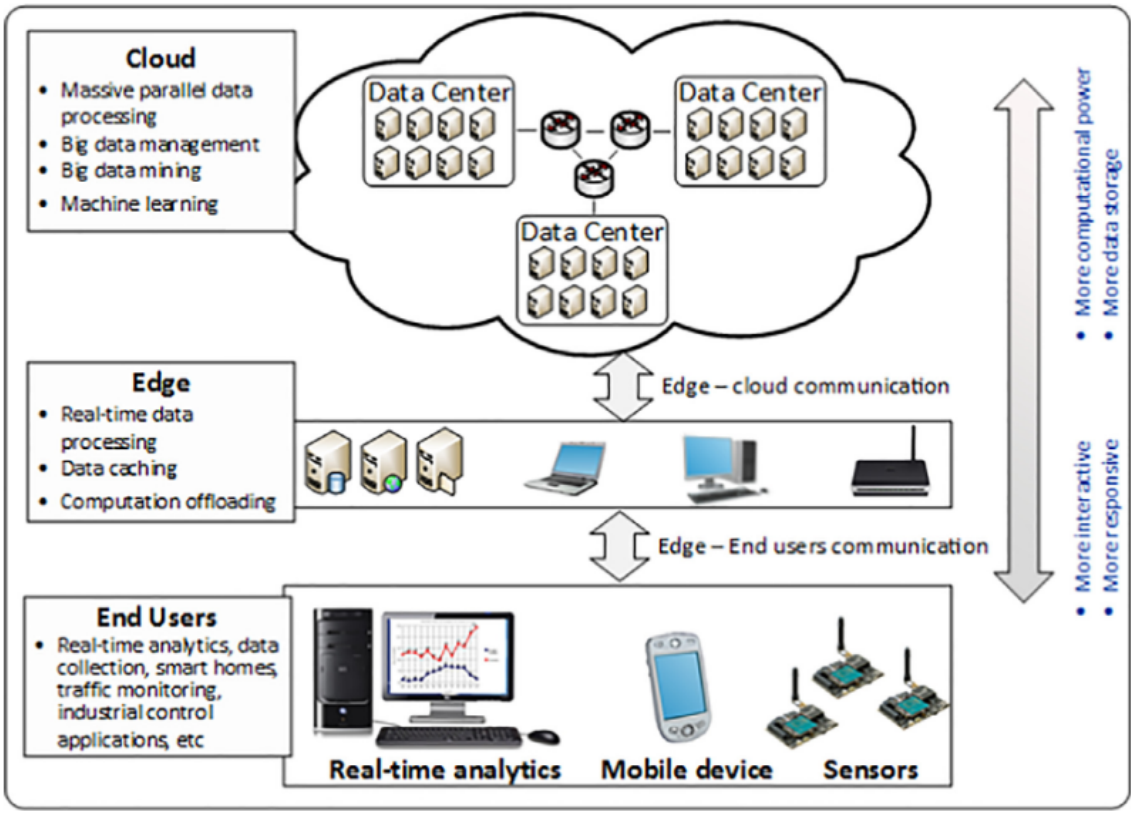

Fig. 5. Edge computing architecture.

Table 1
Effect of distance on round trip time (RTT), packet loss, throughput, and down time [37].

Distance (Server to user)	Network round trip time	Packet loss	Throughput	4 GB download time
Local: <100 miles	1.6 ms	0.6%	44 Mbps	12 min
Regional: 500–100 miles	16 ms	0.7%	4 Mbps	2.2 h
Cross-continent ~ 3000 miles	48 ms	1.0%	1 Mbps	8.2 h
Multi-continent ~ 6000 miles	96 ms	1.4%	0.4 Mbps	20 h

of thousands of interlinked networks, with each network providing access to a small percentage of end users. Even largest networks are usually accessed by only about 5% users [37]. As per statistics collected by Akamai, over 650 networks participate in reaching 90% of all access traffic [37]. A request to/from cloud may take several milliseconds to seconds to travel from client to cloud service provider [38]. Even a slight delay in a user's request may lead to the loss of subscribers and revenue. For instance, it was reported by Bing that a reduction of −1.8% in queries per user and −4.3% in revenue per user was observed due to queries slowing down by an interval of just 2 s [39]. A survey conducted by Forester concluded that majority of the online shoppers have suggested the website response time as a primary factor in giving their customer satisfaction feedback [37]. The survey also found out that more than 40% of the customers can only wait for 3 s for a page to load before leaving the website [37]. In another survey conducted by IDC, it was reported that improvement in performance and reliability of Akamai's enterprise application acceleration services yielded an annual increase from 0.$2 million to $3 million [40]. Therefore, content deployment at local ISPs (network edge) is critical for areas with low connectivity and high response time [37]. Recently, increasing number of ISPs have opened their edge services to other providers and subscribers, and offer various edge-based solutions, such as cloudlets, network functions virtualizations, and mobile edge computing. Table 1 shows that edge provides low latency and reduced data traffic, as the applications are localized to the region where the edge is deployed.

2.1.3. Reduced load on cloud

With the increase in location aware services, huge volumes of data is generated by end user devices on daily basis. For instance, the location-based service Foursquare has 60 million registered users [43] and it receives on the average, >5 million check-ins per day [44]. Similarly, several sports activity logging applications, such as Nike+ [45], Runtastic [46], Runkeeper [47], and Endomondo [48] are becoming popular. These applications run on smartphones and log daily activities of users with the help of various sensors, e.g., accelerometers, GPS, gyroscope, and temperature sensors, typically installed on smartphones. Mostly, the data recorded by the applications is sent to the cloud in the form of tuples, where each tuple contain several pieces of information, such as user id, longitude, latitude, time, distance, speed, duration, calories, weather, and other related items. For instance, a recent study on Endomondo revealed that a single workout on the average generates 170 GPS tuples, and average number of tuples generated per month is between 2.8 and 6.3 billion [49]. With 30 million users, the number of tuples generated per second could reach 25,000 tuples/sec [49]. Considering the IoT enabled smart cities, with thousands of sensors deployed, the numbers of tuples generated per second would be many times higher. When such high ve-

locity real-time data streams will be sent to the centralized cloud servers, the backbone network will get congested and the cloud servers may get overburdened. Moreover, not all of the sensed data is useful. For instance, sensors deployed in Large Hadron Collider (LHC) project generate around 500 EB data per day. However, 99.999% data is filtered out [49]. Edge computing can be leveraged by the application providers to locally process the data to filter unnecessary data, and generate real-time response for the users in the vicinity of the deployed edge. Moreover, data can be trimmed/filtered before sending to the cloud, thereby reducing the network traffic and processing burden from cloud servers.

2.1.4. Reduced load on end user devices

As discussed earlier, the end user devices and IoT generate huge volumes of data on which some form of analytics needs to be performed to generate useful information. However, if end devices, such as smartphone are subjected to such complex tasks, they may sooner run out of resources, e.g., battery drainage. Moreover, the devices may not be compatible due to heterogeneity of technologies. Therefore, the end devices can offload some of their high processing tasks to the nearby edge to reduce their load. Moreover, not all the data generated from the end device may contribute in the computation of useful information. For example, the study conducted on Endomondo sports activity tracking application revealed that even if a jogger stops to take rest, his sensors' stores the same values at regular intervals [49]. Therefore, some form of data filtering can be employed on the edge to discard the redundant data and only filtered data is sent to the cloud. Similarly, in interactive multimedia applications, such as free-view video, client's device is generally used to perform complex tasks like virtual view generation, which are resource intensive and results in battery depletion [87]. Performing such resource intensive jobs at the edge of the network, and delivering synthesized virtual view may result in significant bandwidth and energy savings at client's device.

2.1.5. Reducing energy consumption

Generally, the end user mobile devices and IoT are constrained by computing capabilities, battery life, and heat dissipation. Edge computing enables the offloading of energy consuming application from resource constrained end user devices to the edge servers. The majority of algorithms aim to minimize the energy consumption at the mobile device while subject to the execution delay acceptable by the offloaded application, or to find an optimal tradeoff between these two metrics. The energy consumption in using a cloud service usually depends on the following factors [51]: (a) energy consumption of end user device accessing the service, (b) energy consumption of data center, including energy consumed by internal network, storage, and servers, (c) the volume of traffic exchanged between the user and cloud, (d) the computational complexity of the task to be performed, (e) factors such as the number of users sharing a compute resource, and (f) the energy consumption of the transport network (aggregation, edge, and core networks). Costenaro and Duer studied energy consumption due to data transportation on the internet. The authors found out that 14% of the energy consumption in the Internet is due to the data transportation [50]. Jalali performed a detailed analysis of energy consumption by certain cloud-based applications, when those applications are run directly on cloud and on locally deployed fog based nano data centers [51]. The author showed that online interactive applications generate a substantial amount of traffic and consume more energy due to overheads arising from real-time interaction with the Cloud. The authors used various network analyzing tools to acquire traffic logs that showed the large traffic overhead is associated with establishing/tearing down TCP sessions very frequently and the volume of data transported to and from the user per session (measured in tens to hundreds of KB). The authors recommended that the fog based nano servers can complement the cloud for certain applications that can lead to energy savings, if the application or its components can be offloaded from centralized data centers and run on nano servers. Moreover, energy can be saved by employing intelligent client-side caching techniques, and optimizing the synchronization frequency of contents between edge and cloud [51]. Furthermore, data caching at edge locations reduce burden on the core network, which enable to reduce link rates using green technologies like Adaptive Link Rate (ALR) to make links energy proportional [4].

2.1.6. Data center computation offloading

Edge computing can also be exploited to offload computation from data centers that require limited resources to the edge nodes. For example, the live streaming applications, like Facebook Live, YouTube Live, and Livestream [153] allow users to perform live broadcast. It is reported that during a period of one minute, YouTube users upload 72 hours of new video, Facebook users share 2460,000 pieces of content, WhatsApp users share 347,222 photos, Instagram users post 216,000 new photos, and Vine users share 8333 videos [7,52]. Usually, when a video or photo is uploaded, e.g., to Facebook or YouTube, it is subjected to lossy compressions to reduce the media size. Uploading the high resolution photos and videos from user devices to the cloud occupy lots of bandwidth and may take lot of time in areas where internet connectivity is poor. Similar issues arise in live health monitoring applications, or smart city applications where live streams of data from surveillance cameras and other sensors needs to be uploaded to cloud. Edge computing can be utilized to transfer some of the compression related tasks to the edge devices near to the end users, before uploading to the cloud. Moreover, edge can also be used to encrypt the user data instead of uploading the raw data to the cloud, thereby ensuring security and privacy of user data in the intermediate hops.

In the next subsection, we discuss various technologies that we covered under the domain of edge computing. We discuss characteristics, similarities, and dissimilarities of these technologies, along with some practical examples.

2.2. Edge computing technologies

2.2.1. Fog

Fog computing represents a platform that brings cloud computing to the proximity of end users. The term "Fog" was initially introduced by Cisco and has an analogy with real-life fog [13]. As the clouds are far above the sky, the fog is closer to the earth. The same concept is used by fog computing, where the virtualized fog platform is deployed closer to the end users – between cloud and end users' devices. Although both cloud and fog paradigms share almost similar set of services, such as computation, storage, and networking, yet there are some differences between the two. The fog's deployment targets a specific geographic region. Moreover, the fog is specifically designed for applications that require real-time response with less latency, e.g., interactive and IoT applications. Alternatively, the cloud is centralized and being mostly far from the user, it suffers from some performance limitations in terms of latency and response time for real-time applications. The deployment of IoT in a two tiered architecture with cloud at one end and IoT devices at other end does not fulfill the requirements of low latency, mobility of the "things", and location awareness [53]. Therefore, as indicated in Fig. 6, a multi-tiered architecture is required in which the first part consists of IoT application deployed on "thing" which is an end user device, e.g., a vehicle. The second part of the architecture is fog, connected with end users through a router, access point, wireless access network, or an LTE base station. The final part of the 3-tier architecture is cloud's data center

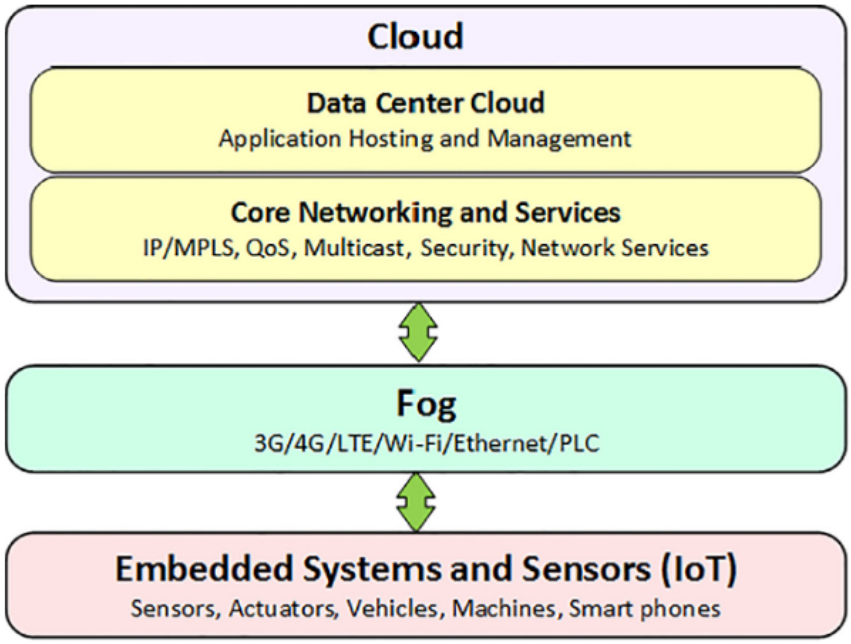

Fig. 6. 3-tier architecture consisting of cloud, fog, and IoT end devices layers.

(e.g. Amazon EC2 [154]). By having the 3-tier architecture, the fog allows IoT applications and services to be operated from edge of the network as well as from end devices, such as gateways, routers, access points, set top boxes, Road Side Units (RSUs), and Machine to Machine (M2M) gateways [33,34,53]. Moreover, such configuration allows fog to perform real-time monitoring, actuation, data analysis with reduced latency, improved QoS, and saving of bandwidth as data are processed at the edge of the network. Due to dense geographic coverage and distributed operations, fog computing promotes fault tolerance, reliability, and maintains scalability of the system. Fog can also perform preprocessing of data before sending it to cloud. This can further reduce the load on cloud network. In future applications, the fog computing is expected to deliver high quality data/video streaming to moving vehicles, mobile nodes, and public places through access points deployed for instance, along highways and malls [9,33].

Fog computing is a novel paradigm and faces various challenges apart from the issues it inherits from cloud computing. These challenges include management of heterogeneous devices, architectural issues, security, mobility, and privacy issues. Fog comprises of heterogeneous devices, with different types of data collected. Interoperability among heterogeneous devices is a challenging task. If the number of connected devices exceeds, this may raise scalability issues for the fog. Moreover, for proper management of resources and load balancing, an efficient resource scheduler is required. Designing such resource scheduler for heterogeneous devices and data is a challenging task. It is also critical to perform proper monitoring and management of devices, especially those running real-time applications. Moreover, the monitoring of traffic and billing mechanism is a must requirement. One of the major challenges in fog computing is to devise a fair billing model for the services offered. Fog services are offered using various pricing schemes and models, and the end users expect high QoS with minimum price. The billing model must be fair and balanced to attract more subscribers and generate high revenue. Due to unavailability of any standard billing model for fog, it is still an open research issue. Fog computing involves setup of expensive devices and networking. It is important to perform pre-deployment testing of fog platform using some simulation tool. However, there is no such standard simulation model/tool available at the moment for fog computing, which makes it an open research issue as well. Finally, the protection against malicious attackers and security threats is also a key research challenge for fog platform.

2.2.2. Cloudlets

Cloudlets are developed by a team at CMU [19,20]. Like fog computing, cloudlet also represents the middle tier of the 3-tier architecture: mobile device – cloudlet – cloud. Cloudlets are viewed as “data center in a box” with a purpose to bring cloud services closer to the mobile users. Internally, a cloudlet consists of a cluster of resource-rich multicore computers with high-speed internet connectivity and a high bandwidth wireless LAN for use by nearby mobile devices. For safety purposes, the cloudlets are enclosed in a tamper-resistant box for ensuring security in unmonitored areas [19].

