Edge and Fog Computing for IoT: A Survey on Current Research Activities & Future Directions

Mohammed Laroui[a,b,c], Boubakr Nour[d], Hassine Moungla[a,c], Moussa A. Cherif[b], Hossam Afifi[c], Mohsen Guizani[e]

[a] *Université de Paris, Paris, France*
[b] *Djillali Liabes University, Sidi Bel Abbes, Algeria*
[c] *Telecom SudParis, Institut Polytechnique de Paris, Palaiseau, France*
[d] *Beijing Institute of Technology, Beijing, China*
[e] *Qatar University, Doha, Qatar*

Abstract

The Internet of Things (IoT) allows communication between devices, things, and any digital assets that send and receive data over a network without requiring interaction with a human. The main characteristic of IoT is the enormous quantity of data created by end-user's devices that needs to be processed in a short time in the cloud. The current cloud-computing concept is not efficient to analyze very large data in a very short time and satisfy the users' requirements. Analyzing the enormous quantity of data by the cloud will take a lot of time, which affects the quality of service (QoS) and negatively influences the IoT applications and the overall network performance. To overcome such challenges, a new architecture called edge computing - that allows to decentralize the process of data from the cloud to the network edge has been proposed to solve the problems occurred by using the cloud computing approach. Furthermore, edge computing supports IoT applications that require a short response time and consequently enhances the consumption of energy, resource utilization, etc. Motivated by the extensive research efforts in the edge computing and IoT applications, in this paper, we present a comprehensive review of edge and fog computing research in the IoT. We investigate the role of cloud, fog, and edge computing in the IoT environment. Subsequently, we cover in detail, different IoT use cases with edge and fog computing, the task scheduling in edge computing, the merger of software-defined networks (SDN) and network function virtualization (NFV) with edge computing, security and privacy efforts, and the Blockchain in edge computing. Furthermore, we present the existing simulation tools. Finally, we also identify open research challenges and highlight future research directions.

Keywords: Internet of Things (IoT), Edge Computing, Cloud Computing

1. Introduction

The successive emergence of new generations of networks has led to serious challenges in terms of providing the requirements of various new applications, most of which require advanced infrastructure in order to provide the necessary resources to ensure high quality of the service provided. For this reason, the short response time in addition to analyzing the various data in a short time is considered one of the most important problems that need care and scrutiny in order to ensure it at the highest level of continuity in providing the service.

The term Internet of things was envisaged and mentioned in the late 1990s by professors at MIT who defined the future world of communication. The main objective behind the IoT is to analyze and process data emanating from massive devices, in the cloud. This process, also called cloud computing [1] where the huge in numbers but rather small amounts of individual data gathered by connected things, can be processed in the cloud. Big data architectures such as Hadoop and Spark were initially employed but because of the dramatic increase of connected things and the evolution of the IoT, the cloud computing quickly showed its limits. It is not efficient to simultaneously support this large number of devices, especially that most applications in the IoT category, are time sensitive. Cloud was down-scaled to edge computing [2]. It allows data analysis process at the network edge and supports the requirements of the future generations of networks.

1.1. Related Surveys

Different surveys have been proposed for each of cloud, edge, and fog computing based IoT. In the next step, we present a review of the related survey papers along with their contributions and limitations, Table 1 provides a summary of these subsections as well as a comparison with our surveys.

Yi *et al.* [3] surveyed fog computing applications and use cases such as augmented reality (AR), content delivery and caching, mobile big data analytics, etc. The authors also discussed different issues related to fog computing with SDN/NFV, QoS metrics, computation offloading, etc. However, the study provided is very short and limited. Mouradian *et al.* [6] provided a review of different evaluation criteria in fog

Email addresses: mohammed.laroui@parisdescartes.fr (Mohammed Laroui), n.boubakr@bit.edu.cn (Boubakr Nour), hassine.moungla@parisdescartes.fr (Hassine Moungla), moussa.alicherif@univ-sba.dz (Moussa A. Cherif), hossam.afifi@telecom-sudparis.eu (Hossam Afifi), mguizani@ieee.org (Mohsen Guizani)

Table 1: A Summary of Existing Related Survey Papers

Ref.	Topics Covered	Limitations	Year
Yi *et al.* [3]	• Overview of fog computing. • Fog computing use cases & applications. • QoS issues. • Security & privacy issues.	• Very short & limited study. • Learned lessons.	2015
Chiang *et al.* [4]	• Networking context of IoT based on edge computing. • Fog computing architecture benefits. • Fog computing use cases & applications.	• Integration of edge computing With other architectures (SDN/NFV, etc.).	2016
Shi *et al.* [5]	• A vision on edge computing. • Edge computing benefits. • Edge computing case studies. • Challenges & Opportunities.	• Edge computing platform & architecture. • Integration of edge computing With other architectures (SDN/NFV, etc.). • Recent research reviews. • Learned lessons.	2016
Mouradian *et al.* [6]	• Fog system research (architecture & algorithms). • Fog computing use cases and evaluation criteria. • Challenges & Research directions. • Learned lessons & Prospects.	• Security & Privacy.	2017
Hu *et al.* [7]	• Fog computing architecture & applications. • Challenges and open issues.	• Simulators for edge computing environment.	2017
Ni *et al.* [8]	• Overview of fog computing (evolution from cloud to fog, fog computing architecture,..). • IoT applications based on fog Computing. • Fog computing security. • Issues, challenges & future research directions.	• Fog computing protocols & services.	2017
Baktir *et al.* [9]	• SDN-Edge computing corporation. • Edge computing uses cases. • Future directions & research areas.	• Platforms & Simulators for SDN-Edge computing environment.	2017
Mukherjee *et al.* [10]	• Fog computing-based architectures. • QoS model in fog computing. • Resource management & Service allocation issues in fog computing. • Fog computing applications. • Open research challenges & Future directions.	• Security & Privacy review in fog computing.	2018
Yu *et al.* [11]	• Review of IoT & edge computing. • IoT-Edge computing integration. • Benefits of edge computing Based-IoT. • IoT-Edge computing challenges.	• Simulators for edge computing environment.	2018
Mahmud *et al.* [12]	• Taxonomy of fog computing. • Fog nodes configuration. • Challenges in fog computing.	• Simulators for edge computing environment.	2018
Omoniwa *et al.* [13]	• Fog/Edge computing-based IoT (FECIoT) architecture. • FECIoT protocols, services & applications. • Security & Privacy in FECIoT. • FECIoT simulation tools. • Open research issues.	• Integration of edge computing with other architectures (SDN/NFV, etc.). • Review of task scheduling in edge computing. • Review of vehicular edge computing (VEC).	2018
khan *et al. [14]*	• Cloud & Edge computing systems. • Cloud & Edge computing applications. • Mobile Edge computing. • Open challenges.	• Integration of edge computing with other architectures (SDN/NFV, etc). • Simulators for edge computing environment. • Review of Vehicular edge computing (VEC).	2019
Our Survey	• IoT overview. • Edge computing architecture. • Edge computing use cases & applications. • Review of task scheduling in edge computing. • Integration of edge computing avec SDN/NFV. • Review of security & privacy efforts. • Review of Blockchain in edge computing. • Future research directions.	-	2020

computing including network architectures and algorithms. The authors described both fog and content delivery networks [15], and a fog system for detecting and fighting fires [16]. Although the authors provided some key guidelines, the paper is missing different important aspects including security and privacy. Similarly, Hu *et al.* [7] presented the fog computing concept from different perspectives such as real-time interactions, low latency applications, mobility support, geographical distribution, etc. The authors highlighted the differences between cloud and fog computing, and the key technologies to enable fog computing based networks. However, the simulation tools and edge computing have not been covered. Mukherjee *et al.* [10] summarized the fog computing architectures based on different technologies and systems. The authors also presented different open research challenges and future directions have been discussed. However, security and privacy efforts have not been reviewed. While Ni *et al.* [8] reviewed the security and privacy issues in fog computing in general and IoT applications in particular. Although, the authors presented some guidelines and research directions, focusing on one aspect is a major point of this work. Mahmud *et al.* [12] focused on fog computing, the key components, and different challenges. The authors provided a comprehensive taxonomy on fog computing but, the simulation tools were not covered. Baktir *et al.* [9] reviewed the edge computing concept and discussed the SDN-Edge computing cooperation paradigm. In addition, different scenarios and use cases have been elaborated, as well as describing the SDN capabilities on top of edge computing. However, it focuses only on one technology (i.e. SDN).

Focusing on IoT networks and applications, Chiang *et al.* [4] focused on IoT applications and its challenges in the fog computing environment. The authors described how fog computing may help to overcome IoT issues in an effective way, such as latency, security, etc. However, the survey paper did not cover recent technologies such as VANET, SDN, NFV, etc. Also, other fog computing mechanism are not covered. Similarly, Shi *et al.* [5] targeted the edge computing, and discussed its need in IoT networks. The authors also provided different use-case scenarios such as video analytics, cloud offloading, smart homes, smart cities, and collaborative edge. At the edge, a summary is provided with some challenges in programmability of edge computing, integration with Named Data Networking [17–20] and MobilityFirst [21], data abstraction, etc. However, this survey did not cover different platforms and network architectures for fog computing. Yu *et al.* [11] targeted edge computing for IoT applications, by providing the advantages of such a merger as well as the challenges. However, the work is missing the platforms, systems, and simulation tools used. Omoniwa *et al.* [13] reviewed edge/fog computing-based IoT applications. The authors provided a wide review on different architectures, protocols, and technologies. However, they did not cover the integration of edge computing with other recent architectures.