Despite significant technological improvement, mobile devices, such as smart phones are still resource deficient when compared to other stationary devices like laptops and servers. This is primarily because of their smaller size, less memory, and shorter battery life. On the other side, there is a significant increase in development of various mobile applications. Most of the emerging applications, such as augmented reality, interactive media, speech recognition, natural language processing, require greater number of resources for processing with minimum latencies [19,34,52]. To meet such demands, cloudlets are designed with virtualization features to specifically provide computational resources to the mobile users. The mobile device, acting as a thin client, can offload computational tasks through a wireless network to a cloudlet, deployed one hop away. However, a cloudlet’s presence in mobile device’s proximity is necessary, as the end-to-end response time with executing applications must be smaller and predictable. If a device goes out of the range of cloudlet, then it should gracefully switch to the distant cloud, or in worst case, solely rely on its own resources. The cloudlet’s simplicity in management makes it trivial to be deployed at a business premises or near an experimental field where sensory devices are producing lot of data that requires processing. An example application can be mobile phone based language translation application. The VM running at cloudlet receives captured speech from mobile device, performs speech recognition and translation,

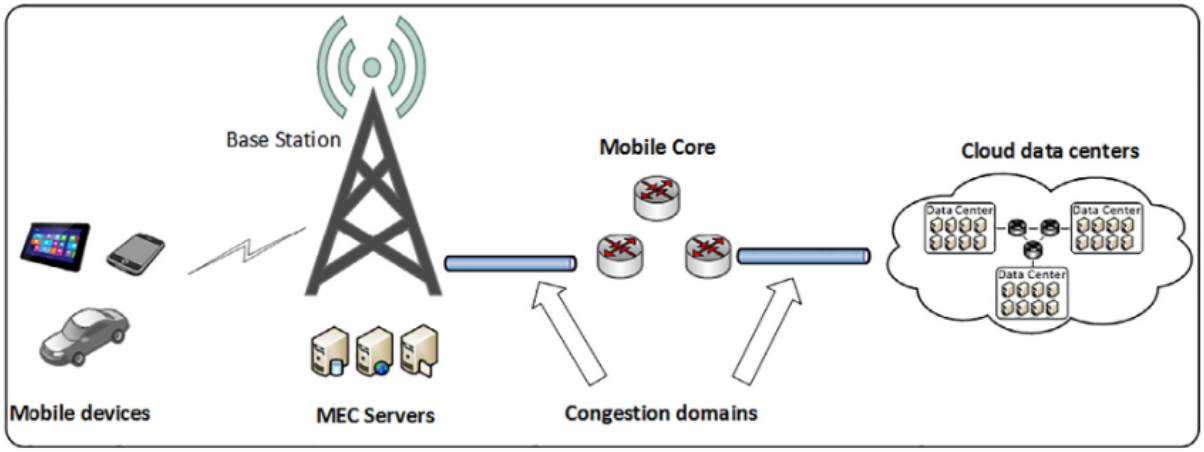

Fig. 7. Mobile edge computing.

and returns the output to the mobile device. The launched VM can be cloned to exploit parallelism in cloudlet. A basic difference between cloud and cloudlet is that a cloudlet contains only the soft state of data or code, whereas a cloud can contain both soft and hard state. Therefore, a cloudlet's failure does not result in data loss of mobile devices.

Satyanarayanan et al. [19] developed a cloudlet prototype, named Kimberly. The cloudlet infrastructure is setup on a desktop computer running Maemo 4.0 Linux, whereas the mobile device used in the prototype is Nokia N810. The mobile user utilizes VM technology to instantiate a service on nearby cloudlet and uses wireless LAN to interact with cloudlet. More technical details about the configuration and setup of Kimberly cloudlet can be found in [19]. Ha et al. [54] implemented a prototype for assisting people with cognitive decline using Google Glass and cloudlet technologies. The technology sends the captured image and other sensing information from a Google Glass to a cloudlet to perform real-time scene interpretation. The proposed system architecture is multi-tier to address the concerns related to limited battery and computation powers of mobile devices while performing the computational tasks on a connected cloudlet. The system gracefully degrades the services in case of network failures and if the device goes out of the range. Ye et al. [55] proposed a bus-based fog computing model in which the fog servers are deployed in buses. The roadside cloudlets can offload some portions of their computation tasks in case of overloading to a bus's fog servers. The authors proposed an optimal allocation strategy based on genetic algorithm using which, the cloudlets offload their tasks to fog servers deployed on busses. In addition, the bus servers can also offer the computational offloading for mobile devices in bus without any disruption and with improved QoS.

2.2.3. Micro datacenters

Microsoft Research under the supervision of Victor Bahl has introduced the concept of micro datacenters as an extension of today's hyper-scale cloud data centers [17,18]. Analogous to Cloudlets, Micro datacenters are also designed to meet demands of applications that require lower latency or that face constraints in terms of battery life or computations. A micro data center, shipped in one enclosure, is a self-contained, secure computing environment that includes all necessary computation, storage, and networking equipment to run customer applications. A micro datacenter can have a size range from 1–100 kW to meet the scalability and latency demands considering the IT load, and can also scale if more capacity is needed in the future.

Micro data centers have a number of applications in domains where real-time or near real-time data processing is required. Examples include, but not limited to, industrial automation, environmental monitoring, oil and gas exploration, construction sites, or any other applications where the sheer volumes of data requires on-site and real-time processing. Some of the micro datacenter's current implementations include Cisco's UCS [56], VCE's V-Blocks [57], or Dell's Active Systems [58]. These are pre-built systems that can be rapidly deployed and reconfigured. The company Schneider Electric offers micro datacenter solutions, such as Smart Bunker and Smart Data Safe [59]. Smart Bunker is designed to host 85 VMs within a 42 U rack assembly. The company also offers smaller micro datacenter's solutions with 23 U size deployed in single rack enclosure. Elliptical Mobile Solutions offer R.A.S.E.R. DX and HD systems [60]. The Elliptical Mobile has also created a complete stand-alone VPLEX system in conjunction with EMC, Microsoft, and AVNET [60]. Huawei is another important player in micro datacenters, whose MicroDC3000L 24 U systems can be used in environments of less than 100 users in an unattended, lights-out operations mode [61].

2.2.4. Mobile edge computing

MEC is designed to bring cloud computing capabilities and IT services environment at the edge of cellular networks [31]. The MEC offers lower latency, proximity, context and location awareness, and higher bandwidth. As reflected in Fig. 7, MEC servers are deployed at cellular base stations enabling flexible and rapid deployment of new applications and services for customers. MEC can be envisioned as cloud servers running at the edge of mobile networks and performing specific tasks that cannot be achieved with traditional cloud network infrastructure. Instead of forwarding all traffic to the remote cloud, the MEC shifts traffic targeted for the centralized cloud to the MEC servers. In this way, the MEC servers running applications and performing related processing tasks closer to the cellular customers reduce network congestion and response time of applications. Either the request is processed directly on MEC server sending quick response to the ender user, or, in some cases, the request may be forwarded to remote cloud.

In September 2014, the ETSI announced an industry specification for MEC [62]. The group of researchers are developing system architecture and standardizing a number of APIs essential for MEC [62]. In 2013, Nokia introduced MEC as a step towards automated driving. Usually, communications between cars and a central cloud has an end-to-end latency more than 100 ms. Base stations with distributed MEC cloudlets have shown an end-to-end latency of

lower than 20 ms. Nokia introduced MEC and geo service application to the LTE base stations that resulted in faster communications [63]. With connected driving via LTE, cars can now communicate almost in real-time over a larger distance and beyond the line of sight. This allows the cars to slow down in advance when there is an emergency situation.

3. Edge computing: the state-of-the-art

Edge computing is envisioned to assist in a number of domains with localized setup and configurations. In this section, we provide a detailed study of various edge potentials, and recent literature review in edge computing applications. We also discuss various edge architectures, and at the end, we present various implementations and simulation methodologies of edge computing as discussed in the literature. Table 2 presents a summary of state of art in edge computing in various domains.

3.1. Edge computing potentials and applications

Mission critical and latency sensitive applications mandate immediate response, and cannot afford communication delays incurred due to distant cloud and shared Internet medium. Some of the examples of real-time applications are, emergency and healthcare services, multi-player gaming, interactive multimedia, and augmented reality applications, etc. Services, such as visual guiding, demand a response time of 25 ms–50 ms, which cannot be achieved from cloud [64]. Ha et al. [38] evaluated the response time of face recognition applications under various network conditions. The study demonstrated that response time may increase to 4.02 s under worst network conditions compared to 620 ms required for a human subject. Such studies clearly demonstrate the needs of edge computing for real-time applications.

Besides minimal latency, serving users at nearby edge also brings related advantages. Some of the benefits are: (a) minimized core network traffic, (b) energy efficiency, and (c) data cost reduction. When users are served from edge for the applications that can be offloaded from cloud, the high volumes of TCP sessions' traffic is reduced on the core network, consequently reducing network traffic, congestion, and latency, and data transit energy [51]. Minimizing core network traffic is important, specifically, in terms of multimedia applications and IoTs, where huge volumes of data transfer from service provider to device (e.g., video streaming) and from device to service provider (e.g., sensor network monitored data and crowd-sourced video). Sharing and filtration of data can be performed at the edge of the network to minimize core network traffic. Edge based IoT solutions are reported to gain around 40% energy efficiency when IoT devices are served from edge locations instead of cloud [25]. Below, we present how edge computing can help to achieve least response time, minimize network core traffic, which results in achieving energy efficiency and reduced data cost. We provide a detailed discussion on a number of edge computing potential applications. Moreover, we also present the state-of-the-art in those areas to demonstrate how edge computing can benefit ICT sector in various ways, what are the possible applications, and future research areas.

3.1.1. Internet of things (IoT) and edge computing

IoT not only encompasses intelligent or M2M devices, but also covers the "dumb" and non-communicable devices, such as objects with Bar Code or RFID tags [65]. Such scenarios lead to IoE paradigm with trillions of interconnected devices, e.g., in case of smart cities (see Fig. 8), producing large streams of big data. With currently more than 9 billion devices connected, future connectivity is predicted to surpass approximately 50 billion devices [7,34,66]. Wireless Aggregated readings from sensors produce enormous amounts of data. For instance, Large Hadron Collider (LHC) in Switzerland uses data from around 150 million sensors, which generate around 500 Exabyte data per day. However, 99.999% data is filtered out and still, only 0.001% of the data produces 25 PB data annually [49]. Boeing 787, fully integrated with IoT sensors, will produce over half a TB of data per flight, said by Virgin Atlantic [67]. Similarly, the self-driving cars by Google generate nearly 1 GB of data every second [68], and the data requires real-time processing for making correct decisions. Wireless Sensor Networks (WSNs), the core components of IoT, are designed to operate at very low power to save battery life. The sensory nodes have small memory, processing power, and low bandwidth. Due to resource deficiency, the nodes cannot perform various compute intensive tasks related to data analysis and reporting. Efficient and real-time communication, processing, storage, and information retrieval of such massive volumes of connected devices is a challenge that can only be served by extensive distribution of processing and storage capability nearest to these devices. Edge technologies are foreseen to be one of the key players in future IoT and IoE paradigms [65]. Moreover, sending huge volumes of sensory data to cloud can cause increased congestion. In this scenario, the edge devices can undertake the task of data processing and analysis. Moreover, the data can be filtered and compressed by edge devices before sending to cloud to conserve bandwidth and minimize data flow. Considering the futuristic vision of IoT and IoE, with billions of connected objects, retrieved data needs to be processed and filtered. Being resource constrained, most of the IoT devices can be envisioned to rely on nearest edge nodes, for processing, filtering, and in some cases data storage. In addition, the actuators serving as edge devices can control physical actions, like open, close, move, etc. by acting in a closed loop system.

Edge computing has numerous applications in smart building control where the IoT devices acting as "things" embedded with sensors and network connectivity perform various monitoring and actuation tasks. Smart buildings are usually installed with numerous IoT based heterogeneous sensors that perform measurement of temperature, vibration, humidity, or various gas levels present in the building. Edge devices can process information from heterogeneous sources to deduce valuable information about the building's current health. Moreover, the edge devices can also make decisions on available data to operate (or actuate) sensors for specific tasks, for instance to lower temperature, inject fresh air, or open ventilators. By making use of edge computing, a building's security can also be improved by performing real-time video processing of surveillance cameras and activate warning alarms, or door locks. It is reported in [69] that there will be an increase in the combined global market for Internet of Things in Buildings (BIoT), rising from $25.65 billion in 2015 to $75.5 billion by 2021, and a combined annual growth rate in BIoT will be about 20.7%. Gooee, a company producing enterprise level IoT solutions for smart lighting has developed 'Full-Stack' operating system to allow manufacturers create IoT enabled lighting solutions [70]. Gooee has partnered with PointGrab [71], a provider of edge-analytics sensing solution. PointGrab performs real-time analytics using edge on the obtained data from buildings and applies its deep-learning and sensing technology to the building ecosystem. The company allows data capture about how and where occupants use building by utilizing its CogniPoint embedded-analytics sensors and edge-computing platform [72]. Intel has been actively involved in IoT enabled smart building solutions and offers a range of products, including, system on chips for secure edge computing, IoT gateways, analytics platforms, security management solutions [73]. Intel partnered with AVOB [74] for "Energy Saver" project, a small and medium sized building energy management solution to provide monitoring and remote control for smart energy management [73]. Intel's BMP integrated with

Table 2
Edge computing applied in various areas.

Area/Application	Reference	Idea presented
Internet of Things	[75]	Datta et al. proposed an IoT architecture for connected vehicles and utilized fog computing as a platform for providing IoT services to connected vehicles.
	[76]	Tawalbeh et al. used cloudlets for big data analytics in a mobile cloud computing environment.
	[77]	Brzoza-Woch et al. presented a fog-enabled embedded system for environmental monitoring.
	[71]	Companies like PointGrab and Gooee partnered to provide IoT enabled lighting solutions with the help of real-time edge computing
	[73]	Intel partnered with AVOB to develop edge enabled remote control and monitoring for IoT based smart energy management
Multimedia and edge computing	[54]	Ha et al. presented an architecture and prototype implementation of a cognitive assistance system using cloudlets and Google glass
	[91]	Simoens et al. proposed GigaSight, to store crowd-sourced videos in a local cloudlet for efficient uploading, downloading, and processing [91].
	[5]	Chen et al. presented the architecture and implementation details of a wearable cognitive assistance application using cloudlets
	[93]	Cai et al. proposed to use cloudlets to assist multi-player gaming to share the received video frames aiming to minimize the server transmission bandwidth usage
	[94]	Méndez et al. used ElectroEncephaloGram (EEG) headsets, smartphones, and fog computing to stream data captured from brain and send it to fog server for processing.
	[98]	Soyata et al. used cloudlets for real-time face recognition at airports named MOCHA using Mobile-Cloudlet-Cloud architecture.
Energy efficiency and edge	[21]	Valancius et al. proposed to employ nano data centers to minimize the energy consumption and latency for VoD
	[99]	Jalalai et al. identified scenarios in which running applications on nano servers used in fog are more efficient than running the same applications on centralized data centers
	[100]	Gai et al. proposed Dynamic Energy-aware Cloudlet-based Mobile cloud computing (DECM) model to minimize additional energy wastage in MCC scenario
	[101]	Sun and Ansari proposed Green Cloudlet Network (GCN) architecture for MCC aimed for process offloading between User Equipment (UE) and software clone at cloudlet with minimal delay and energy consumption
	[66]	Sarkar et al. presented a mathematical model for fog computing paradigm by mathematically quantifying power consumption, service latency, CO2 emission, and computational cost
Smart living	[102]	Presented a generic fog model for smart living comprising of 3 major components: Fog Edge Node (FEN), Fog Server (FS), and Foglet as a middleware program agent
	[40]	Li et al. proposed to use smart agents to mitigate lack of intelligence and reasoning in current smart objects in IoT and smart environments using swarm intelligence. They proposed Rainbow, an architecture for smart multi-agent system using fog computing.
	[103]	Sneppe and Namiot proposed to use mobile edge computing to share data among interoperating services of smart city
	[104]	Taleb et al. presented Follow-Me-Edge (FME) to enable emerging services for smart living using mobile edge computing
	[24]	Naranjo et al. introduced the concept of SmartLocalGrid (SLG) for communication between two micro-grids that allows communication among multiple devices efficiently to enable data processing and real-time decision locally without cloud support.
Health care	[113]	Masip-Bruin et al. proposed fog enabled solution for Chronic Obstructive Pulmonary Disease (COPD) patients' assistance that enable patients to roam and move freely with the automated provision of breathing and oxygen supply
	[6]	Fratu et al. employed fog computing to eWALL EU project to achieve real-time response for Mild Dementia (MD) and COPD patients.
	[112]	Cao et al. proposed FAST, a distributed analytics based fall monitoring system using fog computing for stroke mitigation.
	[54]	Ha et al. presented architecture and prototype implementation of a cognitive assistance system using cloudlets and Google glass
	[5]	Chen et al. presented the architecture and implementation details of a wearable cognitive assistance application using cloudlets
	[114]	Quwaider and Jararweh proposed cloudlet based architecture for collection and processing of data from Body Area Networks (BANs). Authors employed cloudlets to minimize packet-to-cloud energy and packet delay
	[115]	Amraoui and Sethom proposed cloudlet based pervasive healthcare monitoring system for chronic diseases using BANs
	[116]	Althebyan et al. presented a largescale e-health system using edge technologies. The authors proposed wearable textile based sensor, strategically distributed in clothing to continuously monitor patients' health condition
Communication efficiency and edge computing	[117]	Intharawijitr et al. analyzed fog computing in 5G mobile networks paradigm for communication and computation latencies
	[118]	Peng et al. presented the suitability and benefits of using edge computing paradigm in 5G networks and proposed Fog- Radio Access Networks (F-RAN) to mitigate the shortcomings of Cloud Radio Access Networks (CRAN).
	[119]	Nunna et al. presented various use cases for potential context-aware collaboration systems using 5G technology with MEC.
Edge computing architectures and resource management	[120]	Zhang et al. presented a multi-tiered architecture for delay sensitive cloud Data Service Subscribers (DSS).
	[121]	Eui-Nam et al. presented an architecture of a smart gateway with fog computing.
	[122]	Yin et al. proposed Tentacle, a dynamic and on the fly resource provisioning algorithm to procure edge servers for online service providers.
	[26]	Azam and Hu presented a service oriented strategy to effectively and efficiently manage resources in fog computing
	[123]	IoT devices are classified based on a device's nature and mobility to efficiently perform resource allocation. A detailed pricing model was also discussed