Finally, Khan *et al.* [14] presented a comprehensive survey on edge computing, they reviewed the cloud, edge and mobile edge computing systems, besides they detailed the different applications. However, the authors did not cover the integration of edge computing with other recent architectures.

1.2. Our Contributions

In contrast with the aforementioned related surveys, our work focuses on edge/fog computing[1] for the IoT.

In this regard, the major contributions are summarized as follows:

1. We comprehensively review edge computing technology in the environment of the IoT.

2. We describe in detail, the IoT technology, the edge computing technology, and its benefits compared to cloud computing. Moreover, the applications of edge computing based IoT, as well as a detailed description of the most used simulation platforms for the edge computing environment.

3. We present an in-depth overview of the issues raised in the Edge-IoT environment such as *Task Scheduling, SDN/NFV, Security & Privacy, and the Blockchain.*

4. We describe the challenges faced the edge computing on top of the IoT applications such as *scalability, high mobility support, energy management, machine learning & deep learning , and of course security & privacy.*

1.3. Organization

The remainder of the paper is organized as follows. Section 2 presents an overview of the IoT. In section 3, we show a vision on cloud computing. In section 4 the detail of edge computing is presented. In section 5, we present edge computing use cases and applications. Section 6 provides a research review of task scheduling in edge computing. Beside, the SDN/NFV based edge computing is presented in Section 7. Moreover, in Section 8 the security & privacy efforts in edge computing are presented. While the Blockchain findings in edge computing environment are presented in Section 9. Next, the most used simulators are reviewed in Section 10. Section 11 highlights the future research guidelines & directions. Finally, we conclude the paper in section 12.

Figure 2 shows the reading plan of our survey.

2. Internet of Things (IoT)

The evolution of mobiles devices, embedded systems, and vehicles helped to create a smart world of connected devices that may sense, collect data, collaborate, and take decisions without interaction with humans [22]. This smart ecosystem is called the Internet of things [23].

[1] Without loss of generality, we use the terms 'Edge Computing', and 'Fog Computing' interchangeably in this paper.

2.1. *Definition*

It is a new technology envisioned as a network of devices and machines that communicate with each other and the Internet. The IoT is known as one of the important enablers for future technologies. It also has a great interest from companies. In a broader sense, IoT aims to create systems based on the interconnection of smart objects. These objects exchange information among themselves using different protocols, such as Wi-Fi, Bluetooth, ZigBee, etc. The main characteristic of the IoT is the integration of different technologies of communications (e.g. wired & wireless sensors, actuator networks, tracking and identification networks, etc.) to improve the cooperation and interaction between various technologies. Figure 1 shows the application domains of the IoT.

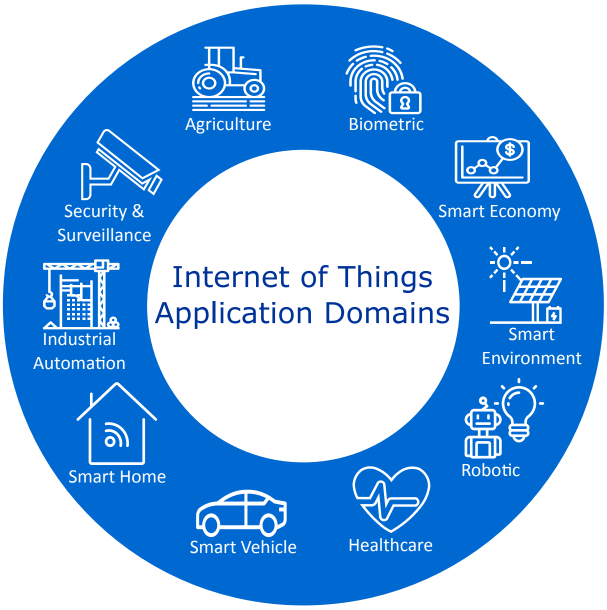

Figure 1: IoT Application domains [24]

2.2. *Fundamental Technologies*

The required components for the IoT consist of three types [24]: (a) actuator: hardware equipped with sensors, (b) middleware of storage and data analytics, and (c) interpretation and visualization tools. The following technologies make up the components defined above:

• *RFID & NFC*: RFID (Radio Frequency Identification) technology [25] is a major innovation in the communication paradigm that enables the design and development of microchips. It allows an automatic identification of anything attached with an electronic barcode. RFID devices, generally called RFID tags, is a microchip for wireless data transmission. the RFID tags send data over the air, the signal is recovered by an RFID reader which allows the identification of objects corresponding to the received information (barcode). For the IoT, the RFID is one of the devices most used for the building of applications, such as controlling privacy [26], smart healthcare and social applications. NFC (Near Field Communication) [27], is a half-duplex protocol for wireless short-range communication which facilitates the mobile phone usage of people, it offers various services of loyalty applications such as access keys for houses and offices. In addition, it allows smart-phones to be used to lock/unlock the house doors, and car, exchange business cards, pay for public transportation, newspaper, and much more.

• *Wireless Sensor Networks (WSN)* [28]: is a large number of smart sensors that aim to collect data (e.g., movement, temperature, etc.), process, analyze and transfer information [24]. A Wireless Sensor Network is mainly composed of the following components: (a) the capture unit (sensor) which is responsible for collecting data as signals, and transforming these signals into digital information understandable by the processing unit, (b) the processing unit which is responsible for analyzing the data captured, (c) the transmission unit that performs all transmission and reception of the data, and (d) the energy control unit which is an essential part of the system, it must distribute the energy available to the other modules in an optimal manner.

Moreover, several types of applications can be in operation using WSN including environmental monitoring (e.g., pollution, landslides, forest fires, etc.), commercial (e.g., smart light control, robotic), military, and medical applications, etc [29].

• *Data Storage and Analytics*: in IoT, an enormous quantity of data is generated and exchanged which requires a very large storage size. For this reason, the data storage is an important issue in the IoT. Different solutions have been proposed for the analysis and storage of data to provide efficient communication, e.g., smart cities, smart and connected communities, and smarter healthcare. In 2012, the storage of data in the cloud became more popular [24], while nowadays, cloud-based storage and analytics are mostly used and preferred which can accelerate data processing and provide a reliable data exchange.

2.3. *Architectures*

The IoT architecture is still under construction and there is no standardized architecture yet proposed [30]. Different IoT architectures have been proposed that can support specific or generic use cases. A basic/generic IoT architecture consists of three layers (a) perception layer that includes sensors, cameras, RFIDs, etc., (b) a network layer that is responsible for the transmission of the collected/generated data from the previous layer, and (c) the application layer that represents the user application. An-other IoT architecture called cloud computing [1] has been designed with 2-tiers perspectives. The IoT devices may connect directly to the cloud for data processing. This architecture has some problems in the IoT, for example, it does not support delay-based applications that require a short response time. To overcome such issues, a new architecture based on edge computing [2] which is a 3-tiers architecture that processes the data on the network edge which provide an efficient services close to end-user devices.