(*continued on next page*)

Table 2 (*continued*)

Area/Application	Reference	Idea presented
	[124]	Nippon Telegraph and Telephone Corporation developed an edge accelerated web platform (EAWP). The EAWP enables the web applications to run on edge servers.
	[125]	Zhu et al. proposed the concept of fog boxes to improve the website experience. The users connect with the internet via edge servers (fog boxes) using HTTP
	[27]	Aazam et al. proposed a service oriented model for fair management of IoT resources using fog computing that allows fair pricing, distribution, and management of resources in IoT
	[126]	Zeng et al. proposed a fog computing supported software-defined embedded system consisting of edge devices (cellular base stations) equipped with computation and storage resources and embedded client systems are general purpose hardware
Edge computing Implementation and Simulation	[129]	Cirani et al. proposed an architecture of Fog nodes as an IoT hub using Constrained Application Protocol (CoAP) protocol
	[128]	Butterfield evaluated Google's Go language for IoT and fog scenario.
	[25]	Sarkar and Misra presented theoretical modeling and mathematical formulation of fog computing architecture considering its various components
	[130]	Gupta et al. presented iFogSim, a simulation environment focusing on evaluation of resource management strategies for fog computing
	[132]	Orsini et al. proposed a mobile edge computing based programming framework CloudAware that allowed the users to offload their compute-intensive tasks from smartphones to the edge servers.
	[133]	Cisco's ParStream is a platform that allows handling of massive volumes of high-velocity data to provide real-time analytics at the edge
	[134]	Vortex fog computing provides platform independent interoperable solutions for intelligent data sharing and analytics platform for business critical IoT applications
	[53]	Cisco Data in Motion (DMo) technology allows data management and analysis of large volumes of data coming through IoT at the edge
	[135]	Cisco IOx is a combination of Cisco IOS, a network operating system, and Linux. The IOx allows hosting capabilities for fog applications, and allows management of network components, such as routers, switches, and compute modules

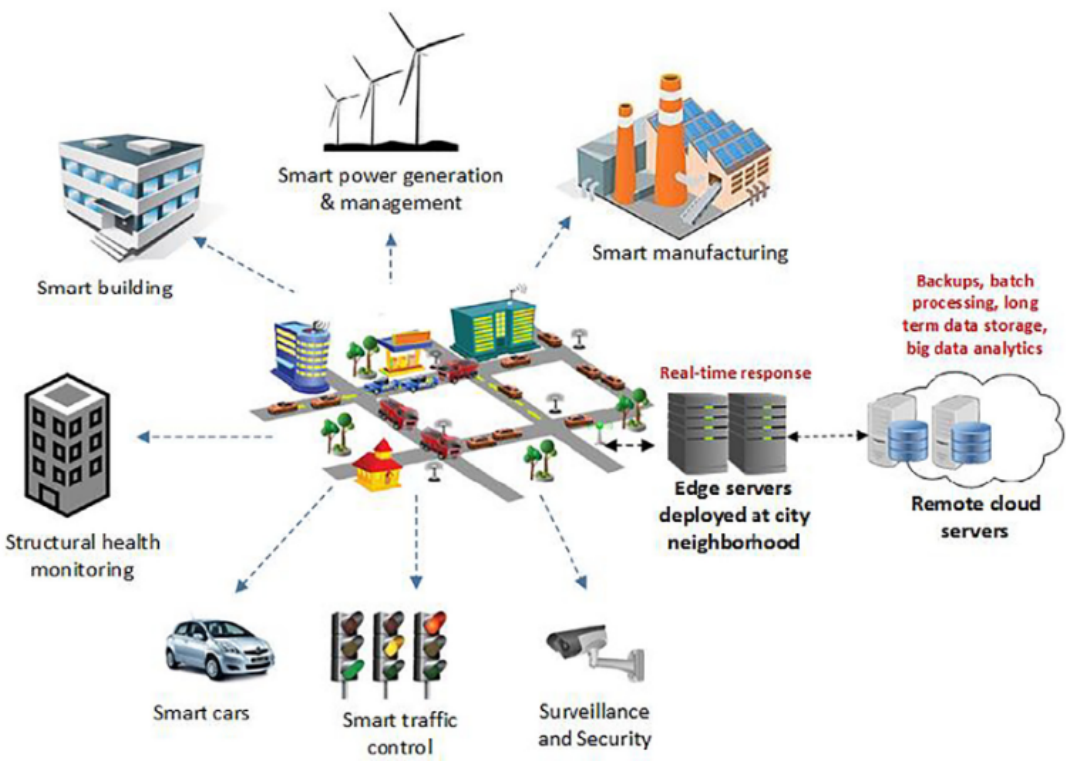

Fig. 8. IoT/Edge enabled smart city.

Candi PowerTools is a secure management platform that connects to various building systems and sensors to access data, performs data filtering, protocols translation, and secure transfer of data to cloud or to on-premise deployed edge servers [73].

Datta et al. proposed an IoT architecture for connected vehicles and utilized fog computing as a platform for providing IoT services to connected vehicles [75]. The architecture consists of: (a) smart phones and sensors fitted on vehicles acting as data source, (b) access points as RSUs, and (c) cloud system. The vehicular sensors utilize sensor markup language to report the sensory data. The data is communicated to RSUs that are connected with fog computing platform having a discovery module. The connected vehicles utilize discovery module to look for application and services provided by the RSUs. The fog platform is deployed at the middle nodes that are placed at the edge of the network. The vehicles are

able to connect with various fog services with low latency, due to wide geographic distribution of fog platform.

Edge computing can provide solutions for big data processing, where the big data represents the large volumes of data generated by IoT devices or sensor networks. With edge computing, on-demand elastic resources can be provisioned for locally processing the big data without sending data to cloud and suffering from drawback of higher latency and bandwidth consumption. A combination of edge and cloud computing can address big data acquisition, aggregation, and preprocessing, reducing the data transportation and storage on cloud. For instance, for large scale environmental monitoring system, the local data can be collected and processed at the regional fog nodes to provide timely feedback to end users, especially in emergency scenarios. In addition, the detailed and thorough analysis, and computational intensive tasks can be performed on the remote cloud. Tawalbeh et al. used cloudlets for big data analytics in a MCC environment [76]. The authors proposed a master-cloudlet management system for inter-cloudlet communications. The authors implemented the proposed model and reported considerable gains in energy efficiency and latency. Authors in [77] presented a fog-enabled embedded system for environmental monitoring. A smart levee monitoring system is proposed for flood warning. The authors proposed to use fog infrastructure to process and filter raw data, before sending it to cloud, and to support rescue teams in emergency situation without connecting to the Internet. The authors designed an embedded system using WSANs and fog for flood risk assessment. The basic idea is to collect the sensed data from WSN and transmit it to fog to analyze the data using an analysis and forecasting application.

To conclude, there are several practical applications of edge computing for IoT, including smart homes and big data processing. The major of focus of most IoT applications is energy conservation, such as implementing IoT based smart lighting [70,73,74]. The IoT generated big data needs to be processed to extract useful patterns [71]. The edge computing is also finding its applications in IoT enabled connected vehicles [75]. Tawalbeh et al. used cloudlets for big data analytics in a MCC environment [76]. In [77] authors utilized IoT and edge for environmental monitoring.

3.1.2. Multimedia and edge computing

Multimedia, specifically, video content is one of the major consumers of overall Internet bandwidth. It has been reported that in 2015, video data comprised around 70% of the total Internet traffic. These figures are predicted to rise to 82% in 2020 [78]. In future IoT scenarios, many multimedia generating gadgets, such as closed circuit TV and visual sensor networks will generate massive volumes of multimedia data [65]. As multimedia requires more bandwidth, processing, and storage, so handling such huge volumes in terms of communication, processing, and storage is a real challenge. Edge computing is envisioned to aid in such scenarios to minimize the overall end-to-end bandwidth usage, distribution, efficient processing, and storage for multimedia [79–81]. Multimedia delivery also incurs high costs. CDNs, like CloudFront charge substantially, when considering Tbps data delivery. It has been reported that YouTube Live and Twitch surpassed 1 Tbps mark during peak hours in 2014 [82]. In another study on Twitch trace analysis, it has been reported that Twitch surpassed 1.5 Tbps video content delivery to viewers across the globe [5]. It also needs to be considered that Internet connectivity and data rates are increasing every term, which means higher data access by users. In 2014 state of the Internet report, Akamai reported an average global bandwidth of 4.5 Mbps, with 59% users having more than 4 Mbps connectivity, among which 13% and 10% had 10 Mbps or higher and 15 Mbps or higher Internet connectivity, respectively [83]. Whereas, in 2016, the average bandwidth globally raised to 6.8 Mbps, with 73% connections having more than 4 Mbps. Among these 73% connections, 35% had more than 10 Mbps, 21% had more than 15 Mbps, and 21% had more than 25 Mbps Internet connectivity [84]. It can be seen that in 2016, way more users have 4 K ready Internet connectivity as compared to 2014. Such large volumes of data are charged heavily, e.g., Amazon CloudFront CDN charges $0.085 per GB for first 10 TB and $0.26 Per GB for higher usage of data transmission [85]. It was observed in 2015 captured logs, Twitch delivered video content at more than 1.5 Tbps [86]. Although, Twitch is owned by Amazon, so the payment matters may be internal. However, if one calculates the total cost required to transmit 1.5 Tera bits (192 Giga Bytes) using Amazon CloudFront with minimum charges, i.e., $0.02 /GB, then it will be $3.84 /s leading to $13,800 /h. Fog computing may be used to cache the popular content at edge and serve the local community from edge or from minimum possible hops, as CDNs are still many hops away from the users. Moreover, live content can also be disseminated from network edge, offering higher bandwidths to viewers. Specifically, in terms of interactive multimedia, which is strictly delay sensitive, e.g., multi-view and free-view video [87], switched view delivery and virtual view synthesis can be performed at edge with minimal delay. If we consider per bit energy consumption across the Internet hops, then it may be realized that even four hops (generally considered as average from CDN to users) may inhibit excessive amount of energy usage and Green House Gases (GHG) emissions.

2013 NSF workshop report predicted that "it will soon be possible to find a camera on every human body, in every room, on every street, and in every vehicle" [88]. Video surveillance plays a significant role in effective urban planning and management administratively, as well as for law enforcement departments. It is estimated that in 2013, that there was one surveillance camera for 11 persons in the United Kingdom [89]. Considering the futuristic scenario of such massive number of cameras and their streams uploaded to the Internet mandates feasible solutions for communication, processing, and storage. Surveillance information may come in a heterogeneous form from multiple sensors. Target tracking and object assessment in such surveillance environment requires information fusion and collective processing. Efficient extraction of information from various streams, analysis, and understanding requires resource, which can be provisioned from Cloud computing. However, long response time and delays prohibit using cloud computing for mission critical, sensitive surveillance, and tracking systems. Edge computing, however, offer the resources, as well as real-time response for such applications.

Most of the captured videos and pictures are stored locally. However, crowd-sourced based video streaming is gaining popularity. Twitch is estimated to serve around 50 million users every month with 150 billion minutes of live video [90]. Simoens et al. proposed GigaSight, to store crowd-sourced videos in a local cloudlet for efficient uploading, downloading, and processing [91]. Processing videos captured in a small geography at a local cloudlet enables searching and processing of related videos easily and efficiently. For instance, if some kid or dog is lost in some theme park or concert, recent videos from same event uploaded to the local cloudlet in recent times may be searched to find the missing.

With the emergence of wearable computing and gadgets, cognitive assistance based applications are becoming a reality. One of the major requirements of cognitive assistance applications is real-time response. Human subjects take from minimum 370 ms to maximum 620 ms to respond an unknown face [92]. The edge computing can be used for real-time cognitive assistance by integrating image capturing, sensing, and processing to deliver response instantly. More than 20 Million Americans suffer from some form of cognitive impairment, for whom, edge computing offers hope and a feasible platform. Ha et al. presented architecture and prototype implementation of a cognitive assistance system using cloudlets and Google glass in [54]. The authors detailed the archi-

tectural requirements of cognitive assistance systems and implementation details. Considering the high delay, cloud platforms cannot be used for task offloading, therefore, authors used cloudlet for efficient communication and processing. Similarly, Chen et al. presented the architecture and implementation details of a wearable cognitive assistance application using cloudlets [5]. The application is designed for cognitive assistance in four different tasks, i.e., free hand sketching, 2D Lego models assembling, context-relevant recommendation of YouTube tutorials, and playing a ping-pong game.

Constraints on network bandwidth, delay, and jitter in cloud gaming seriously impact users' Quality of Experience (QoE). Cai et al. proposed to use cloudlets to assist multi-player gaming to share the received video frames aiming to minimize the server transmission bandwidth usage [93]. Game server sends the encoded video to Adhoc-cloudlet, which in turn transmit the video to the group of connected players. Classification of brain state is a heavily computational intensive and delay sensitive real-time task. Méndez et al. [94] used ElectroEncephaloGram (EEG) headsets, smartphones, and fog computing to stream data captured from brain and send it to fog server for processing. EEG can be used to determine ones' brain states in real time. Based on the processed data, authors demonstrated the effectiveness of work by playing an online Brain Computer Interaction (BCI) "EEG Tractor Beam" game among different users in USA and Taiwan. Near to end users, fog servers successfully processed the data streams and classification/calibration is performed at the cloud servers. Previously, various BCI related projects, such as HeadIT [95], BrainMap [96], and PhysioNet [97] employed physiological signal processing. However, none of these projects were able to interact with their clients in real-time. Edge computing enables the real-time signal processing and enabled client interaction to perform various tasks. Such usage of edge computing resources can be foreseen to bring realistic applications. Soyata et al. used cloudlets for real-time face recognition at airports named MOCHA using Mobile-Cloudlet-Cloud architecture [98]. Cloudlets were employed to minimize the response time. Experimental results demonstrated that inclusion of cloudlets considerably enhanced the performance and response time of MOCHA.

To summarize, edge computing finds its place in numerous multimedia applications, especially in real-time processing of crowdsourced video streams [91] and cognitive assistance applications [54]. The existing works have utilized cloudlet based architectures for cognitive assistance [5] and multi-player gaming [93] to minimize latency and response time required for such applications. Similarly, edge has been utilized to perform real-time processing of EEG data acquired from brain [94] and real-time face recognition applications [98] in a bid to reduce the response time and latency.

3.1.3. Energy efficiency and edge

Energy efficiency is one of the mandatory and key concerns today because of environmental impacts, energy demand, and cost [1]. The ICT sector is one of the major energy consumer, estimated to consume more than 271 billion KWh of energy in data centers in 2010 [3]. Network infrastructure is also one of major energy consumer, estimated to consume around 15.6 billion KWh energy in 2010 [2]. ICT sector is also attributed as a major Green House Gases (GHG) contributor, emitting around 2% of global GHG emissions [4]. The GHG emissions by cloud datacenters are estimated to be 1034 t in 2020 [1], which clearly raise the environmental concerns and calls for appropriate solutions. In recent years, several proposals have been presented to employ edge computing for improving energy efficiency of cloud services and end user devices. MEC enables offloading of compute intensive and energy consuming applications from mobile devices to edge servers, thereby reducing the energy consumption of end devices. The majority of algorithms optimize the tradeoffs between energy consumption at mobile devices and execution delays caused by the offloading of the application. Gao et al. performed various experiments and showed that cloudlets can reduce an energy consumption by up to 42% in a mobile device [164]. Zhang et al. demonstrated that MEC can improve energy efficiency in heterogeneous networks by computation offloading [165]. In [166], the authors have investigated the energy-efficient resource allocation problem for computation offloading. Sardellitti et al. performed the joint optimization of radio and computational resources for multi-cell mobile-edge computing [167]. The major aim of the authors was to minimize energy consumption under latency and power budget constraints in [167]. The tradeoff between power consumption and transmission delay in the fog-cloud computing system is investigated in [168].

Jalalai et al. identified scenarios in which running applications on nano servers used in fog are more efficient than running the same applications on centralized data centers [51,99]. The authors proposed new energy models for shared and unshared network equipment to measure the energy in different scenarios. Nano servers were implemented using Raspberry Pi computers and were measured for traffic and power consumption. The energy consumption of data requests to nano servers were compared with data requests to centralized data centers using energy consumption models. The results indicated that energy can be saved on transport network, when the frequently used contents are pushed to the nano servers near to the requesting user, thus resulting in less traffic on backbone. The authors concluded that for efficient content storage and energy saving, the application architecture could be a hybrid of both fog and cloud.

Gai et al. explored the impact of edge computing considering energy and delay in Mobile Cloud Computing (MCC) [100]. The authors proposed Dynamic Energy-aware Cloudlet-based Mobile cloud computing (DECM) model to minimize additional energy wastage in MCC scenario. The authors proposed a web service at cloudlet layer to search and allocate appropriate cloud resources using dynamic computing for the request, considering energy and latency constraints. Sun and Ansari [101] proposed Green Cloudlet Network (GCN) architecture for MCC. GCN aims at process offloading between User Equipment (UE) and software clone at cloudlet with minimal delay and energy consumption. The GCN architecture also used SDN technology and proposed Cloudlet Network File System (CNFS) to protect data integrity. Sarkar et al. presented a mathematical model for fog computing paradigm by mathematically quantifying power consumption, service latency, CO_2 emission, and computational cost [66]. The performance of proposed model is evaluated by considering large number of Internet-connected devices demanding real-time service. The model is evaluated using a case study of devices generating traffic from hundred most populated cities, being served by eight geographically distributed data centers. The experimental results indicated that with the increase in the number of applications demanding real-time service, the fog computing platform outperforms the traditional cloud computing. The authors further observed that with 50% devices requiring real-time services, the service latency of fog computing decreases by 50%. However, an interesting observation by the authors was that the environments where there are less percentage of applications that demand low latency services, the fog computing appeared to be an overhead over traditional cloud computing. The evaluation parameters utilized by the authors were power consumption and service latency. Power consumption further included consumption due to data forwarding, computation, storage, and data migration. Whereas the service latency was subdivided into transmission latency and processing latency.

In summary, edge computing has been investigated as a motivation for improving energy efficiency of cloud applications. The

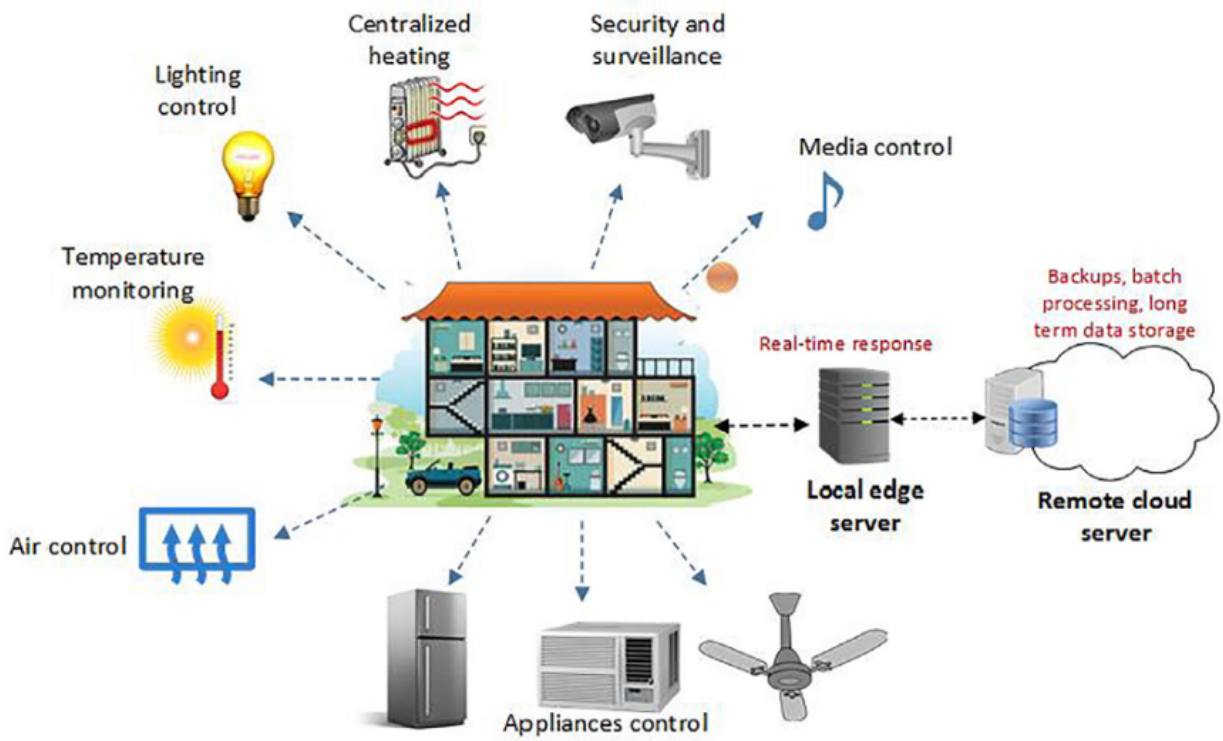

Fig. 9. Smart home using edge computing.

existing literature demonstrated that cloudlets and MEC can reduce the energy consumption for some cloud-based applications through computation offloading [164–166]. The joint optimization of radio and computational resources for multi-cell MEC helps minimizing energy consumption under latency and power budge constraints [167,168]. If properly deployed, the edge computing can augment cloud computing to reduce energy and response time of various cloud applications [51,99]. The delay and energy consumption are investigated together in various proposals to find a balance between the two to improve overall energy efficiency with minimal delay [100,101,66].

3.1.4. Smart living

Communication (delay sensitive) and interaction among smart objects, such as sensors, controllers, and actuators, is a pivotal and common phenomenon in all domains of smart living and pervasive environments [102]. Smart objects and Cloud computing interaction model, used in various smart solutions, such as Cognitive Gateway, depict various limitations and deficiencies in cloud interaction, specifically unpredictable delay and jitter [102]. Edge technologies offer the solution for these problems that hinder the visions and performance of smart living solutions. Advancement in smart devices and sensors are leading to fulfill the smart living visions. Smart Energy, Health, Offices, Protection, Entertainment, and Surroundings (EHOPES) represent the fundamental components of smart living. Fig. 9 reflects the use of edge computing in smart homes. Authors in [102] presented a generic fog model for smart living. The authors represented the fog architecture comprising of 3 major components.

- Fog Edge Node (FEN) is hardware component of fog architecture, which lies near or in close proximity to the smart objects, such as a smart phone, PCs, access points, set top boxes, located at one-hop proximity. FENs act as the endpoint of this fog architecture. FEN may perform basic processing, storage, and information filtration. The significance of FEN lies in providing various access methods (wired or wireless) to smart objects such as sensors and actuators, e.g., Bluetooth, ZigBee, Wi-Fi, etc.
- Fog Server (FS) represents the fog instances placed inside a micro-data center or cloudlet, representing a powerful virtualized server, offering inter-play between FEN and cloud. FS can offer processing power storage, and be used to take collective decisions based on information from various servers of FEN in smart environments. FS sits between FEN and cloud servers and offers required processing, storage, collective control, and updating of information. Smart objects may communicate directly to FS bypassing PEN in various models depending on the smart objects capabilities and requirements.
- Foglet is a middleware program agent, installed on fog nodes (FEN and FS) for dynamic and scalable services. Foglet offers provision for catering the heterogeneity in smart objects, applications, network management, and protocols. High level of privacy and security may be achieved using customized Foglets employing security techniques between FEN and FS. FEN lies in close proximity of smart objects and client, and has negligible privacy concerns, thus using custom privacy and security procedures to diminish the eavesdropping or leakage of information between FEN and FS, or even FEN and cloud.

Smart living and new IoT paradigms call for smart interactions among objects and ability to take decision based on presumed information. Authors in [40] proposed to use smart agents to mitigate lack of intelligence and reasoning in current smart objects in IoT and smart environments using swarm intelligence. They proposed Rainbow, an architecture for smart multi-agent system using fog computing. The Rainbow is a three layered architecture, having the physical "things" (smart objects), such as sensors and actuator constituting first layer. The intermediate layer, a middleware represented these things as Virtual Object (VOs), acting as an

intelligent agent exposing an abstract representation of smart object or thing. The VOs are coupled in computations nodes, named as Gateways. Different VOs or agents in a gateway may work together to achieve some high level goal. The gateways perform required processing and only the fine-grained agent process is sent to cloud server (which constitute the third layer of Rainbow architecture), leading to filtered and processed information to be executed on cloud servers. The authors detailed the design of three smart city applications, i.e., cyber physical system (CPS) for catering noise pollution, CPS for drainage network, and smart streets.

Sneppe and Namiot proposed to use mobile edge computing to share data among interoperating services of smart city [103]. As various smart city services use data from multiple sources, therefore, an edge based storage to store and receive local data enhance service efficiency and minimizes latency and core traffic. Taleb et al. presented Follow-Me-Edge (FME) to enable emerging services for smart living using mobile edge computing [104]. The FME is an extended version of Follow-Me-Cloud (FMC) concept [105,106] for edge computing to enable low latency services. The idea is to enable service to keep track of user and always service user from nearest edge service. The authors discussed the FME service using case studies, where a user watching a video while riding a bus is served by an edge location say Edge A. As the user is mobile, and gradually moves near to Edge B, the video and related streaming virtual function are migrated from Edge A to Edge B, so that the user may be served from Edge B. The authors presented the FME architecture and performed simulation based evaluation to depict live migration latency.

With the evolution of intelligent transportation system (ITS), a large number of sensors are deployed in city premises that collect traffic data on daily basis [107,108]. The live streams captured through video cameras require real-time processing and minimal latency, and therefore, the information cannot be sent to the traditional cloud as the response time will be higher. Edge devices embedded with traffic lights constitute smart traffic lights that receive real-time traffic information and coordinate among each other to create a dynamic green wave or to send warning signals in case of any road emergencies [155]. For instance, a camera mounted on a signal can detect flashing lights of an approaching ambulance and switch the street lights to allow free movement of ambulance through the intersection. Edge connected wireless access points can allow vehicle to vehicle, vehicle to access point, and access point to access point communications and numerous other applications, thus allowing information transfer and sharing among moving vehicles with minimum latency. For instance, traffic light system in Chicago, USA, is controlled with the help of smart sensors and edge computing [109]. Traffic volume data is collected from individual traffic lights. The IoT enabled smart traffic application computes real-time traffic congestion at network edge and automatically alter the timings of traffic signals, thereby allowing the smooth flow of vehicles.

A potential application of edge computing is smart grid. Smart grid constitutes smart meters, smart appliances, renewable energy sources, and energy efficient resources, as reflected in Fig. 10. The energy load balancing and distribution applications running on smart grid require real-time processing and actuation capabilities. The data generated by grid sensors and devices is processed at the edge servers, and filtered out to be consumed locally or sent to the higher tiers for visualization, reporting, and transactional analysis. In this way, the edge computing reduces the amount of traffic that would be otherwise sent to cloud for analysis if the edge layer is not present. The long term reporting and business intelligent analytics are provided by cloud computing. Smart meters installation in households of USA has witnessed exponential growth from 6% in 2008 to 89% in 2012. It is estimated that in 2019, various homes and small businesses will be having around 19 million smart meters [110]. With 500,000 smart devices, Austin energy gathered around 100 Terabytes of data. Smart meters send power usage updates every 15 minutes. With millions of smart meters, this will result in huge data, demanding substantial storage and bandwidth resources. Considering the smart grid paradigm, with power devices connected, and exchanging information will further aggravate the needs. Authors in [24] presented an approach to use fog computing for smart grids. The authors introduced the concept of SmartLocalGrid (SLG) for communication between two micro-grids. SLG allows communication among multiple devices efficiently to enable data processing and real-time decision locally without cloud support. Use of fog computing to mitigate the limited bandwidth capacity of Power Line Communication (PLC) is discussed in [111]. The authors proposed a distributed data aggregation and processing of consumer smart meters using fog computing. The simulation results depicted a great improvement in latency and response time when and intermediate fog layer is used for smart grid.

To summarize, edge computing finds its applications in energy, health, offices, protection (security), entertainment, and surroundings – the factors that constitute smart living. As we saw above, the recent works proposed: (a) fog enabled models for smart living to reduce latency and response time [102,104], (b) multi-agent based architectures to induce intelligence in smart living objects [40], (c) data sharing models for interoperating services in a smart city, (d) smart traffic control systems for controlling traffic lights using edge computing [109], and (e) models for smart power distribution using edge computing in smart grids [24,111].

3.1.5. HealthCare

Edge computing paradigms are foreseen to play significant role in eHealth care solutions and smart health [112]. Pervasive health monitoring applications are widely growing area of biomedical research offering various novel healthcare solutions, where most of the solutions are rooted in cloud computing. However, real-world user experience for these cloud based smart healthcare applications is unsatisfactory and poor because of the long delays and response times between application and cloud [112]. Edge computing technologies portrays great potential as a viable solution for pervasive healthcare applications to elevate the user experience and minimize delay [112]. A recent analysis of an eHealth application shows that around 25,000 tuples of health data flows every second, which will increase with the proliferation of IoT and smart city implementations to millions of tuples [33]. Several solutions for Chronic Obstructive Pulmonary Disease (COPD) patients' assistance are proposed using Fog computing, which enable patients to roam and move freely with the automated provision of breathing and oxygen supply [113,6]. Such assistance system will save patients from health deterioration and hospital expenses. Patients with COPD require assistance with the amount of oxygen required in various stages, such as during rest or walking. COPD breath assistance system employs the idea of constant patient state monitoring using BAN sensors. The required amount of oxygen depends on the arterial blood gas measurements. The extracted information is sent to fog instances, which calculate the exact amount of oxygen required by patient. The actuators on oxygen supply devices and cylinders react to the processed information and start supplying the required amount of oxygen. The oxygen supply does not depend only on the patient's physical conditions, rather it encompasses various parameters, such as patient condition, air pollution, and air quality. Fratu et al. employed fog computing to eWALL EU project to achieve real-time response for Mild Dementia (MD) and COPD patients [6]. eWALL offers a prefabricated system with various sensors to monitor various vital signs and habits of MD and COPD patients [6].

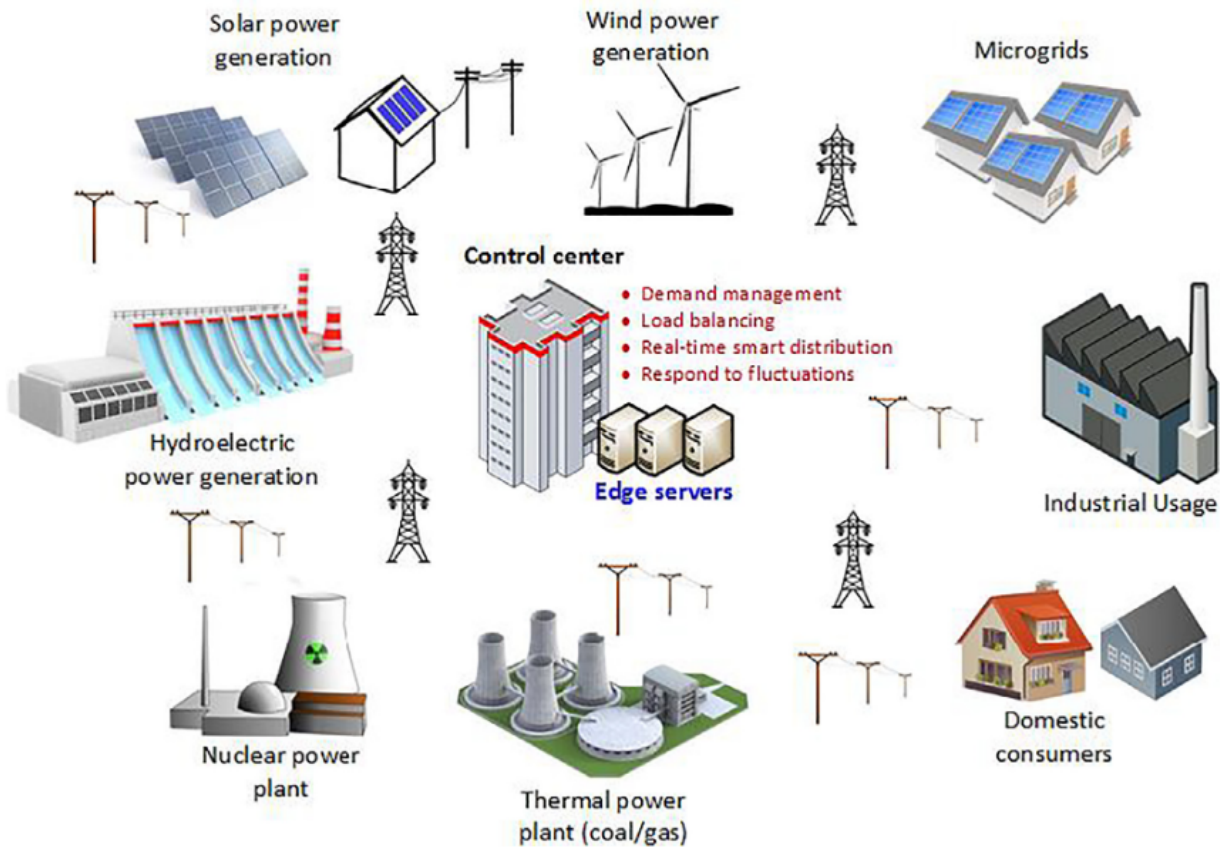

Fig. 10. Smart grid using edge computing.