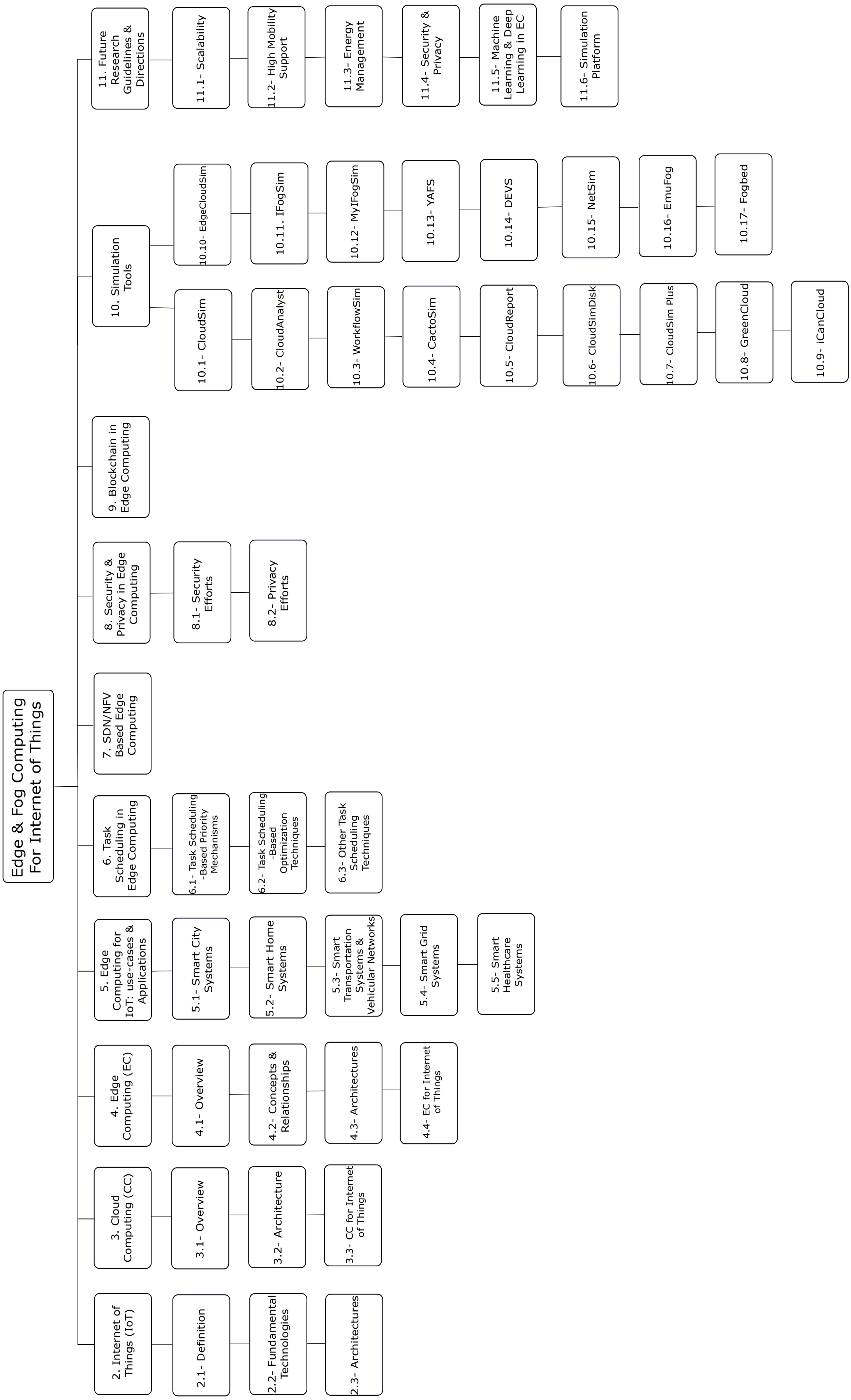

Figure 2: Survey reading plan.

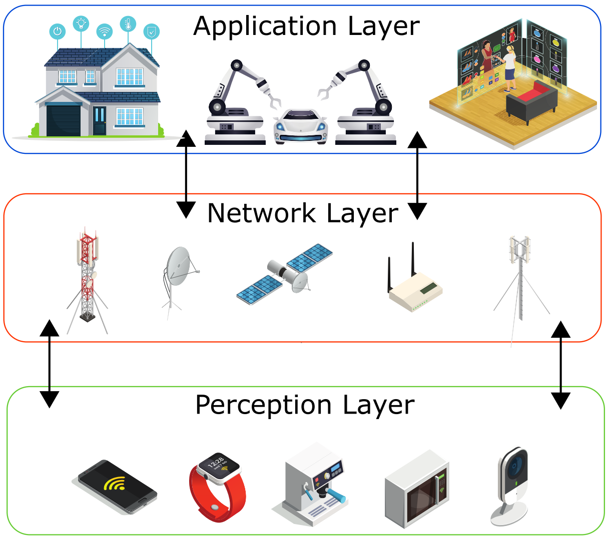

Figure 3: Three-layer IoT Architecture.

3. Cloud Computing (CC)

The rapid increase in connected/smart devices in the world and the change in users' and applications' requirements with a large quantity of data processing led to a set of novel technologies to allow fast data processing and reliable services. Cloud computing is one of these technologies.

3.1. Overview

The use of smartphones and computers has increased exponentially. This growth has heightened the need for efficient architectures that have naturally emerged to support the increase of connected devices, as well as the data generated and processed. Cloud computing [1] is an innovation paradigm, it seeks to provide various services to end-users in the cloud. Different types of cloud can be deployed including private, public, hybrid, and community. The public cloud [31] provides services to a large number of users on the Internet. The private cloud [32] offers specific services to private organizations. The community cloud [33] aims to provide services to a group of organizations. Finally, the hybrid cloud [34] is good for organizations to balance between cost and issues of control.

3.2. Architecture

An architecture must define what kind of service to offer by the cloud system. Three main categories of cloud services (as shown in figure 4) can be distinguished:

1. *Software as a service (SaaS):* [35] is an application programming interface (API) or web service(s) that encourages developers to create their applications in the cloud. SaaS provides software solutions that can be accessed via the Internet without the need to install any application on a user's local computer. 2. *Platform as a service (PaaS):* [36] is a way to rent operating systems, network capacity, storage and hardware over the Internet. This model allows users to run their applications in virtualized servers in the cloud without the need to extend their

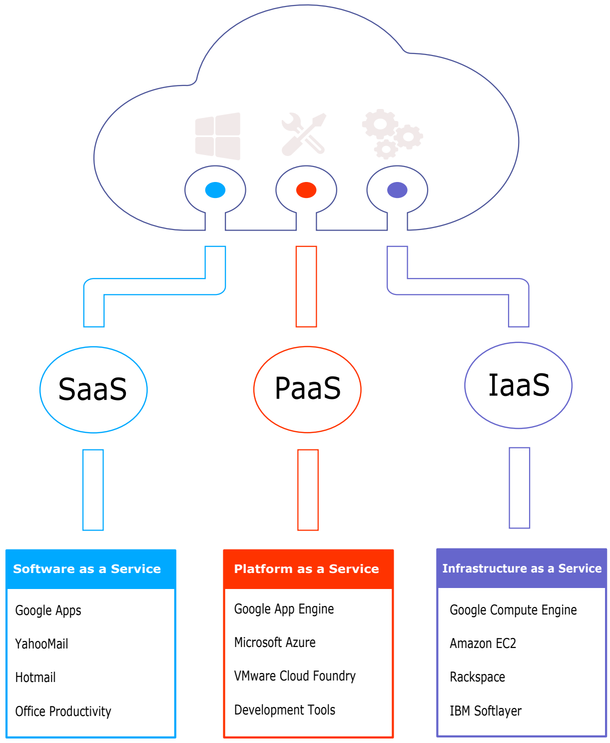

Figure 4: Cloud Computing Services

local resources. PaaS generally provides services for designing and deploying applications. 3. *Infrastructure as a Service (IaaS):* [37] allows users to manage and control software and hardware resources at the cloud. The use of IaaS has different benefits where users can access applications and platform from anywhere, using any-device, and from any-network, provide virtual infrastructure, as well as provide services of load balancing and a large capacity for computing.

3.3. CC for Internet of Things

The cloud computing and the IoT are two main technologies that contribute to our daily life by providing various services to the IoT users. Therefore, we can merge them into one global technology, namely CloudIoT paradigm [38]. Botta *et al.* [39] tried to integrate cloud and the IoT, by allowing the cloud to provide the required services to IoT devices such as storage and computation. Similarly, Neagu *et al.* [40] proposed a health monitoring service called the HM oriented Sensing Service scenario (HM-SS) that provides large medical facilities based on the cloud-IoT architecture. Ismail *et al.* [41] studied the efficiency of a virtual machine placement and task scheduling algorithm in the Cloud-IoT architecture in terms of energy consumption by data centers. Almolhis *et al.* [42] presented a review of the security issues in the CloudIoT system. In the *CloudIoT* paradigm, the end users connect directly to the cloud using the Internet and start exchanging data over the network which results a massive quantity of data in a short time, a generic cloud computing architecture for IoT is illustrated in figure 5. However, the centralized architecture of the cloud is not efficient to process the massive quantity of data generated

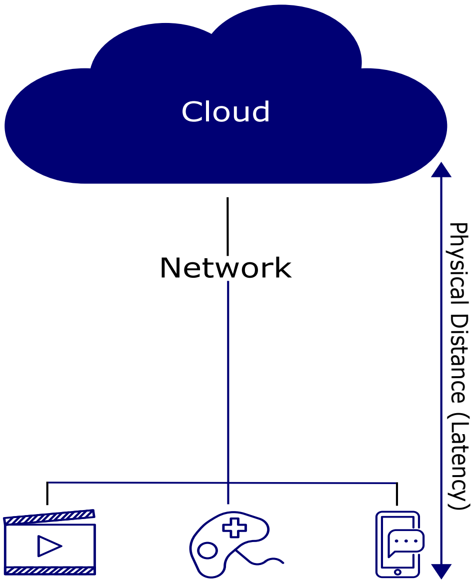

Figure 5: Generic Cloud Computing Architecture.

by the IoT devices that require a short response time. To overcome such an issue, an alternative paradigm, namely edge computing, has been used which processes the data in connected devices or the local gateways.

4. Edge Computing (EC)

The IoT will potentially connect billions or trillions of devices by the year 2025 [43]. All these devices generate a massive quantity of data, that will reach 500 zettabytes [5]. Furthermore, the new generation of applications such as video streaming, online gaming,..etc, requires a short response time [5, 44], besides, energy consumption is an important issue in wireless communication because of the limited resources of IoT devices that can not provide computation and processing locally. Thus, the centralized architecture does not fulfill IoT application requirements by providing storage, computation, and networking resources in data-centers that are owned by companies, such as Google, Microsoft, and Amazon. In addition, the data generated and processed needs to be sent to the cloud for processing which takes a large response time and may effect both end-user QoS and experience.

4.1. Overview

The edge computing aims to be the future IoT solution that solves different issues including time-constrained and computation-based applications. The benefits of processing data at the network edge are: reduce the networking load and communication latency, break the monopoly of big inventors [2], give the small and medium inventors every chance to help nurture future innovations, reduce the energy consumption of the mobile nodes, and eliminating the congestion within the core network, as well as provide more reliability, security, and the privacy protection.

4.2. Concepts & Relationships

To cope with the confusion over the difference between edge computing and fog computing, we start this discussion by describing both edge and fog computing from different perspectives.

Some researchers define edge and fog computing as the same concept with differences only in their names [45], while others differentiate them as two different concepts. Goscinski *et al.* [46] claims that edge computing focuses on data processing at the edge, while fog computing is located out between the cloud and the edge cloud, and thereby include the edge. On the other hand, Chiang *et al.* [47] said that the fog computing is an end-to-end architecture that distributes the control, storage, computing and networking function closer to end-users along the cloud-to-things, while the edge refers to the edge network, with equipment such as base stations, home gateways and edge routers. Furthermore, Pan *et al.* [48] said that the fog computing is a background of the IoT, it extends the cloud computing and different services to the devices such as switches, routers, multiplexers, etc. While the edge computing pushes data, services and applications from the core to the network edge, based on the core-edge topology [5, 49]. The video analytics, smart city, smart home and cloud offloading are application examples of edge computing.

In a nutshell, both edge and fog computing have the same research topics and both of them aim at the decentralization of data processing from the cloud to the network edge.

4.3. Architectures

The fog/edge layer generally is located between the cloud and the end users (as shown in figure 6), and includes the following main components [45, 50, 51]: (a) *Authentication and Authorization*: identify the access control rules and policies, (b) *Offloading Management*: defines types of information in the offloading process, the partition for offloading and the manner to design an optimal offloading scheme, (c) *Location Services*: that tends to learn the mobility model by mapping the network with the physical locations, (d) *System Monitor*: to provide different information such as usage, workload and energy to the other components, (e) *Resource Management*: which is responsible for resource discovery and allocation, dynamic joining and leaving of the fog node and maintaining and provisioning of resources, and (f) *VM Scheduling*: that aims to provide an optimal strategy for virtual machine scheduling.

4.4. EC for Internet of Things

The edge provides networking services, storage and computation to end users in IoT [52]. For example, a CCTV network doesn't need to send all data to the cloud, the movement detection algorithms or facial recognition running on the fog/edge layer, hence saving storage space and bandwidth. So, the fog/edge will be the best option of applications that require a temporary storage. In addition the fog/edge can play its role in the future, where the IoT and wireless sensors networks presents with the integration of heterogeneous protocols and devices for enhancing services [53].

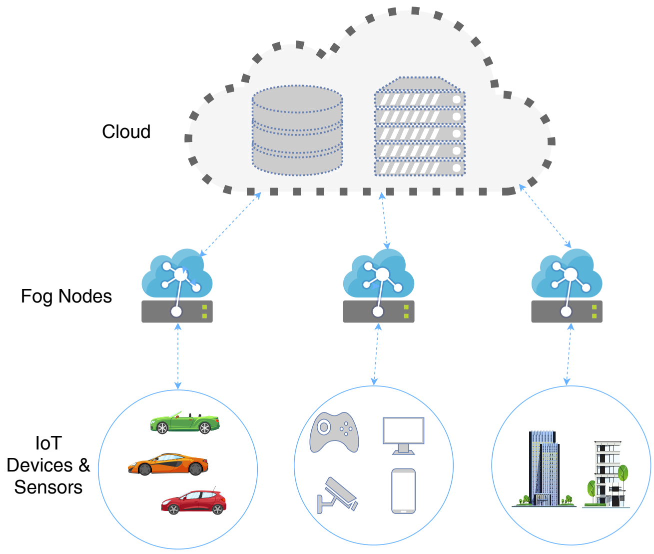

Figure 6: Generic Edge Computing Architecture

The edge computing is an appropriate paradigm for the current Internet in general and the IoT in particular in which this distributed architecture may provide a high-level communication and efficiency in the edge network rather than at the cloud-level. It supports a wide range of today's IoT applications, especially those that require a short response time.

table 2 outlines the main characteristics of cloud compared with fog/edge computing.

5. Edge Computing for IoT: Use Cases & Applications

The edge computing provides an efficient platform for smart IoT applications, such as smart homes, smart vehicles, smart grid, smart cities, smart healthcare, etc. This section surveys the existing efforts on IoT application based edge computing, while table 3 outlines the main features for such proposed applications.

5.1. Smart City Systems

A smart city [77] is an urban area where numerous sectors pull together to achieve results efficiently through the real-time analysis of information collected from multiple sources. Smart cities aim to decrease the energy waste and traffic congestion which may directly enhance the quality of life. Different technologies are used in smart cities which will be a massive economic market worth 1.565 trillion dollars by 2020, increasing to 3.3 trillion dollars by 2025 [55]. The deployment of smart cities still faces different challenges and issues such as big data analysis, large-scale sensing networks due to thousands of connected sensors and actuators in a large geographic area, machine-to-machine (M2M) and cooperative communication (CC) in wireless networks that aim to improve the efficiency in M2M network resources, and road traffic networks consist of studding and integrating data learning and processing algorithms in the intelligent traffic system [78].

Smart cities require an efficient and intelligent data analysis for data monitoring in order to achieve a fast and automated decision without the need for human interaction [79]. Hereby, guarantee a reliability of components and safety of people. Moreover, the smartness requires an efficient platform which contains many advanced algorithms, such as supervised and non-supervised machine learning algorithms [80], density distribution modeling [81] and sequential data learning [82].

The edge computing is an ideal platform for smart cities [83], due to its design, the data process will happen at the network edge instead of the cloud [5] and enhance the network delay and user experience. This is an essential element to build reliable smart cities that are characterized by the following:

• *Huge Quantity of Data*: by 2019, 1 million people for each city will produce more than 180 Petabyte (PB) data per day [84]. The data generated varies from transport, utility, traffic lights, etc. Handling all the data in a centralized cloud is unrealistic as the processing of the data will take a long time and too much resources. For this, the data processing in edge of the network will resolve the problems of centralized cloud by providing an efficient services for data processing which is required by smart cities applications.

• *Low Latency*: Most smart cities applications require a low latency such as public safety and health emergencies [85]. The edge computing is a promising paradigm, since it could organize the network structure and decrease the data transmission time. Diagnosis and decision could be made at the network edge which would decrease the latency compared to the centralized cloud paradigm that has a long response time [5].

• *Location Awareness*: some smart cities applications, such as utility management and transportation, are geographic-based in nature. The edge computing supports the location awareness by collecting the data based on geographic sites without the need to send data to the cloud. This prevents the data access delay, and increases the QoS, as well as speeding up and providing a transparency of the information transferred [5].

5.2. Smart Home Systems

The smart home [86] aims to allow intelligent control of different smart devices that are connected inside homes, such as TV, air-conditioner, cooker, fridge, etc. The intelligent services [87] can be classified into three types: (a) *Home Automation Services*: such as controlling air cleaners, air conditioners and curtain movements, (b) *Home Security Services*: such as preventing gas explosions and detection of potential crimes and lastly, (c) *Home Management Services*: such as intelligent control of smart devices like TV and cooker.

The connected smart devices generate a large quantity of data for smart control and decision-making inside homes. The analysis and the processing of this data require an enormous quantity of resources and storage which need a scalable architecture to guarantee quality and continuity of service without any degradation. The edge computing offers a highly distributed architecture for building smart homes in an efficient way [53] by processing the data at the network edge which provides: low latency, efficient data processing and less energy consumption.

Table 2: Characteristics of Fog/Edge compared to Cloud

Characteristics	Cloud	Edge/Fog
Location	Centralized.	Distributed in different geographical locations.
Capacity	Very large size data centers.	A lot of small size fog nodes that form a large system.
Energy consumption	High.	Low.
Latency	High, because of the large distance between the end users and the cloud.	Low, because of the short distance between the edge and the end users.
Proximity of resources and services	Far from end users, in the data centers.	Close to end users, in the edge of network.
Applications	Supports applications that doesn't require short delay. Mainstream cloud applications.	Support most types of applications. VR. Smart homes. Smart vehicles. Smart cities.
Service Cost	High, due to the monopoly of data centers by big companies.	Lower cost, due to the processing of data at the network edge.

The edge computing-based smart home aims to enhance the future IoT applications, especially those that require a short response time, such as intelligent control of smart devices and the surveillance inside the home.

5.3. Smart Transportation Systems & Vehicular Networks

A smart vehicle environment [88, 89] consists of connected cars, it can be classified in the following communication models: vehicle-to-vehicle (V2V) [90], vehicle-to-infrastructure (V2I) [91], and vehicle-to-grid (V2G) [92]. The edge computing offers an efficient distributed architecture for smart vehicles. It can support high mobility of vehicles and interactions among them [93]. Different applications in smart vehicles and transportation systems may benefit from edge computing, including:

• *Safety Applications*: this kind of applications can be used to adjust hazards on the road [94], sending notifications and warnings about crashes, curve speeds, traffic violations, precrash sensing, etc. They could include sensing of approaching emergency vehicles.

• *Convenience Applications*: include personal routing and advice of congestion situations [95], as well as, in some incidents such as network breakdown and power failure. In some scenarios, the connected vehicle can play an important role in the monitoring of road and weather by sharing information from vehicle sensors, thus providing SOS and emergency calls.