Ambient Assisted Living (AAL) may play a key role to elevate senior citizens' life style and independence [6]. Real-time processing of information gathered from sensors is one of the key attributes of AAL system. Previously, such real-time processing for delay-sensitive applications was inconvenient and difficult. Edge technologies, like fog computing now offer the required processing capabilities in real-time for AAL systems. Cao et al. [112] propose FAST, a distributed analytics based fall monitoring system using fog computing for stroke mitigation [112]. Around one third of stroke mortalities may be prevented if stroke related risk factors, such as falling may be efficiently mitigated. Authors proposed new fall detection algorithms based on analysis techniques for non-linear time series and acceleration magnitude values, along with filtering techniques.

With the emergence of wearable computing and gadgets, cognitive assistance based application are becoming a reality. One of the major requirements of cognitive assistance applications is real-time response. Human subjects take from up to maximum 620 ms to respond an unknown face [92]. The edge computing can be used for real-time cognitive assistance by integrating image capturing, sensing, and processing to deliver response instantly. More than 20 million Americans suffer from some form of cognitive impairment, for whom, edge computing offers hope and a feasible platform. Ha et al. presented an architecture and prototype implementation of a cognitive assistance system using cloudlets and Google glass in [54]. The authors detailed the architectural requirements of cognitive assistance systems and implementation. Considering the high delay, cloud platforms cannot be used for task offloading, therefore, authors used cloudlet for efficient communication and processing. Similarly, Chen et al. presented an architecture and implementation details of a wearable cognitive assistance application using cloudlets [5]. The application is designed for cognitive assistance in four different tasks.

Quwaider and Jararweh proposed cloudlet based architecture for collection and processing of data from Body Area Networks (BANs) [114]. Authors employed cloudlets to minimize packet-to-cloud energy and packet delay. The authors simulated the proposed architecture using CloudSim simulator to illustrate energy efficiency and low latency. Amraoui and Sethom proposed cloudlet based pervasive healthcare monitoring system for chronic diseases using BANs [115]. The authors presented a new architecture using SDNs and cloudlets for fast communication and analysis, and to handle heterogeneity in device and access networks. Althebyan et al. [116] presented a largescale e-health system using edge technologies. The authors proposed wearable textile based sensor, strategically distributed in clothing to continuously monitor patients' health condition. The sensed information from sensors is transmitted to a handheld device, such as a smart phone or tablet. The handheld device forwards the information to cloudlet to process and take necessary action. In case of any abnormality, instructions are alarmed on the handheld device along with other necessary actions, e.g., automated call to ambulance service along with GPS location of patient for instant medical assistance. The authors simulated the proposed model using CloudExp simulator to evaluate the scalability.

In summary, the edge computing has been employed in healthcare to meet the real-time response requirements of applications. We have seen that edge computing plays a pivotal role in health applications for: (a) COPD patients that require real-time oxygen monitoring [113], (b) the monitoring of patients suffering from mild dementia [6], (c) fall detection and stroke mitigation applications [112], (d) cognitive assistance systems including Google glass [54,5], and (e) BANs based pervasive healthcare [114–116]. All such

applications require real-time response, so cloud computing can be augmented with edge to perform computations near to the end users to reduce the latency and response time.

3.1.6. Communication efficiency and edge computing

5G aims to offer minimal latency as compared to 4G to its users [117]. Three distinct objectives of 5G include: (a) pervasive connectivity, (b) millisecond latency, and (c) gigabit connection [12]. Various 5G applications will use cloud support, however, communication delay and latency in far located cloud resources may pose a major barrier to achieve one of the pivotal goal of 5G networks, i.e., extremely low latency. Therefore, fog computing offers a realistic solution to minimize latency [117]. Intharawijitr et al. [117] analyzed fog computing in 5G mobile networks paradigm for communication and computation latencies. The authors presented a mathematical fog model for 5G networks. The authors evaluated their model using simulations to measure the impact of user demand, and current load in fog servers on computation and communication latency. In [118] Peng et al. presented the suitability and benefits of using edge computing paradigm in 5G networks and proposed Fog- Radio Access Networks (F-RAN) to mitigate the shortcomings of Cloud Radio Access Networks (CRAN). The authors presented FRAN architecture to use FRANs for radio resource management, signal processing of local radio, and distributed storage tasks. Performing these radio related tasks on fog nodes will aid to alleviate front haul burden and avoid centralized baseband unit based signal processing, leading to realization of minimal latency being one of the key goals of 5G.

Nunna et al. presented various use cases for potential context-aware collaboration systems using 5G technology with Mobile edge computing [119]. The authors presented a remote robotic tele surgery scenario, where surgeons remotely direct surgical robots to perform surgery. Another case study of a road accident is described where the application automatically calls the ambulance along with defining clear path for the ambulance by redirecting traffic automatically by using traffic signal. Such content-aware systems demand less than 10 ms response time, which is not possible without using 5G with edge technologies [119].

To summarize, edge computing can improve communication efficiency and reduce latency in 5G networks by bringing frequently accessed resources closer to the end user. As we saw above, the existing solutions try to balance user demand and load in terms computation and communication on edge servers [117]. The radio resource management using edge computing alleviate front haul burden resulting in reduced latency for 5G networks [118]. Several collaborative applications using 5G technologies, such as remote surgery and automatic emergency response, can also take benefit of edge computing platform to reduce their overall latency [119].

3.2. Edge computing architectures and evaluation

Edge computing is in its infancy and currently lacks a standardized architecture, protocols, interoperability and communication patterns, and resource management. Some of the generic architectures have been proposed in literature. Moreover, new implementation and evaluation mechanisms are also mentioned. Some of the key architectures and implementation details are presented below.

3.2.1. Edge computing architectures and resource management

Edge computing follows a three-tier architecture in general, comprised of end device, edge layer (fog, cloudlet, MEC, MDC), and cloud data center. Zhang et al. presented a multi-tiered architecture for delay sensitive cloud Data Service Subscribers (DSS) [120]. Three considered tiers are DSS, Massive Data Centers (MDCs), and Fog instances. Resource management is achieved using Game programming. Multi-leader, multi-follower Stakelberg games are used for interaction between fog and MDCs, and single-leader single-follower Stakelberg game between DSS and MDCs. One of the major contributions in the paper is to consider the competition among various fog instances and MDCs. Azam and Huh [121] presented an architecture of a smart gateway with fog computing. The proposed architecture had several layers. The physical and virtualization layer manage the physical nodes, virtual nodes, virtual sensor networks, and WSNs as per the system requirements. Monitoring layer monitors networks and activities of underlying nodes and also monitors which node is performing what task at what time and what are current and future requirements. Monitoring layer also considers the energy consumption and remaining energy of nodes to take preemptive measures on time. Preprocessing layer performs data filtering, trimming, and other data management tasks so that only the necessary, and more meaningful data is generated. The transport layer uploads the preprocessed and filtered data to the cloud, thus putting least burden on core network. Using a testbed, the authors evaluated the performance of their architecture by analyzing the communication between gateway and the cloud. The performance parameters utilized during testing were upload delay, bulk-data upload delay, synchronization delay, and bulk-data synchronization delay.

Yin et al. [122] proposed Tentacle, a dynamic and on the fly resource provisioning algorithm to procure edge servers for online service providers. The framework identifies the best location based on the users' proximity and service requirements considering the Network Coordinate (NC) system based ranking. The edge location is identified as a tuple represented as $\langle city, Autonomous\ System\,(Ad)\rangle$. The tuple may further be classified based on the multiple server clusters available at the edge location as edge site. The authors extended the procurement of edge servers beyond ISPs and Internet Exchange Points (IXP) to CDN provider micro data centers, such as Akamai, LimeLight, and EdgeCast.

Azam and Hu presented a service oriented strategy to effectively and efficiently manage resources in fog computing [26]. The authors considered a customer based resource estimation model considering various traits of customers. In their model, Cloud Service Customer (CSC) and fog negotiate for resource requirements to provide specific services and SLA. Based on the required service and agreed SLA, resource requirements and advanced allocations are estimated. Service requests are generated by smart objects. Therefore, appropriate prediction and resource pre-allocation are essential for efficient and fair service delivery. The authors formulated resource estimation, and evaluated their model using simulations. The authors extended their work in [27] by categorizing IoT devices based on devices' mobility and nature to efficiently perform resource allocation. A detailed pricing model was also discussed in the extended work [27]. Do et al. proposed a resource allocation algorithm to optimize the traffic distribution between fog and data center for video streaming applications [123].

A service oriented model for resource management within IoT devices is proposed by Aazam et al. [26] that utilized fog computing for fair management of resources. Given with user requirements and characteristics, the proposed work addresses the issues related to resource management, such as resource prediction, resource estimation, advance reservation, and pricing. The implementation of the proposed work is performed in java, whereas the model was evaluated with CloudSim simulator.

Zeng et al. [126] proposed a fog computing supported software-defined embedded system (FC-SDES). The authors investigated task scheduling problem in FC-SDES. The proposed system consists of edge devices (cellular base stations) equipped with computation and storage resources and embedded client systems are general

purpose hardware. A computation task in the proposed system can be processed either at client side or edge side. Initially, a task image is not fully loaded into embedded system, rather the image resides in the edge server and client retrieves the image from the edge server at runtime. By balancing workload on both sides, the system tries to minimize overall computation and transmission latency of the requests. The proposed system also investigates the replica placement problem of task image on storage servers. The task completion minimization problem is formulated using mixed-integer nonlinear programming (MINLP) considering the task scheduling and task image placement constraints. The results indicated the decrease in maximum task completion time in FC-SDES.

Edge servers' provisioning is generally planned, however, in some cases, edge provisioning may be on the fly and dynamically consider the flash crowd or requirements of users within a specific area, e.g., in an emergency response situation. Dynamic edge service provisioning and its prospects are discussed in [122]. Some of the factors that may be considered to choose a situation to dynamically provision edge resources, or select an appropriate servers' position are:

- User demand for a specific service.
- Nature of service, e.g., delay sensitive or real-time.
- Current average response time or RTT for users and current average cost of service.
- Benefits of provisioning edge servers to deploy the service near to users in terms of delay and cost.
- Complexity of provisioning of edge servers and deploying the service on the fly.
- Nature of the service and required capacity.
- Average distance of the provisioned resource from the users.

Some situations, e.g., emergency response systems, sometimes mandate the provisioning of service nearest to the users considering the nature of service and situation. Appropriate service deployment schemes, e.g., Foglets-based ready to deploy middleware setups (see Section 3.1) may be used to install the service on provisioned servers to start service immediately.

3.2.2. Edge computing implementation and simulation

Edge computing is an emerging area, so little implementation, testing, and simulation solutions are available. Some authors implemented fog models using various distributed systems API, such as Go and Constrained Application Protocol (CoAP) [127]. CoAP is a light weight protocol using User Datagram Protocol (UDP), to be used as a reference protocol for IoTs and Web of Things (WoT). CoAP specifically targets resource constrained IoT devices and alleviates various overheads imposed by the HTTP protocol. Go programming language by Google is a statically typed, concurrency and garbage collection enabled language developed at Google. Go was developed to aim large distributed systems considering scalability [128].

Cirani et al. [129] proposed an architecture of Fog nodes as an IoT hub using CoAP protocol. Fog node can be placed at the edge of the network to interact with multiple physical IoT networks. Fog node implements various protocols and act as CoAP server to perform various functions, such as border router between various resource constrained IoT networks, perform resource and service discovery, act as resource directory, CoAP and HTTP gateway for intercommunication, and caching. The authors implemented the Fog node using Californium, which is a java based CoAP implementation. The authors deployed and evaluated Fog node using Model B of Raspberry Pi (RPi) single board computer. Butterfield evaluated Google's Go language for IoT and fog scenario [128]. The results depicted that Go language can be used to implement fog architecture for IoT solutions. The authors used RPi for prototype implementation and detailed simulation for evaluation and comparison with Cirani et al. [129] fog implementation. The authors demonstrated the suitability of Go language for fog implementations.

Sarkar and Misra [25] presented theoretical modeling and mathematical formulation of fog computing architecture considering its various components. The authors carried out a comparative analysis of fog model with cloud computing considering service latency and energy consumption. Results depicted that in a scenario, when even a portion (25%) of IoT devices enjoy real-time services from fog, the energy consumption is 40% less in fog as compared to when devices are served from cloud computing. The formulation considered some realistic assumption to formulate the fog architecture, which are: (a) terminal nodes (IoT devices) are aware of their geo-spatial location, (b) fog tier is comprised of multiple intelligent devices with processing, routing, and storage capacities, (c) devices in fog tier can communicate and share computational and network load, and data, and (d) fog devices offer mobility support to terminal devices. The Terminal Node (TN) is represented by six-tuple comprised of, Id, status, type, location, hardware specifications, and an array containing ids of all applications running on device. A fog device is represented by a three-tuple comprised of, device Id, device type, and device specifications. Multiple fog devices clustered together constitute a Fog Instance (FI), which is represented by three-tuple comprised of FI Id, access point Id through which FI is connected to cloud, and an array representing all of the fog devices currently connected to FI.

Gupta et al. presented iFogSim, a simulation environment focusing on evaluation of resource management strategies for fog computing [130]. The simulator evaluates the impact of resource allocation on energy consumption, latency, operational cost, and network congestions. The performance metrics are calculated by simulating edge devices, cloud data centers, and the links interconnecting edge devices to data centers. A Sense-Process-Actuate application model was mainly considered for simulation. The iFogSim was built on Cloudsim, where communication is performed by passing messages or sending events, so no real network traffic is simulated. Therefore, fine-grained network details, accuracy, or realistic communication results and latencies cannot be achieved [131].

In [132], Orsini et al. proposed a MEC based programming framework CloudAware that allowed the users to offload their compute-intensive tasks from smartphones to the edge servers. This facilitates the users to speed-up the execution and develop scalable and elastic mobile applications for mobile edge. Cisco's ParStream is a platform that allows handling of massive volumes of high-velocity data to provide real-time analytics at the edge [133]. The ParStream works 20 times faster than an average database, and utilizes complex compression and indexing capabilities to provide large scale, faster data access. ParStream continuously analyzes real-time IoT data as it is loaded and can perform spontaneous querying. Vortex [134] is an intelligent data sharing and analytics platform for business critical IoT applications. Vortex fog computing provides platform independent interoperable solutions for embedded, mobile, and enterprise environments, thereby targeting areas such as healthcare, energy, transportation, and industrial automation. Cisco Data in Motion (DMo) technology allows data management and analysis of large volumes of data coming through IoT at the edge [53]. The DMo is based on extensible, scalable, and modular architecture and is designed to capture real-time data and control flows, translating data into information for use by higher order applications within the system [53]. Cisco IOx [135] is an application environment that is a combination of Cisco IOS, a network operating system, and Linux. The IOx allows hosting capabilities for fog applications, and allows management of network components, such as routers, switches, and compute mod-

ules. Moreover, the IOx provides open-source tools to allow developers create applications that execute on Cisco IoT infrastructure. With IOx, the fog applications can communicate with IoT devices via M2M protocols, and can send data to the cloud by translating non-standard protocols to IP.

Edge computing can be exploited to enable smart e-commerce. In a traditional setup, a customer performing online purchase may need to update shopping cart many times before performing the checkout. However, all the updates made to the cart need to be sent to the cloud. The edge supported e-commerce website will allow the update of shopping cart at a local, nearby edge, and when the customer will perform the final checkout, the updated information will be sent to the cloud only once, thereby reducing the traffic on cloud. Nippon Telegraph and Telephone Corporation (NTTC) developed an edge accelerated web platform (EAWP) [124]. The EAWP enables edge support for web applications. The user's device is relieved of processing the whole application, as the loads are distributed close to the user on the edge servers. Such mechanism allows the high-speed execution of web applications, even when the end user device has limited resources to run the application. The load distribution and data transfer is optimized considering the user context. Various experiments conducted on EAWP revealed significant reduction in cloud application response time, and NTTC reported a reduction by a factor of 100 at most. The proposed EAWP allows any other web applications to run on its execution environment conforming to the traditional HTML standards without any reprograming requirement. Zhu et al. [125] proposed the concept of fog boxes to improve the website experience. The users connect with the internet via edge servers (fog boxes) using HTTP. The fog boxes perform various optimizations to reduce latency.