• *Smart Traffic Lights*: smart lights synchronize the connected vehicles by sending warning signals using 4G or WiFi to share traffic data which can help vehicles in different situations [96] including problems with traffic lights [97], and warnings of approaching pedestrian crossings [98].

• *Smart Parking Systems*: Most big cities faces traffic congestion. As a consequence, finding an empty space in a parking lot is difficult and expensive. For this it is important to solve the congestion problem by building smart parking connected to smart vehicles to automatically control the access to available spaces in parking lots. The parking system informs the smart vehicles if there are available spaces in the parking lot using notification messages through a wireless network or publish-subscribe communication model. This allows people to gain time and effortlessly find a place to stop their vehicles.

• *Commercial Applications* : represents paid services of diagnostics for vehicle problems and location-based services, such as entertainment and advertisements, updates of social networks, etc [99]. These applications are provided by private companies to generally offer special applications or sensors installed on-board the vehicles which can directly connect with the edge server of the associated company.

The edge computing is an efficient platform that provides various benefits and responses to smart vehicles and intelligent transportation requirements. The edge servers are localized nearby to smart vehicles, which allows rapid and efficient services to enhance the quality of life and experience.

5.4. Smart Grid Systems

A smart grid system [100] consists of smart meters which exist in different locations to measure the real time status for the distribution of electricity. This information is analyzed by a centralized server called SCADA [101], that by its return sends commands about emergency demands, responds to any change request to stabilize the system and protects the power grid.

The edge computing may offer great services to smart grid [64, 102]. With the edge computing paradigm, the SCADA can be equipped with a decentralized model which can improve the network cost, security and scalability. In addition, integrating power generators (e.g., solar panels, wind farms, etc.) with the main power grid that offers a centralized control of the power network. With the edge computing, the smart grid system will turn into a hierarchical architecture (multi-tier architecture) with interactions among the SCADA [52]. In such a system, the edge layer is in charge of the micro-grid and exchanges information with the neighboring edge and the higher tiers. The final coverage is managed by SCADA which is responsible for economic analytics.

Table 3: Summary of IoT Applications using Edge Computing.

Use Case	Ref.	Purpose	Limitations
Smart Cities	[54]	• Power monitoring/element control. • Access control and cabinet telemetry. • Event-Based video. • Traffic management. • Connectivity on demand.	• There are not any analytical approach to substantiate their design choices.
	[55]	• A distributed architecture that supports the integration of many infrastructures in smart cities.	• Lack of mobility. • Lack of QoS management.
	[56]	• A framework perform green survivable for collaborative edge computing in the Wireless-Optical Broadband Access Network (WOBAN) supporting smart cities.	• Lack of scalability.
	[57]	• Fog computing architecture for smart living application	• Require the scalability and mobility support.
	[58]	• Fog computing-based smart urban surveillance (traffic monitoring).	• Huge computational resources.
	[59]	• An OpenStack platform (Stack4Things) based fog computing in smart cities.	• Lack of dynamic discovery of internet objects. • Require the support of multi-protocol in the applications and communications level.
	[60]	• A multi-tier model for smart cities applications in fog computing.	• Real-time data processing become huge in the space based storage.
Smart Homes	[61]	• Smart devices connected via home gateways constitute a local home network to help persons in activities of daily life.	• Require security mechanism.
	[62]	• Home energy management system in fog computing environment.	• Require security mechanism.
	[63]	• Intelligent decision system in fog computing environment to manage residence requests.	• Require security mechanism.
Smart Grid	[64]	• A fog computing based smart grid model for control the balance energy load and power usage.	• Require to ensure the security.
	[65]	• A cloud-fog computing based model for resource management in smart grids.	• Require to ensure the security.
	[66]	• A resource allocation model for fog computing based micro grids to optimize resources in the residential building.	• Require to ensure the security of the system.
Healthcare	[67]	• Telehealth application in fog computing environment.	• Require mechanisms of scalability. • The mobility criteria is not met.
	[68]	• Smart e-health gateway in Fog-IoT to assist healthcare services.	• Require to ensure the security of the system.
	[69]	• Remote patient health monitoring system in fog computing environment.	• Require some mechanisms for securing the data and the fog system.
	[70]	• Fog-Cloud architecture for monitoring the health of person during working hours.	• Require some mechanisms for optimal network load.
	[71]	• IoT based architecture for u-Healthcare monitoring.	• Require security mechanism.
	[72]	• An architecture named FIT for analyzing and processing the clinical data of patients.	• The Scalability is not considered. • The mobility of patients is not met. • The interoperability is not met.
	[73]	• An architecture for applications that offering support to persons influenced by COPD.	• The mobility is not met. • The interoperability is not met.
Smart Vehicles	[74]	• Smart surveillance for vehicle tracking.	• Require to increase the confidence of accuracy for space-time tracks from the systems.
	[75]	• A vehicular architecture named VFC, where the vehicles are used as infrastructures for computation and communication.	• Need for architectural modules to guarantee the scalability.
	[76]	• An architecture for smart vehicles, where the fog server is located at M2M gateways and RSUs.	• Need to guarantee the scalability.

5.5. *Smart Healthcare Systems*

In healthcare systems [103], data management is one of the sensitive issues, as users' data contain important and private information that need to be analyzed and processed in an efficient and secure way. Remote Patient Monitoring (RPM) [104] allows the monitoring of patients regardless of their location, and their caregivers and families are engaged remotely [105]. The RPM system contains three modules: data acquisition, visualization and diagnostics. To acquire the data, the patient is equipped with sensors (e.g., blood glucose sensor), after acquiring the data it will be sent from the patient's smart phone to the processing unit (diagnostic module) for processing. Finally, the calculated analytics are displayed by the visualization module.

The edge computing guarantees the service efficiency in smart medical systems [106] which is considered as an important technology in the IoT ecosystem. The data analysis of the patient requires an efficient platform that guarantees short response time, security, etc., which is provided by the edge computing architecture.

6. Task Scheduling in Edge Computing

The task scheduling problem in edge and fog computing [120] consists of assigning tasks to fog nodes that are located at the network edge, and providing a high-performance execution for end-users' requests. The objective is to find an optimal exploitation of memory allocation and CPU execution for tasks. In the following, we review the existing task scheduling techniques categorized into different classes:

6.1. *Task Scheduling-based Priority Mechanisms*

Priority-based task scheduling is a mechanism where tasks are scheduled according to the priority order. In the following, we present the existing solutions proposed for priority-based task scheduling in the edge computing environment.

Pham *et al.* [122] presented a task scheduling algorithm in the Fog-Cloud environment. The proposed algorithm performs the scheduling by specifying the priorities of tasks, and determining which node to execute tasks. The obtained results show that the proposed algorithm provides an efficient balance between the cost and performance of task execution compared to other algorithms, such as Dynamic Level Scheduling (DLS) algorithm [143], and Heterogeneous Earliest-Finish-Time (HEFT) algorithm [144]. However, this algorithm did not take into consideration the energy consumption.

Wang *et al.* [119] proposed an approach for task scheduling named HealthEdge, which sets the priorities for different tasks using the collected data of human health status and determines if the said task should run in the local device or in the cloud. Based on real traces of five patients, the performance evaluation shows that HealthEdge can efficiently assign different tasks between the network edge and the cloud., which reduces the task processing, bandwidth and time, as well as increasing the workstation utilization in the local edge. The major drawback of this solution is ignoring the energy consumption.

Choudhari *et al.* [113] proposed a priority levels-based task scheduling algorithm in the fog environment. The fog server processes the tasks sent from the clients to the fog layer if the required resources to perform the tasks are available in the assigned fog server and are satisfied by the fog layer, the tasks processed by one or many fog servers in the fog layer. Otherwise, (no resources available), the tasks are forwarded to the cloud for processing. The performance evaluation results show that this approach reduces the response time and decreases the cost.

6.2. *Task Scheduling-based Optimization Techniques*

Optimization-based task scheduling algorithms allow us to determine an optimal assignment of a lot of tasks submitted to be executed using the lowest number of resources. In edge computing, different solutions have been proposed whose objective is to reduce the resources utilization in the edge layer.

Liu *et al.* [114] proposed an algorithm for task scheduling in the fog environment, namely Adaptive Double fitness Genetic Task Scheduling (ADGTS). This algorithm optimizes the communication cost. In addition, it provides perfect performance compared to the Min-Min algorithm. However, this approach did not take into consideration the energy consumption and did not discuss the complexity of the used model.

Hoang *et al.* [115] proposed a task scheduling algorithm in Fog-based Region and Cloud (FBRC). The data processing happens in both the local regions and/or in the remote cloud servers. The authors formulated an integer program named FBRC-IP for task scheduling and showed that it is an NP-hard problem. Hence, they designed a heuristic algorithm to solve this problem. The results demonstrate the performance of the proposed model in terms of resources utilization and latency. However, this approach did not take into account the energy consumption.