For instance, the fog boxes perform caching of the content and reduce the size of HTML objects in case the network is slow. In case of network congestion, the fog boxes reduce the graphics' resolutions, thereby maintaining the acceptable response times for end users.

To summarize, we presented numerous state of the art architectures, implementations, and simulation models/platforms for edge computing. It can be observed from the above discussion that because of a relatively new technology and in its evolutionary phase, the edge computing lacks any standard architecture and simulation platform. Most of the above discussed architectures and implementations are specialized, i.e., they target specific application scenarios and with aims to optimize various parameters, such as latency, response time, and energy consumption, etc. Moreover, there is yet to exist a complete simulation platform for edge computing that can be configured with numerous architectures and edge based pricing models. In Table 3, we present a summary of various state of the art edge-based architectures, implementations, and simulations. Next section presents the various challenges faced in adoption of edge computing technologies.

4. Edge computing challenges

Being a new technology, edge computing faces numerous challenges, in addition to the challenges it inherits from cloud computing. Most of the challenges faced by edge computing is due to the non-standardization of edge technologies. In this section, we highlight some of the important challenges of edge computing that pave the path for future research in the edge technologies. Table 4 presents a summary of research works in the areas selected.

4.1. Resource management and allocation

Some generic edge architectures and resource allocation and management mechanisms have been proposed in the literature. However, in-depth and detailed analysis and testing is still missing. In [117], a comparison of computation and communication latency is presented. It has been observed that computational latency may increase based on the current load of the fog server. Therefore, appropriate resource management and allocation is a key. For time critical and real-time system, priority-aware computation is required in Fog instances. Delay sensitive tasks may be marked as high priority, and fog node needs to handle such requests immediately. Appropriate cost model also needs to be formulated and designed, where extra charges may be received from priority jobs in case of high load. Cloud providers already depict such cost model, where the cost incurred during peak hours is different from off-peak hours. For instance, Amazon costs different for scheduled Reserved Instances in peak and off-peak hours [136]. No standard, detailed, and realistic design is available to standardize edge computing architecture, resource management, cost models, and interaction with cloud [65]. Some of the initial and simple architectures have been proposed, however, they lack practical implementation and resource management aspects [65]. One of the challenges is to identify the edge provisioning site based on the dynamic number of users and application demands. Edge site provisioning mandate two basic requirements: (a) good proximity between users and edge servers and (b) sufficient capacity to serve user demands [122]. Workload allocation is another challenge because of the complex real-time decision making involved about putting how much workload on each edge layer. If too many edge layers are involved, the latency may reach cloud's latency, or even greater. The workload allocation strategy needs to balance various conflicting objectives, such as latency, bandwidth, energy, and cost. Some metrics need to be prioritized over others, and optimization must be performed dynamically making the task challenging.

4.2. General purpose computing in edge

An important consideration is to enable General Purpose Computing (GPC) in edge technologies. For instance, base stations are equipped with customized Digital Signal Processors (DSPs) for specific tasks and workloads, which are unsuitable for GPC [35]. Moreover, base stations are not considered suitable because of cost and architectural concerns to be used for GPC [64]. Furthermore, existing edge computing software solutions, such as Cisco IOx [135] and Nokia software solution [137] are specialized solutions designed from specific hardware and are unsuitable in heterogeneous edge environments. Considering general purpose processors to be used in base stations or routers or other edge devices require heavy investments and may pose performance concerns [35]. Being in its infancy, there are very limited insights in edge to cloud interaction models. Edge resources or services may act as a proxy on behalf of user, or act as a forwarding agent in case edge is unable to service the users' request. As edge instances also cache data, therefore, appropriate models for synchronization and updates is also required. The programming models for edge will require task and data level parallelism to support real-time applications. The programming languages involved in programming models will need to take into account the diversity and heterogeneity of devices. These requirements are different from traditional cloud computing where the use cases are well defined and most hardware and software frameworks are compatible.

4.3. Security and privacy

One of the major challenges in deployment and adoption of edge computing paradigm is privacy and security. Edge computing implementations in terms of size and investments are way smaller than cloud infrastructure and way more in terms of number and

Table 3
Summary of edge computing architectures, resource management, evaluations, and simulations.

Area/Application	Reference	Idea presented
Edge computing architectures	[120]	Zhang et al. presented a multi-tiered architecture for delay sensitive cloud Data Service Subscribers (DSS). Three considered tiers are DSS, Massive Data Centers (MDCs), and Fog instances.
	[121]	Eui-Nam et al. presented an architecture of a smart gateway with fog computing. The proposed architecture had several layers, such as physical and virtualization layer, monitoring layer, preprocessing layer, transport layer
	[126]	Zeng et al. proposed a fog computing supported software-defined embedded system (FC-SDES). Task scheduling problem is investigated in FC-SDES. The system consists of edge devices equipped with computation and storage resources and embedded client systems are general purpose hardware.
	[129]	Cirani et al. proposed an architecture of Fog nodes as an IoT hub using CoAP protocol. Fog node can be placed at the edge of the network to interact with multiple physical IoT networks.
	[25]	Sarkar and Misra presented theoretical modeling and mathematical formulation of fog computing architecture considering its various components. The authors carried out a comparative analysis of fog model with cloud computing considering service latency and energy consumption.
Edge computing resource management	[120]	Resource management is achieved using Game programming. Multi-leader, multi-follower Stakelberg games are used for interaction between fog and MDCs, and single-leader single-follower Stakelberg game between DSS and MDCs are implemented.
	[122]	Yin et al. proposed Tentacle, a dynamic and on the fly resource provisioning algorithm to procure edge servers for online service providers. The framework identifies the best location based on the users' proximity and service requirements considering the Network Coordinate (NC) system based ranking.
	[26]	Azam and Hu presented a service oriented strategy to effectively and efficiently manage resources in fog computing. The authors considered a customer based resource estimation model considering various traits of customers. In their model, Cloud Service Customer (CSC) and fog negotiate for resource requirements to provide specific services and SLA.
	[27]	Aazam et al. categorized IoT devices based on devices' mobility and nature to efficiently perform resource allocation. A detailed pricing model was also discussed
	[123]	Do et al. proposed a resource allocation algorithm to optimize the traffic distribution between fog and data center for video streaming applications
	[26]	A service oriented model for resource management within IoT devices is proposed that utilized fog computing for fair management of resources and addresses issues related to resource management, such as resource prediction, resource estimation, advance reservation, and pricing.
	[122]	Dynamic edge service provisioning and its prospects are discussed in this proposal.
Edge computing evaluation	[121]	Eui-Nam et al. used a testbed to evaluate the performance of their architecture by analyzing the communication between gateway and the cloud. The performance parameters were upload delay, bulk-data upload delay, synchronization delay, and bulk-data synchronization delay.
	[128]	Butterfield evaluated Google's Go language for IoT and fog scenario. The results depicted that Go language can be used to implement fog architecture for IoT solutions. The authors used RPi for prototype implementation and detailed simulation.
Edge computing implementation	[124]	Nippon Telegraph and Telephone Corporation (NTTC) developed an edge accelerated web platform (EAWP). The EAWP enables edge support for web applications. The user's device is relieved of processing the whole application, as the loads are distributed close to the user on the edge servers.
	[26]	A service oriented model for resource management within IoT devices is proposed by Aazam et al. The implementation of the proposed work is performed in java, whereas the model was evaluated with CloudSim simulator.
	[129]	Cirani et al. implemented the Fog node using Californium, which is a java based CoAP implementation. Fog node implements various protocols and act as CoAP server to perform various functions, CoAP and HTTP gateway for inter-communication, and caching.
	[132]	Orsini et al. proposed a MEC based programming framework CloudAware that allowed the users to offload their compute-intensive tasks from smartphones to the edge servers. This facilitates the users to speed-up the execution and develop scalable and elastic mobile applications for mobile edge.
	[133]	Cisco's ParStream is a platform that allows handling of massive volumes of high-velocity data to provide real-time analytics at the edge.
	[134]	Vortex fog computing provides platform independent interoperable solutions for embedded, mobile, and enterprise environments, thereby targeting areas such as healthcare, energy, transportation, and industrial automation.
	[53]	Cisco Data in Motion (DMo) technology allows data management and analysis of large volumes of data coming through IoT at the edge. The DMo is based on extensible, scalable, and modular architecture and is designed to capture real-time data and control flows, translating data into information for use by higher order applications within the system.
	[135]	Cisco IOx is an application environment that is a combination of Cisco IOS, a network operating system, and Linux. The IOx allows hosting capabilities for fog applications, and allows management of network components, such as routers, switches, and compute modules.
	[124]	Nippon Telegraph and Telephone Corporation (NTTC) developed an edge accelerated web platform (EAWP). The EAWP enables edge support for web applications. The user's device is relieved of processing the whole application, as the loads are distributed close to the user on the edge servers.
	[125]	Zhu et al. proposed the concept of fog boxes to improve the website experience. The users connect with the internet via edge servers (fog boxes) using HTTP. The fog boxes perform various optimizations to reduce latency.
Edge computing simulation	[130]	Gupta et al. presented iFogSim, a simulation environment focusing on evaluation of resource management strategies for fog computing. The simulator evaluates the impact of resource allocation on energy consumption, latency, operational cost, and network congestions.

granularity. The organizations offering small edge computing solutions belonging to small businesses are less interested in investing in security and privacy infrastructure [138]. Moreover, at the core of the edge computing are the enabling technologies, such as peer-to-peer systems, wireless networks, distributed systems, IoT, and virtualization platforms. To secure edge, all the aforementioned technologies must be secured while keeping in consideration that the interoperability and integration of devices should not be compromised. Addressing the security of cloud is comparatively easier than edge because of the centralized nature and a single controlling authority of cloud paradigm. In edge, the services migrating from one device to other, or from one edge to another edge de-

Table 4
Summary of edge computing work in various areas as presented in Section 4.

Area/Application	Ref.	Idea presented
Resource management	[117]	A comparison of computation and communication latency is presented. It has been observed that computational latency may increase based on the current load of the fog server. Therefore, appropriate resource management and allocation is a key.
	[157]	A cost efficient resource management scheme is presented for fog computing supported medical cyber physical systems (FC-MCPS). The base station association, task distribution, and virtual machine placement is investigated jointly. Problem is formulated into a mixed-integer non-linear programming and linearized to mixed integer linear programming.
	[158]	A hierarchical game framework for resource management in fog computing is proposed. A three-layer hierarchical game framework to solve challenges in fog computing is designed. Stackelberg sub-game is used for interaction between data server operators and fog networks. Matching sub-game is used for the interaction between fog networks and authorized data service subscribers.
	[159]	A fog computing structure presented along with crowd-funding algorithm to integrate spare resources in the network. Incentives mechanisms are implemented to encourage owners to share their resources.
General purpose computing	[160]	Designed SONM'S secure and cost efficient fog supercomputer for general purpose computing, from mobile app hosting to video rendering and DNA analysis. Users all over the world can leverage their idle computer power to become part of SONM network.
Security	[161]	Proposed a security technique called Encrypted Data Flow Mechanism (EDFM) based on the concept of Fog computing to secure cloud storage from unauthorized/illegal access. The simulated environment utilizes a fog data center called Broker to hide the actual cloud storage underneath it.
	[162]	Proposed various potential threats to IoT fog, and existing security measures to mitigate those threats.
Scalability	[163]	The paper proposed a fog computing paradigm that utilizes buses' network for service offloading. The bus based fog servers provide fog services to passengers, as well as perform computation offloading to road side cloudlets. Allocations are performed using genetic algorithm (GA).
Data abstraction	[121]	Azam et al. proposed a smart gateway architecture for cloud computing. The gateway performs data collection, preprocessing, filtering, and reconstruction of data into useful form, and uploads only necessary data to the cloud.

ployed by different vendors can create vulnerabilities, and security provisions in this regard are not widely studied. All the privacy concerns related data transfer from user to edge and from edge to cloud must be taken into account. The user devices may not be resourceful enough to run complex cryptography algorithms to encrypt the data. Similarly, the edge devices may consist of micro servers (e.g., fog made of raspberry pi computers), which may take longer to encrypt data, thus increasing the latency. More importantly, most of the end users usually are not aware of privacy and security, so there must be some automatic mechanisms ensuring run-time privacy of user data. For instance, in a survey, it was revealed that 80% of Wi-Fi users have their wireless routers set on default password out of 439 million subscribers, and 49% user networks are unsecured [7]. Moreover, 89% of public Wi-Fi hotspots are unsecured [7]. It is reported in [9] that by year 2020, 10% of all the attacks will target IoT systems. It is also critical to isolate a user's private data from other data collected by third party applications. For instance, an activity tracking application should not be able to access the electricity usage data of a user [7]. The specific data control access mechanisms should be implemented on edge frameworks to ensure data privacy.

Edge computing can also act as a middleware to secure the data at edge before sending it out to the Internet and cloud [65]. IoT devices being lightweight with limited battery, processing, and storage, are not suitable to perform security related tasks, such as encryption. However, security and privacy being one of the utmost concerns in Cloud of Things (CoT) paradigm, edge computing inherently can offer a middle tier to secure the data before sending to cloud, offering a convenient solution to resource constrained devices. Stojmenovic et al. [139] discussed the security and privacy issues of fog computing. The authors studied the effects of man-in-the-middle attack on fog computing, and discussed the consequences of this attack on CPU and memory consumption of fog devices. Rodrigo et al. [140] presented a detailed survey on security threats and challenges on mobile edge, and fog computing. LocalGrid's fog computing platform provides standardizations and secured end-to-end data communications from edge devices to cloud [141].

4.4. Scalability

Cloud computing utilizes resource from multiple data centers with tens to hundreds of thousands of serves. However, edge technologies are very high in number, and possess small number of computation and storage resources. Resource overprovisioning is infeasible because of the cost and energy concerns. Therefore, scalability and rapid resource provisioning is of significant importance. Considering the limited resources and delay sensitive services in edge computing domain, timely provisioning of resource to service the request is vital. In case of non-availability of resources, time critical applications, specifically services related to healthcare and emergency evacuation may have catastrophic impact. Similarly, user interactive and multimedia related applications cannot tolerate extra delay for waiting or request forwarding to nearest edge service. Therefore, necessary resource provisioning strategies, and priority based provisioning are required to be discussed by research community.

4.5. Data abstraction

One of the important challenges of edge computing is data abstraction. Data abstraction is the preprocessing and trimming of data at edge before sending the data to cloud. The IoT devices produce huge volumes of data. Sending such large datasets to cloud will lead to the congestion of both the backbone network and overburdening of datacenters.

The data should be preprocessed and filtered at edge device to remove noise, low quality data, and for privacy protection (by truncating the unauthorized data). However, the data abstraction impose several challenges. If too much trimming of data is performed, this may result in the loss of some useful information, thereby reducing the precision/accuracy of data. If data is subjected to less trimming, unwanted data may also be sent towards cloud causing extra burden on resources. Azam and Huh [121] proposed a smart gateway architecture for cloud computing. The smart gateway can either be directly connected with the IoT devices using single-hop link, or multiple IoT devices are connected with base stations and sink nodes, which in turn are connected with gateway. The gateway performs data collection, preprocessing, filtering, and recon-

struction of data into useful form, and uploads only necessary data to the cloud.

4.6. Fault tolerance and quality of service

Maintaining acceptable levels of QoS and fault tolerance is an important issue in edge computing. Due to distributed nature of edge, the existing methods of fault tolerance in cloud will not be applicable to edge computing. The edge is primarily designed for real-time applications, so the fault tolerance should be proactive and there must be automatic recovery from faults. The edge devices should not be overburdened so that the minimum level of QoS must be maintained. Therefore, a proper monitoring mechanism should be deployed that inspect the peak hour usage of edge nodes, thereby facilitating the task partitioning and scheduling in flexible manner. Another challenge in maintaining QoS in edge computing is when multiple edges are involved in collaboration, also known as collaborative edge [7]. For instance, such scenario may occur when a user moves from the area of coverage of one edge to another edge. In this case, the user data must be available on the other edge node. A solution to this issue is to cache user data on multiple edges in collaboration. However, this will raise the issue of increased traffic among participating edges. Therefore, optimal data placement and replication strategies needs to be designed that reduce the latency and traffic, within minimum acceptable thresholds of QoS.

5. Conclusions

This survey discussed in detail the emerging technologies and the state-of-the-art in edge computing and its various applications. With the multifold increase in IoT enabled devices and their applications, especially those that require near real-time response, the traditional cloud computing paradigm faces numerous challenges in terms of latency, scalability, and computation. The cloud data centers are deployed at far places due to which response time could be a few milliseconds to few seconds. Moreover, the user application may be generating large volumes of data to be sent to cloud that may cause significant overhead on backbone network. Edge computing is solution to the aforementioned problems, as it brings the computational and storage resources closer to the end user devices, and reduce burden on cloud. The edge computing technologies discussed in the survey are: fog, cloudlets, micro datacenters, and mobile edge. The aforementioned technologies have some basic differences, but they are all based on the same idea having similar objectives, i.e., to bring the computation and storage resources at the edge of the network near the end users. As a key contribution, a comprehensive list of the potentials, applications, architectures, and evaluations are also presented in the survey, along with the state of the art in the aforementioned. In the end, some of current challenges of edge computing are discussed.

As a future work, we aim to develop a simulator for edge computing that will allow users to model the three layers: (a) IoT layer, consisting of heterogeneous IoT devices, (b) edge layer, consisting of networked edge servers, and (c) cloud layer, connected with edge and backbone network. The users will be able to test their applications' efficiency in terms of response time, energy consumption, latency, and computational resources usage. We also aim to design and integrate a billing model with the simulator that will allow researchers to design and test their applications that generate maximum revenues at service provider's end with reduced price for customer without compromising the QoS parameters and given set of constraints.

Acknowledgement

This publication was made possible by NPRP grant # [8-519-1-108] from the Qatar National Research Fund (a member of Qatar Foundation). Samee U. Khan's work was supported by (while serving at) the National Science Foundation. The findings achieved herein are solely the responsibility of the author[s].

References

[1] K. Bilal, S.U.R. Malik, S.U. Khan, A.Y. Zomaya, Trends and challenges in cloud data centers, IEEE Cloud Comput. 1 (1) (2014) 10–20.
[2] K. Bilal, S.U.R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, U.S. Khan, A. Abbas, N. Jalil, S.U. Khan, A taxonomy and survey on green data center networks, Future Gener. Comput. Syst. 36 (2014) 189–208.
[3] K. Bilal, S.U. Khan, A.Y. Zomaya, Green data center networks: challenges and opportunities, in: 11th IEEE International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, December 2013, pp. 229–234.
[4] K. Bilal, S.U. Khan, S.A. Madani, K. Hayat, M.I. Khan, N. Min-Allah, J. Kolodziej, L. Wang, S. Zeadally, D. Chen, A survey on green communications using adaptive link rate, Cluster Comput. 16 (3) (2013) 575–589.
[5] Z. Chen, L. Jiang, W. Hu, K. Ha, B. Amos, P. Pillai, A. Hauptmann, M. Satyanarayanan, Early implementation experience with wearable cognitive assistance applications, in: Proceedings of WearSys 2015, Florence, Italy, May 2015.
[6] O. Fratu, C. Pena, R. Craciunescu, Fog computing system for monitoring mild dementia and COPD patients, in: 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2015, pp. 123–128.
[7] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.
[8] IDC FutureScape: Worldwide Internet of Things 2016 Predictions, https://www.idc.com/research/viewtoc.jsp?containerId=259856 Accessed: January, 2017.
[9] IDC FutureScape: Worldwide Internet of Things 2017 Predictions [online]. Available: https://www.idc.com/research/viewtoc.jsp?containerId=US40755816, Accessed: January, 2017.
[10] Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf, Accessed January, 2017.
[11] 4 Vs of Big Data, [online] Available: http://www.ibmbigdatahub.com/sites/default/files/infographic_file/4-Vs-of-big-data.jpg, Accessed: January, 2017.
[12] E. Borcoci, Fog computing, mobile edge computing, cloudlets - which one? 2016 Conference SoftNet, Rome, August 21, 2016.
[13] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Helsinki, Finland, 2012, pp. 13–16.
[14] F. Bonomi, R. Milito, P. Natarajan, Jiang Zhu, Fog computing: a platform for internet of things and analytics, in: Book Chapter, *Big Data and Internet of Things: A Roadmap For Smart Environments*, in: Studies in Computational Intelligence, vol. 546, Springer, March 2014, pp. 169–186.
[15] M.T. Beck, M. Werner, S.F.L. Maximilian, T. Schimper, Mobile edge computing: a taxonomy, The Sixth International Conference on Advances in Future Internet, 2014.
[16] G.I. Klas, Fog computing and mobile edge cloud gain momentum, Open Fog Consortium, ETSI MEC and Cloudlets, 2015.
[17] V. Avelar, Practical Options for Deploying Small Server Rooms and Micro Data Centers, Revision 1, White Paper [online] Available: http://www.datacenterresearch.com/whitepaper/practical-options-for-deploying-small-server-rooms-and-micro-9014.html, Accessed: January, 2017.
[18] B. Brown Microsoft Researcher: Why Micro Datacenters Really Matter To Mobile's Future, [online] Available: http://www.networkworld.com/article/2979570/cloud-computing/microsoft-researcher-why-micro-datacenters-really-matter-to-mobiles-future.html, Accessed: January, 2017.
[19] M. Satyanarayanan, P. Bahl, R. Cáceres, N. Davies, L. University, The case for VM-based cloudlets in mobile computing, IEEE Perv. Comput. 8 (4) (2009), doi:10.1109/MPRV.2009.82.
[20] Cloudlet-based Mobile Computing [online]. Available: http://elijah.cs.cmu.edu/, Accessed: January, 2017.
[21] V. Valancius, N. Laoutaris, L. Massoulié, P. Rodriguez, Greening the internet with nano data centers, in: Proceedings of the 2009 ACM Conference on Emerging Networking Experiments and Technology, CoNEXT, Rome, Italy, 2009, doi:10.1145/1658939.1658944.
[22] Content Delivery Networks Interconnection (CDNI), [online] Available: https://datatracker.ietf.org/wg/cdni/charter/, Accessed: January, 2017.
[23] Final Version of NIST Cloud Computing Definition Published, October 25, 2011. [online] Available: https://www.nist.gov/news-events/news/2011/10/final-version-nist-cloud-computing-definition-published, Accessed: February, 2017, 2011.
[24] P.G.V. Naranjo, M. Shojafar, L. Vaca-Cardenas, C. Canali, R. Lancellotti, E. Baccarelli, Big data over SmartGrid - a fog computing perspective, 24th International conference on software, telecommunications, and computer networks, 2016.

[25] S. Sarkar, S. Misra, Theoretical modelling of fog computing: a green computing paradigm to support IoT applications, IET Netw. 5 (2) (2016) 23–29.
[26] M. Aazam, E. Huh, Dynamic resource provisioning through fog micro datacenter, in: The 12th IEEE International Workshop on Managing Ubiquitous Communications and Services, 2015, pp. 105–110.
[27] M. Aazam, E. Huh, K. Hee, Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT, IEEE 29th International Conference on Advanced Information Networking and Applications, 2015.
[28] T.H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, Fog computing: focusing on mobile users at the edge, [online] arXiv: 1502.01815, 2015.
[29] I. Stojmenovic, S.W.X. Huang, H. Luan, An overview of Fog computing and its security issues, Concurr. Comput. Pract. Experience. (2015), doi:10.1002/cpe.3485.
[30] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: Proceedings of the 2015 Workshop on Mobile Big Data, Hangzhou, China, 2015, pp. 37–42.
[31] A. Ahmed, E. Ahmed, A survey on mobile edge computing, 10th IEEE International Conference on Intelligent Systems and Control, 2016, doi:10.1109/ISCO.2016.7727082.
[32] M. Aazam, E. Huh, M. St-Hilaire, C. Lung, I. Lambadaris, Cloud of things: integration of IoT with cloud computing, in: Robots and Sensor Clouds, in: Studies in Systems, Decision and Control, vol. 36, Springer, August 2015, pp. 77–94.
[33] A.V. Dastjerdi, R. Buyya, Fog computing: helping the internet of things realize its potential, IEEE Comput. Soc. 49 (08) (2016) 112–116.
[34] V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, R. Buyya, Fog Computing: principles, architectures, and applications, in: R. Buyya, A.V. Dastjerdi (Eds.), Internet of Things Principles and Paradigms, Elsevier, USA, 2016 ISBN: 978-0-12-805395-9, Chapter 4.
[35] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, D.S. Nikolopoulos, Challenges and opportunities in edge computing, in: IEEE International Conference on Smart Cloud (SmartCloud), 2016, pp. 18–20.
[36] 5 Ways Netflix is Changing Telecom Networks, [online] Available: https://www.linkedin.com/pulse/5-ways-netflix-changing-telecom-networks-avi-dorfman-1, Accessed: December, 2016.
[37] E. Nygren, R.K. Sitaraman, J. Sun, The Akamai network: a platform for high-performance internet applications, ACM SIGOPS Oper. Syst. 44 (3) (July 2010) 2–19.
[38] K. Ha, P. Pillai, G. Lewis, S. Simanta, S. Clinch, N. Davies, M. Satyanarayanan, The impact of mobile multimedia applications on data center consolidation, in: 2013 IEEE International Conference on Cloud Engineering (IC2E), March 2013, pp. 166–176.
[39] Green Clouds, [online] Available: http://www.greenclouds.in/resources/, Accessed: January, 2017.
[40] A. Giordano, G. Spezzano, Smart agents and fog computing for smart city applications, Smart Cities (2016) 137–146, doi:10.1007/978-3-319-39595-1_14.
[41] S. Agarwal, M. Philipose, P. Bahl, Vision: the case for cellular small cells for cloudlets, in: Proceedings of the fifth international workshop on Mobile cloud computing & services, 2014, pp. 1–5.
[42] T. Maqsood, O. Khalid, R. Irfan, S.A. Madani, S.U. Khan, Scalability issues in online social networks, ACM Comput. Surv. (2016).
[43] Foursquare by the Numbers [online]. Available: http://venturebeat.com/2015/08/18/foursquare-by-the-numbers-60m-registered-users-50m-maus-and-75m-tips-to-date/, Accessed: January, 2017.
[44] Foursquare Aims [online]. Available: https://techcrunch.com/2013/03/16/foursquare-aims-at-a-moving-target-as-it-tries-to-close-another-round-of-funding/, Accessed: January, 2017.
[45] Nike+ Sports Tracking [online]. Available: http://nikeplus.nike.com, Accessed: January, 2017.
[46] Runtastic Sports Tracking [online]. Available: https://www.runtastic.com/, Accessed: January, 2017.
[47] Runkeeper Sports Tracking [online]. Available: http://runkeeper.com, Accessed: January, 2017.
[48] Endomondo Sports Tracking [online]. Available: http://www.endomondo.com, Accessed: January, 2017.
[49] R. Cortesa, X. Bonnaireb, O. Marina, P. Sensa, Stream processing of healthcare sensor data: studying user traces to identify challenges from a big data perspective, in: The 4th International Workshop on Body Area Sensor Networks (BASNet-2015), Procedia Computer Science, 52, 2015, pp. 1004–1009.
[50] D. Costenaro, A. Duer, The megawatts behind your megabytes: going from data-center to desktop, ACEEE Summer Study on Energy Efficiency in Buildings, 2012.
[51] F. Jalali, Energy Consumption of Cloud Computing and Fog Computing Applications PhD Thesis, The University of Melbourne, 2015.
[52] Data Never Sleeps 2.0 [online]. Available: https://www.domo.com/learn/data-never-sleeps-2, Accessed: January, 2017.
[53] Cisco Data in Motion [online] Available: https://developer.cisco.com/site/data-in-motion/discover/overview/, Accessed: January, 2017.
[54] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, M. Satyanarayanan, Towards wearable cognitive assistance, in: Proceedings of the Twelfth International Conference on Mobile Systems, Applications and Services (MobiSys 2014), Bretton Woods, NH, June 2014.
[55] D. Ye, M. Wu, S. Tang, R. Yu, Scalable fog computing with service offloading in bus networks, in: 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), 2016, pp. 247–251.
[56] Cisco Unified Computing System [online] Available: http://www.cisco.com/c/en/us/products/servers-unified-computing/index.html, Accessed: January, 2017.
[57] VxBlock and Vblock Systems [online] Available: https://www.emc.com/en-us/converged-infrastructure/converged-systems.htm#collapse–, Accessed: January, 2017.
[58] Dell Active System Manager [online] Available: http://www.dell.com/learn/us/en/05/large-business/solution-converged-infrastructure-asim, Accessed: January, 2017.
[59] Schneider Electrics Microdata centers [online] Available: https://www.schneider-electric.com/b2b/en/solutions/system/s4/data-center-and-network-systems-micro-data-center, Accessed: January, 2017.
[60] DX Raser [online] Available: https://www.ellipticalmobilesolutions.com/raserdx.html, Accessed: January, 2017.
[61] MicroDC Solution [online], Available: http://e.huawei.com/en/solutions/industries/retail/shoppingmall/medium-sized-store, Accessed: January, 2017.
[62] Mobile Edge Computing [online] Available: http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing, Accessed: January, 2017.
[63] Mobile Edge Computing, Nokia [online], Available: https://networks.nokia.com/solutions/multi-access-edge-computing, Accessed: January, 2017.
[64] S. Agarwal, M. Philipose, P. Bahl, Vision: the case for cellular small cells for cloudlets, in: Proceedings of the International Workshop on Mobile Cloud Computing & Services, 2014, pp. 1–5.
[65] M. Aazam, E. Huh, Fog computing: the cloud-IoT/IoE middleware paradigm, IEEE Potentials (2016) 40–44.
[66] S. Sarkar, S. Chatterjee, and S. Misra, Assessment of the suitability of fog computing in the context of internet of things, *IEEE Trans. Cloud Comput.*, doi:10.1109/TCC.2015.2485206.
[67] Boeing 787s to Create Half a Terabyte of Data [online] Available: http://www.computerworlduk.com/data/boeing-787s-create-half-terabyte-of-data-per-flight-says-virgin-atlantic-3433595/, Accessed: January, 2017.
[68] Self-driving Cars Will Create 2 Petabytes of Data, [online] Available: https://datafloq.com/read/self-driving-cars-create-2-petabytes-data-annually/172, Accessed: January, 2017.
[69] The internet of things in smart commercial buildings 2016 to 2021, Market Prospects, Impacts and Opportunities [online] Available: http://www.memoori.com/portfolio/internet-things-smart-commercial-buildings-2016-2021/, Accessed: January, 2017.
[70] Gooee Enterprise Iot Lighting and Ecosystem [online] Available: https://gooee.com/, Accessed: January, 2017.
[71] PointGrab, [online] Available: http://www.pointgrab.com/about, Accessed: January, 2017.
[72] IoT Partnership for Smart Building Lighting [online] https://www.senseware.co/smart-building-solutions-integration-key/, Accessed: January, 2017.
[73] C. Boles, Intel IoT Solutions Transforming Smart Buildings from the Ground Up [online] Available: http://blogs.intel.com/iot/2016/06/07/intel-iot-solutions-transforming-smart-buildings-ground/, Accessed: January, 2017.
[74] Saving Energy with Intel and AVOB | Smart Building Management [online]. Available: http://www.intel.com/content/www/us/en/internet-of-things/solution-briefs/avob-smart-building-management-brief.html, Accessed: January, 2017.
[75] S.K. Datta, C. Bonnet, and J. Haerri, Fog computing architecture to enable consumer centric internet of things services, *EURECOM*, Biot, France.
[76] L.A. Tawalbeh, W. Bakheder, H. Song, A mobile cloud computing model using the cloudlet scheme for big data applications, in: Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on, IEEE, 2016, pp. 73–77.
[77] R. Brzoza-Woch, M. Konieczny, P. Nawrocki, T. Szydlo, K. Zielinski, Embedded systems in the application of fog computing – levee mon0069cFDtoring use case, in: 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES), 2016, pp. 1–6.
[78] White paper: "Cisco Visual Networking Index: Forecast and Methodology, 2015–2020," 2015, Document ID:1465272001663118.
[79] N. Chen, Y. Chen, Y. You, H. Ling, P. Liang, R. Zimmermann, Dynamic urban surveillance video stream processing using fog computing, in: 2016 IEEE Second International Conference on Multimedia Big Data, 2016, pp. 105–112.
[80] K. Bilal, A. Erbad, Impact of multiple video representations in live streaming: a cost, bandwidth, and QoE analysis, IEEE International Conference on Cloud Engineering (IC2E'17), April 2017.
[81] F. Chen, C. Zhang, F. Wang, J. Liu, X. Wang, Y. Liu, Cloud-assisted live streaming for crowdsourced multimedia content, in: IEEE Transactions on Multimedia, vol. 17, Sept. 2015, pp. 1471–1483.
[82] K. Pires, G. Simon, YouTube live and Twitch: a tour of user-generated live streaming systems, in: Proceedings of the 6th ACM Multimedia Systems Conference (MMSys '15), New York, 2015.
[83] Akamai, Akamai's State of the Internet Q1, Akamai's State of the Internet Q1, vol.7, 2014 report[online] Available: https://www.akamai.com/uk/en/multimedia/documents/state-of-the-internet/akamai-state-of-the-internet-report-q4-2014.pdf .
[84] Akamai, Akamai's State of the Internet Q1, Akamai's State of the Internet Q1, vol. 9, 2016 report[online] Available: https://www.akamai.com/es/es/multimedia/documents/state-of-the-internet/akamai-state-of-the-internet-report-q1-2016.pdf .