Pham *et al.* [125] proposed the Cost-Makespan aware Scheduling heuristic algorithm (CMaS) in the collaborative cloud and fog environment. The system architecture used composed of three layers: (a) the bottom-most layer that contains IoT devices such as smart-phones and wireless sensor devices, etc. (b) the middle layer that consists of fog computing components which are intelligent fog devices, such as access points, switches and routers, and (c) the upper-most layer that consists of cloud computing components which contain a large number of heterogeneous VMs or cloud nodes. The proposed task scheduling approach is composed of three phases: (i) first, the task prioritizing phase that assigns a priority level to each task, (ii) second, the node selection phase that assigns each task to a processing cloud or fog node in order to achieve an optimal value of the utility function that defines the trade-off between the cloud cost and the schedule length, and (iii) finally, the task reassignment phase that improves the QoS of the system by guaranteeing to satisfy the user-defined deadline. Compared to other algorithms that are Greedy for Cost algorithm (GFC), the well-known HEFT [144], and the CCSH [145], the proposed algorithm is more cost-effective and has better performance. However, this approach did not take account the energy consumption.

Table 4: Improved criteria of each research.

Ref.	QoS Enhanced Metric							Security Concerns	
	Energy	**Latency**	**Delay**	**Bandwidth**	**Memory**	**CPU**	**Cost**	**Security**	**Privacy**
[107] [108] [109] [110] [111]	✓	✗	✗	✗	✗	✗	✗	—	—
[111] [112] [113] [114] [115] [116] [117] [118]	✗	✓	✗	✗	✗	✗	✗	—	—
[119] [114]	✗	✗	✗	✓	✗	✗	✗	—	—
[120]	✗	✗	✗	✗	✓	✗	✗	—	—
[121] [122] [123] [119] [124] [109] [125] [120]	✗	✗	✗	✗	✗	✓	✗	—	—
[126] [127]	✗	✗	✗	✗	✗	✗	✓	—	—
[128] [129] [130] [131] [132] [133]	✗	✗	✗	✗	✗	✗	✗	Data Sec.	—
[134] [130] [135]	✗	✗	✗	✗	✗	✗	✗	Authentication	—
[136] [137]	✗	✗	✗	✗	✗	✗	✗	Network Sec.	—
[138]	✗	✗	✗	✗	✗	✗	✗	—	Collusion Att.
[139]	✗	✗	✗	✗	✗	✗	✗	Spoofing Att.	—
[140]	✗	✗	✗	✗	✗	✗	✗	Malware & DDoS	—
[141]	✗	✗	✗	✗	✗	✗	✗	—	Data
[142]	✗	✗	✗	✗	✗	✗	✗	—	User

Fan *et al.* [124] presented a solution based on Ant Colony Optimization (ACO) for task scheduling in the IoT environment as a multi-level 0-1 Knapsack problem, which is an NP hard problem. In the proposed algorithm, the value of pheromone is placed between hosts and tasks, which enables the proposed algorithm to maximize the net profits. The results of the simulation show that the proposed solution outperforms the existing heuristic with Min-min and FCFS algorithms. Despite the fact that the proposed algorithm improves system performance, it is costly as it requires a lot of resources for computation.

Kabirzadeh *et al.* [109] proposed a hyper heuristic algorithm based on test and select technique for scheduling problems in the fog network. The proposed algorithm is used in intelligent surveillance as a case study. In addition, the authors compared the proposed algorithm to three other algorithms; Genetic Algorithm (GA) [146], Particle Swarm Optimization Algorithm (PSO) [147], Ant Colony Optimization Algorithm (ACO) [148], and Simulated Annealing Algorithm (SA) [149]. The performance evaluation shows the efficiency of the proposed hyper-heuristic in terms of energy consumption and execution time. However, the main drawback is the high computational requirements.

Bitam *et al.* [120] proposed an optimization method called Bees Life Algorithm (BLA). In the first step of BLA, the mobile device sends a request to the fog node located at the network edge. Next, the fog node sends the required parameters and the requested data as a job directly to the administrator node. Afterwards, the administrator node breaks down the received job into a collection of tasks. Here, the bees life algorithm is executed to find the best solution for job scheduling, where each fog node receives its task, executes the job, and sends the results back to the administrator node. Based on the received results, the administrator node waits for the final results. Finally, the final results are sent to the mobile users. The performance evaluation showed that the BLA method is more effective in terms of memory allocation and execution time compared to the Particle Swarm Optimization Algorithm (PSO) [147] and the Genetic algorithm (GA) [146]. However, this approach did not take account the energy consumption.

Wan *et al.* [110] presented an Energy-aware Load Balancing and Scheduling (ELBS) method based on fog computing in smart factory. The authors established an energy consumption model and an optimization function for load balancing of the manufacturing cluster. They used a PSO algorithm [150] to obtain an efficient solution. Then, they used a multi-agent system in order to guide the workload scheduling of the equipment with the task scheduling mechanism. The obtained results generated from the candy packing line showed that the ELBS method provides an optimal load balancing and tasks scheduling. However, this method requires resources such as computation cost.

Zeng *et al.* [121] proposed an optimization of image placement and task scheduling based on mixed-integer non-linear programming (MINLP) problem [151] in a fog computing platform supported by SDN Embedded System (FC-SDES). The computational resources are provided from fog nodes and embedded clients. The storage servers are shared by both computation servers and clients. This approach structures resources locally from embedded clients and fog devices. However, it was unavoidable to use more resources from the cloud as the latter is so intensive and large-scale tasks submitted from clients will take a long time to be processed.

Li *et al.* [107] presented the Task Scheduling of Edge-Cloud System (TSECS) algorithm for task scheduling and resource allocation. TSECS is based on the Markov Decision Process (MDP) and uses real-life data. The authors compared the TSECS algorithm to two other algorithms: a load-balanced algorithm that consists in dispatching the requests to the cloud or edge servers based on serving capacity with the same management of TSECS, and the Best Effort algorithm that allows servers to keep running until the queue is empty. The experiment's results show that the proposed algorithm has less consumption of energy, but it takes a long response time.

Yang *et al.* [108] proposed a Maximal Energy-Efficient Task Scheduling algorithm (MEETS) algorithm for homogeneous fog networks. The authors formulated the optimization problem as a multi-nodes programming. Then, they presented the MEETS algorithm with a low complexity to get an efficient solution for the energy consumption. Finally, they derived the joint time and the task modulation allocations to maximize the efficiency of the consumed energy of the fog networks. The performance evaluation shows that MEETS algorithm achieves a better energy efficiency. However, the proposed algorithm requires a lot of computation resources.

6.3. Other Task Scheduling Techniques

Different efforts have been proposed to address the task scheduling issue in edge computing. Cardellini *et al.* [112] evaluated a distributed QoS aware scheduler for data stream processing, that is operating in a fog computing platform. The authors introduced different components: (a) the worker monitor component: is responsible for obtaining the outgoing and incoming data rate for each computing component (executor) that executes a set of tasks on the fog node. (2) the QoS monitor that estimates the QoS, and is responsible for obtaining availability, inter-node information and intra-node utilization. Then this information is forwarded to the distributed adaptive scheduler, and (3) the adaptive scheduler that runs a loop iteration periodically and verifies each candidate's task to be executed. This algorithm enhances the run-time adaptation capabilities in the system and improves the application performance. Nevertheless, some instability caused by complex fog topologies which decreases the data stream processing application's availability.

Verma *et al.* [123] proposed a Real-Time Efficient Scheduling and load balancing algorithm, namely RTES, for task scheduling and load balancing in a fog computing platform. The results of the evaluation performance show that the RTES algorithm achieves a short execution time and completes tasks before their deadline compared with other algorithms like the priority algorithm [152], FCFS algorithm [153], and Multi Objective Tasks scheduling algorithm [154]. However, the proposed approach did not take the energy consumption into consideration.

Various and efficient scheduling mechanisms have been proposed for edge computing that aim to guarantee a short time for task processing and efficient resource allocation. Therefore, the key problem is the decision on where the submitted tasks can be scheduled in an efficient way which is offered by different proposed task scheduling techniques as shown in this section. Besides, the future can be more prosperous for task scheduling issues at the network edge that becoming an important challenge for future networks.

7. SDN/NFV Based Edge Computing

The concept of SDN [155] and NFV [156] has emerged as a popular mechanism for management and virtualization of network services and functions through an abstraction of functionality on a large cluster of devices in the network. The concept of SDN/NFV can be applied to edge computing at a more abstract level by allowing a centralized control, storage of network resources in the edge infrastructure.

Truong *et al.* [116] proposed an architecture called FSDN by merging fog computing and SDN technology in a vanet network. The FSDN VANET architecture is composed of the following components: SDN controller which is the main intelligence responsible for the control of all the network. It also works as a fog orchestration for the fog layer, SDN wireless nodes that represent the data plane elements, SDN road-side-unit which is a fog device, SDN road-side-unit controller which is a cluster of RSUs controlled by the SDN controller and is responsible for the forwarding of data and storing local road system information, and a cellular base station which is controlled by the SDN controller and is responsible for conveying data and carrying voice calls as well as a local intelligence and a fog device. The authors claimed that the proposed architecture optimizes the resources utility and reduces latency. However, this work did not present any experimental results or theoretical formulation to validate its architecture.

Huang *et al.* [117] proposed an SDN-based QoS provisioning mechanism for fog computing advanced WSNs, which realizes a dynamic QoS configuration. The fog nodes receive data sent from sensors then leave the QoS provisioning and process the data content to the SDN controller. The performance evaluation results show that the proposed mechanism achieves a great performance by improving latency. Nevertheless, this work did not describe the effectiveness of the mechanism in large-scale networks.

Liang *et al.* [111] proposed an architecture called software defined and virtualized RANs (SDVRANs) with fog computing. The SDN is used to split up the control plane and the data plane. Furthermore, the network virtualization allows to share network resources to diverse applications. Moreover, The authors presented an example of SaaS named *OpenPipe*, they also used a hybrid control model in order to support the fog computing in SDN. The effectiveness of the proposed network architecture based on a lab demo show that this architecture can fulfill a better performance in terms of low overheads, low latency, and less energy consumption. However, this work did not provide explanations of the use of this technology in the future virtualized networks.

Tomovic *et al.* [118] proposed an architecture for IoT which combines fog computing and SDN. The fog computing platform resolves the problem of latency, while SDN allows sophisticated mechanisms for resource management and traffic control by a centralized control plane. The proposed IoT architecture allows a high level of scalability and low latency. However, this work does not present any theoretical or experimental results to validate its architecture.

The SDN paradigm based on fog computing can solve some important issues, such as irregular connectivity. Furthermore, the NFV decouples the network function by abstraction and virtualization technologies which can notably improve the telecommunication service provisioning flexibility.

8. Security & Privacy in Edge Computing

Edge computing uses a wide range of communication technologies including mobile data acquisition, peer-to-peer networking, WSNs, mobile signature analysis and so on. It is necessary to apply advanced security mechanisms, such as cryptography techniques to secure edge computing, it may be at the system design level or network architecture level.

8.1. Security Efforts

Al Hamid *et al.* [128] proposed a security model to maintain the privacy of medical data in fog environment based healthcare systems. The authors used a one-round authenticated key protocol based on bi-linear pairing cryptography which allows a session key to be generated between participants and data to be exchanged between them securely. The model ensures the security of private data by allowing participants to communicate between themselves securely. However, the key produced by the proposed protocol is static, which may have other security issues.

Similarly, Abdul *et al.* [134] presented a security mechanism to face images using zero watermarking and visual cryptography in the edge environment. The advantage of this mechanism is the copyright protection and authentication of different multimedia content. Although this work is practical, it did not describe how the proposed mechanism could secure the system against different attacks.

Mukherjee *et al.* [136] proposed an end-to-end middleware for IoT security in a cloud-fog environment. The proposed middleware is composed of two main components: a flexible security module that allows users to configure the required higher security and an intermittent security module to allow users to reuse an encrypted session from the recent past. The obtained results show that the proposed middleware can guarantee the security in the communications. However, some attacks can affect the security such as Distributed Denial of Service (DDoS) and man in the middle (MITM).

Diro *et al.* [137] proposed a fog computing based publish-subscribe lightweight protocol based on Elliptic Curve Cryptography (ECC) that has four security requirements: scalability, integrity/confidentiality, authentication and performance. The proposed security scheme provides lower resource usage and shorter key lengths. In addition, it provides lower overheads and better scalability compared to RSA. However, the proposed scheme has some issues, such as the choice of a suitable curve.

Huang *et al.* [129] proposed a secure data access control scheme with computation outsourcing and cipher text update in the fog computing environment. All the sensitive data for the owner is encrypted, then outsourced to the cloud for storage. Only the user that has the attributes that satisfy the access policy can directly decrypt the ciphertext. Similarly, only the user that has the attributes that satisfy the update policy can renew the stocked ciphertext. Although The results of experiments' illustrate that the proposed scheme can achieve data access control and secure cipher text, the proposed scheme does not support the efficiency in data search.

Hu *et al.* [130] proposed security and privacy mechanisms in face resolution and identification framework based on fog computing. They designed a data integrity/encryption scheme and an authentication scheme to meet the demands of integrity, availability and confidentiality in the face identification and resolution processes. The performance evaluation results prove that the proposed scheme guarantees the security of the system and the preservation of privacy. However, the proposed scheme requires a large amount of computation resources.

Cui *et al.* [135] proposed an edge-computing concept for message authentication in vanet networks. The proposed solution resolves different issues, such as the failure to search for invalid messages, redundant authentication by centralizing all the authentication tasks of vehicles on RSUs and the Edge Computing Vehicles (ECVs). The performance evaluation shows that the proposed scheme can rapidly identify valid and invalid messages when attacks are carried out. However, a MITM attack can affect the proposed scheme.

He *et al.* [131] proposed a data security storage model for fog computing, namely FCDSSM. They included in their architecture a control layer (top layer), authentication service layer (medium layer) and a data storage layer at the bottom. This model permits the integration of data storage and security mechanism to be achieved in large-scale IoT applications. The numerical results prove that the FCDSSM model has a better scalability and it can be adapted to the security of big data in large-scale IoT applications. However, the proposed model did not protect the system against data-clone attacks that can affect the storage system by the duplication of data, which can use up a massive amount of storage capacity.

He *et al.* [139] proposed a method to detect GPS spoofing attacks using a combination of visual sensors and Inertial Measurement Units (IMU) of unmanned aerial vehicles (UAVs) [157]. The method is demonstrated using drones. The obtained results prove the efficiency of the proposed method in terms of detected GPS spoofing. Nevertheless, there is a risk of MITM attack.

Alharbi *et al.* [140] proposed a fog-based security system namely FOCUS that aims to protect the IoT against a dangerous attacks named malware cyber-attacks. The FOCUS system uses a Virtual Private Network (VPN) to secure the communication channel from the IoT devices. Also, FOCUS protects the VPN server against DDoS attacks by adopting a challenge-response authentication, which can enhance the security of the IoT system. The results of experiments' validate the efficiency of the FOCUS system which can effectively filter out the different malicious attacks with small network bandwidth consumption and low response time. However, this work needs to be verified in a real fog computing framework.

Ni *et al.* [132] proposed a fog-assisted mobile crowd-sensing (Fo-MCS) framework that tends to improve the precision of task allocation with the help of fog nodes. Moreover, they proposed a fog-assisted secure data deduplication scheme, namely Fo-SDD, in order to reduce the overhead and allow the fog nodes to detect and remove the replicate data in sensing reports and which provides a high level of security against duplicate-replay attacks and brute force attacks. Nevertheless, this work requires some security mechanisms against Denial-of-Service and MITM attacks.

8.2. Privacy Efforts

Wang *et al.* [138] proposed a privacy-preserving scheme with differential privacy levels in the fog computing environment, called privacy-preserving content-based subscribe/publish scheme (PCP) with differential privacy in fog environment, which can efficiently achieve a protection against collusion attacks. However, the authors did not provide a deficiency in the real scenario in terms of reducing the availability of the aggregated data streams.

Okay *et al.* [141] proposed an additive privacy based Secure Data Aggregation scheme for Fog Computing based Smart Grids (FCSG) which ensures data privacy based on homomorphic encryption [158]. This encryption scheme enables users to perform a different operations on encrypted data without affecting the privacy of the data. The performance results show that the proposed scheme ensures low communication and storage, as well as ensuring end-to-end confidentiality and guaranteeing the privacy of collected data. However, the proposed scheme requires the system to be protected against some dangerous attacks, like MITM.

Wang *et al.* [142] proposed a high-level privacy protection mechanism for the Internet of medical things (IoMT) in the fog computing environment, called Fog-based Access Control Model (FACM), which deploys an access control layer at the fog server. The proposed solution offers a high-level privacy protection with a short execution time. However, the algorithm used requires high computational resources.

In a nutshell, the security and privacy aspects in edge computing is an important element that must be developed in a critical way, to ensure the security against malicious and dangerous nodes/attacks that threaten the fog system functionalities and the private of data and end-users.

9. Blockchain in Edge Computing

The Blockchain [159] is a technique for information transmission and storage without control organs. Technically, it is a distributed database, its information sent by the users and the internal links to the database are checked then grouped in blocks at uniform time intervals, the whole being secured by cryptography and consequently forming a chain. By extension, a Blockchain maintains a list of records protected from modification or falsification by storage nodes, so it is a distributed, secure registry of all transactions made since the startup of the distributed system.

The Blockchain was initially designed for the cryptocurrencies such as Bitcoin [160] and it is used for several sectors like intelligent transportation, agriculture and the Internet of energy. Moreover, the IoT can be combined with the Blockchain technology [161] in several domains such as healthcare, SDN and Smart Electric Vehicles, which allow the IoT to benefit from the decentralized resource management, lower operational cost, robustness against attacks and threats and so on. Moreover, the Blockchain can be used with the edge computing [162] in order to facilitate communications between the edge nodes and the IoT devices which enhance the efficiency of the edge computing based IoT networks.

Jiao *et al.* [163], proposed an edge computing model in a mobile Blockchain network. Where the mining process of miners is offloaded to the edge computing service provider (ECSP). Moreover, a combinatorial auction-based pricing approach is used for the allocation of edge resources to the miners. The proposed model maximizes the social welfare and also guarantees the incentive compatibility (IC). Nevertheless, the ECSP revenue has not considered.