[85] Amazon CloudFront Pricing [online] Available: https://aws.amazon.com/cloudfront/pricing, Accessed: January, 2017.
[86] Twitch, The 2015 Retrospective [online] Available: https://www.twitch.tv/year/2015, Accessed: January, 2017.
[87] A. Hamza, M. Hefeeda, Adaptive streaming of interactive free viewpoint videos to heterogeneous clients, 7th International Conference on Multimedia Systems (MMSys '16), New York, 2016.
[88] S. Banerjee, D.O. Wu, Final Report from the NSF Workshop on Future Directions in Wireless Networking, National Science Foundation, Nov. 2013.
[89] M. Satyanarayanan, P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, W. Hu, B. Amos, Edge analytics in the internet of things, IEEE Perv. Comput. 14 (2) (April-June 2015).
[90] M. Mukerjee, D. Naylor, J. Jiang, S. Seshan, H. Zhang, Practical, real-time centralized control for CDN-based live video delivery, SIGCOMM Comput. Commun. Rev. 45 (4) (2015) 311–324.
[91] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, M. Satyanarayanan, Scalable crowd-sourcing of video from mobile devices, in: Proceedings of the Eleventh International Conference on Mobile Systems, Applications and Services (MobiSys 2013), Taipei, Taiwan, June 2013.
[92] M. Ramon, S. Caharel, B. Rossion, The speed of recognition of personally familiar faces, Perception 40 (4) (2011) 437–449.
[93] C. Wei, V.C. Leung, L. Hu, A cloudlet-assisted multiplayer cloud gaming system, Mob. Netw. Appl. 19 (2) (2014) 144–152.
[94] S.J.R. Méndez, Y. Wang, T.R. Mullen, T. Jung, Augmented brain computer interaction based on fog computing and linked data, 2014 International Conference on Intelligent Environments (IE), 2014.
[95] headit [online] Available: http://headit.ucsd.edu/, Accesses: January, 2017.
[96] PhysioNet [online] Available: https://www.physionet.org/, Accessed: January, 2017.
[97] BrainMap [online] Available: http://www.brainmap.org/, Accessed: January, 2017.
[98] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, W. Heinzelman, Cloud-vision: real-time face recognition using a mobile-cloudlet-cloud acceleration architecture, in: 2012 IEEE Symposium on Computers and Communications (ISCC), 2012, pp. 59–66.
[99] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, R.S. Tucker, Fog computing may help to save energy in cloud computing, IEEE J. Sel. Areas Commun. (2015), doi:10.1109/JSAC.2016.2545559.
[100] K. Gai, M. Qiu, H. Zhao, L. Tao, Z. Zong, Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing, J. Netw. Comput. Appl. 59 (2016) 46–54.
[101] X. Sun and N. Ansari, Green cloudlet network: a distributed green mobile cloud network, arXivpreprint arXiv: https://arxiv.org/abs/1605.07512 (2016).
[102] J. Li, J. Jin, D. Yuan, M. Palaniswami, K. Moessner, EHOPES: data-centered fog platform for smart living, in: 2015 International Telecommunication Networks and Applications Conference (ITNAC), 2015, pp. 308–313.
[103] M. Sneps-Sneppe and D. Namiot, On mobile cloud for smart city applications, arXiv preprint arXiv: https://arxiv.org/abs/1605.02886, 2016.
[104] T. Tarik, S. Dutta, A. Ksentini, M. Iqbal, H. Flinck, Mobile edge computing potential in making cities smarter, IEEE Commun. Mag. (2016).
[105] A. Ksentini, T. Taleb, F. Messaoudi, A LISP-based implementation of follow me cloud, IEEE Access 2 (Oct. 2014) 1340–1347.
[106] T. Taleb, A. Ksentini, An analytical model for Follow Me Cloud, in: Proc. IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, December 2013.
[107] M.U.S. Khan, O. Khalid, Y. Huang, F. Zhang, R. Ranjan, S.U. Khan, J. Cao, K. Li, B. Veeravalli, A. Zomaya, MacroServ: A route recommendation service for large-scale evacuations, IEEE Trans. Services Comput. 10 (4) (2017) 589–602.
[108] O. Khalid, M.U.S. Khan, Y. Huang, S.U. Khan, A.Y. Zomaya, EvacSys: a cloud-based service for emergency evacuation, IEEE Cloud Comput. 3 (1) (2016) 60–68.
[109] Towards Applications of Fog Computing for Designing Smart City Verticals [online] Available: http://www.telxperts.com/fog/towards-applications-fog-computing-designing-smart-city-verticals/, Accessed: January, 2017.
[110] Y. Yan, W. Su, A fog computing solution for advanced metering infrastructure, in: 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), 2016, pp. 1–4, doi:10.1109/TDC.2016.7519890.
[111] M. Saleem, H. Nazmudeen, A.T. Wan, S.M. Bukhari, Improved throughput for power line communication (PLC) for smart meters using fog computing based data aggregation approach, 2016 IEEE International Smart Cities Conference (ISC2), 2016, doi:10.1109/ISC2.2016.7580841.
[112] Y. Cao, S. Chen, P. Hou, D. Brown, FAST: a fog computing assisted distributed analytics system to monitor fall for stroke mitigation, 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), 2015, doi:10.1109/NAS.2015.7255196.
[113] X. Masip-Bruin, E. Marín-Tordera, A. Alonso, J. Garcia, Fog-to-cloud computing (F2C): the key technology enabler for dependable e-health services deployment, 2016, doi:10.1109/MedHocNet.2016.7528425.
[114] M. Quwaider, Yaser Jararweh, Cloudlet-based for big data collection in body area networks, in: IEEE 8th International Conference for Internet Technology and Secured Transactions (ICITST), 2013, 2013, pp. 137–141.
[115] A.E. Amraoui, K. Sethom, Cloudlet softwarization for pervasive healthcare, in: 2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, 2016, pp. 628–632.
[116] Q. Althebyan, Q. Yaseen, Y. Jararweh, M. Al-Ayyoub, Cloud support for large scale e-healthcare systems, Ann. Telecommun. (2016) 1–13.
[117] K. Intharawijitr, K. Iida, H. Koga, Analysis of fog model considering computing and communication latency in 5 g cellular networks, IEEE International Conference on Pervasive Computing and Communications Work in Progress, 2016.
[118] M. Peng, S. Yan, K. Zhang, C. Wang, Fog computing based radio access networks: issues and challenges, IEEE Netw. 30 (4) (2016) 46–53.
[119] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, A. Schneider, Enabling real-time context-aware collaboration through 5 g and mobile edge computing, in: IEEE 2015 12th International Conference on Information Technology-New Generations (ITNG), 2015, pp. 601–605.
[120] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, Z. Han, Fog computing in multi-tier data center networks: a hierarchical game approach, IEEE ICC 2016 SAC Cloud Communications and Networking, 2016.
[121] M. Aazam, E. Huh, Fog computing and smart gateway based communication for cloud of things, in: Proc. IEEE Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 2014, pp. 464–470.
[122] H. Yin, X. Zhang, H.H. Liu, Y. Luo, C. Tian, S. Zhao, F. Li, Edge provisioning with flexible server placement, IEEE Trans. Parallel Distrib. Syst. (2016), doi:10.1109/TPDS.2016.2604803.
[123] C. Do, N.H. Tran, C. Pham, A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing, in: International Conference on Information Networking (ICOIN), 2015, pp. 324–329.
[124] Accelerating Innovation and Collaboration for the Next Stage [online], Available: http://www.ntt.co.jp/news2013/1311ehzt/pdf/xgxf131108d_all.pdf, Accessed: January, 2017.
[125] J. Zhu, D.S. Chan, M.S. Prabhu, Improving web sites performance using edge servers in fog computing architecture, IEEE Seventh International Symposium on Service-Oriented System Engineering, 2013.
[126] D. Zeng, L. Gu, S. Guo, Z. Cheng, S. Yu, Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system, IEEE Trans. Comput. (2015), doi:10.1109/TC.2016.2536019.
[127] Z. Shelby, K. Hartke, C. Bormann, The Constrained Application Protocol (CoAP), Internet Engineering Task Force, Jun. 2014 RFC 7252 (Proposed Standard) [Online]. Available: http://www.ietf.org/rfc/rfc7252.txt .
[128] E.H. Butterfield, Fog Computing with Go: A Comparative Study, CMC Senior Theses. Paper 1348. http://scholarship.claremont.edu/cmc_theses/1348, 2016
[129] S. Cirani, G. Ferrari, N. Iotti, M. Picone, The IoT hub: a fog node for seamless management of heterogeneous connected smart objects, 2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking - Workshops (SECON Workshops), 2016, doi:10.1109/SECONW.2015.7328145.
[130] H. Gupta, A.V. Dastjerdi, S.K. Ghosh, R. Buyya, iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments, Cloud Computing and Distributed Systems Laboratory, Univ. of Melbourne, 2016 *technical report CLOUDS-TR-2016-2* http://cloudbus.org/tech_reports.html.
[131] U.U. Rahman, O. Hakeem, M. Raheem, K. Bilal, S.U. Khan, L.T. Yang, Nutshell: cloud simulation and current trends, IEEE International Conference on Smart City (SmartCity), December 2015.
[132] G. Orsini, D. Bade, W. Lamersdorf, Computing at the mobile edge: designing elastic android applications for computation offloading, 8th IFIP Wireless and Mobile Networking Conference (WMNC), 5–7 Oct., 2015.
[133] Cisco ParStream [online] Available: http://www.cisco.com/c/en/us/products/analytics-automation-software/parstream/index.html, Accessed: January, 2017.
[134] Vortex [online] Available: http://www.prismtech.com/vortex, Accessed: January, 2017.
[135] Cisco IOx [online] Available: http://www.cisco.com/c/en/us/products/cloud-systems-management/iox/index.html, Accessed: January, 2017.
[136] Amazon EMR Pricing, [online] Available: https://aws.amazon.com/emr/pricing/, Accessed: January, 2017.
[137] Mec - Applications at the Network Edge, [online] Available: http://networks.nokia.com/portfolio/solutions/mobile-edge-computing#tab-highlights, Accessed: January, 2017.
[138] U. Shaukat, E. Ahmed, Z. Anwar, F. Xia, Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges, J. Netw. Comput. Appl. 62 (2016) 18–40.
[139] I. Stojmenovic, S. Wen, X. Huang, H. Luan, An overview of Fog computing and its security issues, Concurr. Comput. (2015), doi:10.1002/cpe.3485View.
[140] R. Romana, J. Lopeza, M. Mambob, Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges, Future Generation Computer Systems (2016). http://dx.doi.org/10.1016/j.future.2016.11.009.
[141] LocalGrid Fog Computing Platform, [online] Available: http://www.localgridtech.com/, Accessed: January, 2017.
[142] Akamai, [online] Available: https://blogs.akamai.com/2016/07/portugal-france-sets-live-sports-streaming-record-on-akamai.html.
[143] Internet of Everything (IoE) [online] Available: https://newsroom.cisco.com/ioe, Accessed: February, 2017.
[144] Heterogeneous Network (Hetnet) [online] Available: https://www.ericsson.com/br/res/thecompany/docs/press/media_kits/hetnet_infographic_vertical_04.pdf, Accessed: February, 2017.
[145] M. Wang, P.P. Jayaraman, R. Ranjan, K. Mitra, M. Zhang, E. Li, S. Khan, M. Pathan, D. Georgeakopoulos, An overview of cloud based content delivery networks: research dimensions and state-of-the-art, in: Transactions on Large-Scale Data- and Knowledge-Centered Systems, in: Lecture Notes in Computer Science, vol. 9070, March 2015, pp. 131–158.

[146] W. Chu, L. Wang, H. Xie, Z. Zhang, Z. Jiang, Network delay guarantee for differentiated services in content-centric networking, Comput. Commun. 76 (2016) 54–66.
[147] R. Wang, X. Peng, J. Zhang, K.B. Letaief, Mobility-aware caching for content–centric wireless networks: modeling and methodology, IEEE Commun. Mag. (2016) 77–83.
[148] S.H. Ahmed, S.H. Bouk, D. Kim, Content-Centric Networks An Overview, Applications And Research Challenges, Content-Centric Networks An Overview, Applications And Research Challenges, 2016 ISBN: 978-981-10-0064-5.
[149] Twitch Tv [online] Available: https://www.twitch.tv/, Accessed: February, 2017.
[150] YouTube Live [online] Available: https://www.youtube.com/live, Accessed: February, 2017.
[151] Periscope [online] Available: https://www.periscope.tv/, Accessed: February, 2017.
[152] YouNow Broadcast Live [online] Available: https://www.younow.com/, Accessed: February, 2017.
[153] Livestream [online] Available: https://livestream.com/, Accessed: February, 2017.
[154] Amazon Web Services [online] Available: https://aws.amazon.com/, Accessed: February, 2017.
[155] C. Brennand, F. Cunha, G. Maia, FOX: a traffic management system of computer-based vehicles fog, 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, doi:10.1109/ISCC.2016.7543864.
[156] Demand For Mayweather-Mcgregor Fight Crashed Pay-Per-View Servers [online] https://www.engadget.com/2017/08/27/mayweather-mcgregor-fight-crashes-ppv-servers/, Accessed: September, 2017.
[157] L. Gu, D. Zeng, S. Guo, A. Barnawi, Y. Xiang, Cost efficient resource management in fog computing supported medical cyber-physical system sign in or purchase, IEEE Trans. Emerging Topics Comput. 5 (1) (2017) Jan.-March.
[158] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, Z. Han, A hierarchical game framework for resource management in fog computing, IEEE Commun. Mag. 55 (8) (2017) 52–57.
[159] Y. Sun, N. Zhang, A resource-sharing model based on a repeated game in fog computing, Saudi J. Biol. Sci. 24 (3) (March 2017) 687–694.
[160] SONM Aplha, https://sonm.io/, Accessed September 2017.
[161] K. Garg, J. Singh, A proposed technique for cloud computing security, in: Innovations in Computer Science and Engineering, June 2017, pp. 89–95.
[162] K. Lee, D. Kim, D. Ha, U. Rajput, H. Oh, On security and privacy issues of fog computing supported internet of things environment, 2015 6th International Conference on the Network of the Future (NOF), 30 Sept.-2 Oct, 2015.
[163] D. Ye, M. Wu, S. Tang, R. Yu, Scalable fog computing with service offloading in bus networks, 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud), 25-27 June, 2016.
[164] Y. Gao, W. Hu, K. Ha, B. Amos, et al., Are Cloudlets Necessary? School of Computer Science Carnegie Mellon University Pittsburgh, 2015.
[165] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5 G heterogeneous networks, IEEE Access. J. 4 (2016) 5896–5907.
[166] C. You, K. Huang, H. Chae, B.H. Kim, Energy-efficient resource allocation for mobile-edge computation offloading, IEEE Transa. Wirel. Commun. 16 (2017) 1397–1411.
[167] S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multi-cell mobile-edge computing, IEEE Trans. Signal Inf. Process. Netw. 1 (2015) 89–103.
[168] R. Deng, R. Lu, C. Lai, T.H. Luan, H. Liang, Optimal workload allocation in fog–cloud computing toward balanced delay and power consumption, IEEE Internet Things (2016) 1171–1181.
[169] K. Bilal, A. Erbad, Edge computing for interactive media and video streaming, in: IEEE Conference on Fog and Mobile Edge Computing (FMEC), 2017, pp. 68–73.
[170] G. Paschos, E. Bastug, I. Land, G. Caire, M. Debbah, Wireless caching: technical misconceptions and business barriers, IEEE Commun. Mag. 54 (8) (2016) 16–22.

Kashif Bilal is a postdoc researcher at Qatar University, Qatar. He received his PhD from North Dakota State University, USA in 2014. He is an Assistant Professor at COMSATS Institute of Information Technology, Pakistan. He was awarded NDSU CoE Researcher of the Year 2014 and COMSATS CS Researcher of the year 2016 awards. His research interests include cloud computing, edge technologies, data center networks, and multimedia in cloud.

Osman Khalid completed his Ph.D. in 2014 at the North Dakota State University, Fargo, USA. His area of research includes opportunistic networks, recommendation systems, and trust and reputation systems. He is an Assistant Professor at COMSATS Institute of Information Technology, Pakistan.

Aiman Erbad is an Assistant Professor and Director of Research Support at Qatar University. Dr. Erbad obtained a Ph.D. in Computer Science from the University of British Columbia, a Master of Computer Science in Embedded Systems and Robotics from the University of Essex and a Bachelor of Science in Computer Engineering from the University of Washington. His research interests span cloud computing, distributed systems and multimedia networking and systems.

Samee U. Khan is a Program Director at the National Science Foundation, where he is responsible for the Smart & Autonomous Systems program, Critical Resilient Interdependent Infrastructure Systems and Processes program, and Computer Systems Research cluster. He also is a faculty at the North Dakota State University. He is an ACM Distinguished Speaker, an IEEE Distinguished Lecturer, a Fellow of the IET, and a Fellow of the BCS. Samee's research interests include optimization, robustness, and security of computer systems. He is an associate editor of the IEEE Access, IEEE Communications Surveys and Tutorials, IET Wireless Sensor Systems, Scalable Computing and Communications, IET Cyber-Physical Systems, and IEEE IT Pro.