Luong *et al.* [126], developed an optimal auction based on the deep learning [164] for resource allocation in the edge computing environment based mobile Blockchain networks, they constructed a multi-layer neural network architecture that first performs some transformations of the miners' bids, then calculates the allocation and the conditional payment rules. They designed the neural networks data training by using the miners' valuations, then using the training data, they trained the neural networks by tuning parameters to enhance the revenue of the ECSP. The obtained results showed that the proposed scheme achieved a higher revenue of the ECSP compared to the baseline scheme. However, only one unit of resource is considered in the auction.

Xiong *et al.* [127] considered the edge computing as a resource management for mobile Blockchain applications where the mining process can be offloaded to the ECSP because the mining process requires the proof-of-work puzzle to be solved, which needs large computation resources which are not adopted in the mobile applications [165]. They analyzed two pricing schemes for miners, a uniform pricing and discriminatory pricing, then they formulated a Stackelberg model to study the maximization of the ECSP revenue and the benefit of miners. The performance evaluation showed that the ECSP achieved the maximum revenue under the uniform pricing, moreover, the discriminatory pricing helps the ESP to achieve a greater profit by encouraging a higher demand of service from miners. However, this work only addressed the revenues of the ESP, it requires other quality of service (QoS) metrics.

Casado *et al.* [133], proposed a Blockchain-based architecture with a view to improve the data security, it consists of the following layers: an IoT layer, a Blockchain layer and an edge computing layer. Moreover, they proposed a self-organized and distributed algorithm based on the game theory executed in the edge computing layer where it is applied on the data collected by the IoT devices whose objective is to improve the false data detection and the data quality. However, the complexity of the algorithm is missing from this work.

The integration of Blockchain with IoT and the edge computing offers many benefits for the future applications in the Internet, such as data security, efficient energy consumption and of course efficient services. Moreover, the Blockchain guarantees the security of the interactions in the system and the users' privacy, in addition the edge computing provides a distributed computing model for the connected devices to execute their tasks, also it offers a rapid and efficient processing and computing services to the Blockchain such as the mining process.

Table 4 outlines the improved criteria of each research.

10. Simulation Tools

In this section, we present the most recent simulators and tools used for cloud, fog and edge environment. Table 5 outlines the main characteristics of each simulator.

10.1. CloudSim

[167] is a cloud-based computing simulator. Developers defined all functionalities of the simulator and how CloudSim can support the cloud components, such as resource provisioning policies, data centers, and virtual machines. CloudSim can support simulation with specific CPU parameters used for specific VMs, as well as different policies (time-shared and space-shared). In addition, it supports the simulation of cloud computing scenarios, the simulation of large-scale virtualized servers and supports the user-defined resources allocations. However, it does not support the cellular network models, and it requires the network topology to be displayed in a visual tool.

10.2. CloudAnalyst

[166] is a cloud-based simulation tool that uses the Java platform and extends the functionality of CloudSim. CloudAnalyst can be used to simulate cloud applications with different deployment scenarios. It allows the description of applications including differences in formation, such as location of data centers and users, the number of resources and users, the processing and response time of all requests generated. In addition, it allows a Graphical User Interface (GUI) to be used and defines the simulation settings with a high degree of flexibility and configurability. Also, it offers the repeatability of experiments. However, this simulator does not support the cellular network models and the user device mobility.

10.3. WorkflowSim

[168] is an extension of CloudSim. It enables researchers to study and evaluate the performance of their optimization techniques with more simplicity and accuracy, it supports different systems configuration such as the failure and monitoring management, multiple overhead models such as workflow engine delay and data transfer delay. In addition, it supports different optimization techniques such as job retry, clustering, scheduling, partitioning, etc. However, the simulator does not support the cellular network models , and it requires the network topology to be displayed in visual tool.

10.4. CactoSim

[169] is a simulator developed in the CACTOS project [2] and it is a set of tools for studying and analyzing the applications and infrastructure performance. CactoSim is used to study the efficiency of different optimization techniques for the cloud and for operations decision. However, this simulator doesn't support the cellular network models , the user device mobility.

10.5. CloudReport

[170] is an extension of cloudSim, which is able to use a graphical interface for simulating distributed environments of cloud computing. It uses the Cloudsim engine as a core component and it provides a GUI to allow users to set the computational hosts, amount of RAM, processing capacity, available bandwidth and the scheduling algorithms. In addition, it provides detailed statistics for simulation output results including energy consumption, CPU utilization, etc. However, this simulator does not support the cellular network models .

10.6. CloudSimDisk

[171] is an extended version of the CloudSim toolkit that focuses on simulating and modeling the energy and storage hardware in the cloud-based infrastructures. It includes algorithms of disk management, energy-aware storage in data centers and power models of HDD. Nevertheless, it does not support the cellular network models , the scalability of complex systems and it requires the network topology to be displayed in a visual tool.

10.7. CloudSim Plus

[172] is extensible simulator that enables the simulation of cloud application services. It allows researchers to focus on the design issues, regardless of the details related to cloud services and infrastructure. In addition, it provides interfaces and classes to allow the implementation of heuristics, such as Simulated Annealing, Ant Colony Systems, Tabu Search and so on. Also, it supports the process of the Google Cluster Data trace [183]. However, it does not support the cellular network models and it requires the network topology to be displayed in a visual tool.

[2] CACTOS Project: http://cactos-cloud.eu

Table 5: Summary of Simulator Tools.

Simulators	Language(s)	Portability	Distributed Arch.	Persistence Type	Web API	GUI	Documentation	License(s)	Result Format
CloudAnalyst [166]	Java	✓	✗	XML	✗	✓	✓	Apache 2, No data	PDF
CloudSim [167]	Java	✓	✗	Yaml	✗	✗	✓	Apache 2	Text
WorkflowSim [168]	Java	✓	✗	Classes (Java)	✗	✗	✓	Apache 2, LGPL 3	Text
CactoSim [169]	Java	✓	✗	Ecore	✗	✓	✓	EPL, GPL, Apache 2	CSV, EDP2
CloudReports [170]	Js, Java	✓	✗	SQLite Data Base	✗	✓	✓	Apache 2, GPL 3	Text, Javascript
CloudSimDisk [171]	Java	✓	✗	Classes (Java)	✗	✗	✓	Apache 2, LGPL 3	Text, XLS
CloudSim Plus [172]	Java	✓	✗	Classes (Java)	✗	✗	✓	GNU GPLv3	Text
GreenCloud [173]	C++, TCL, Shell	✗	✗	TCL	✓	✓	✓	GPL	Dashboard plots
iCanCloud [174]	C/C++, Shell	✓	✗	NED	✗	✓	✗	GPL 3, GNU, Academic	Text
EdgeCloudSim [175]	Java	✓	✗	XML	✗	✗	✓	GNU GPLv3	CSV
IfogSim [176]	Java	✓	✗	JSON	✗	✓	✗	Apache 2, No data	PDF, XLSX
MyIfogSim [177]	Java	✓	✗	JSON	✗	✓	✗	Apache 2, No data	PDF, XLSX
Yafs [178]	Python	✓	✗	DAG	✗	✗	✓	MIT License	CSV
DEVS [179]	C++	✓	✗	.ma File	✗	✗	✓	MIT License	Text
NetSim [180]	C/C++	✗	✗	Netsim, XML	✗	✓	✓	Commercial	Text, CSV
EmuFog [181]	Java	✓	✗	JSON	✗	✗	✓	MIT License	Text
Fogbed [182]	Python	✗	✗	JSON	✗	✗	✓	Apache 2	Text

10.8. GreenCloud

[173] is a simulator for cloud computing environment. It provides a detailed modeling of the consumed energy by the network devices, such as network switches and servers. In addition, it allows the development of novel solutions in monitoring, workload scheduling and resource allocations. It does not support the integration of cellular network components.

10.9. iCanCloud

[174] is a simulation tool aiming to simulate cloud computing systems. Its objective is to predict the tradeoffs between performance and cost. In addition, it allows uni-core and multi-core systems to be simulated quickly, provides a fast, easy and flexible tool to obtain results quickly. However, It does not support the integration of cellular network components.

10.10. EdgeCloudSim

[175] contains five main modules: a core simulation module which is responsible for loading and running scenarios from the configuration file and saving simulation result into files; a networking module that handles the transmission delay in WAN and WLAN for upload and download data; an edge orchestrator module which is responsible for deciding how and where to handle client request; a mobility module which is responsible for updating the location of mobile nodes; and a load generator module that acts as generator of tasks. It provides a simulation environment for edge computing scenarios and supports different mobility and network models, such as WLAN and WAN. However, it requires the task migration between the Cloud or Edge VMs, executing tasks on mobile devices, an energy consumption model for datacenters and mobile devices. In addition, a model for network failure caused by congestion, the distance among mobile devices and access points.

10.11. IFogSim

[176] is a fog-based computing simulation tool that supports the evaluation of resources, such as energy consumption, network congestion, latency and operational cost. This tool simulates cloud datacenters, edge devices, and components of fog computing architecture that may contain IoT sensors and actuators. IFogSim is an event-based implementation, that aims to facilitate the representation of the network topology with a GUI with the capability to draw fog devices, actuators, connecting

link and sensors. Nevertheless, it does not support the mobility of edge servers and mobile devices and the migration of virtual machines.

10.12. MyiFogSim

[177] is an extension of IFogsim that allows the migration of virtual machines among cloudlets through the support of mobility. The implementation of these simulators add the following: Coordinate that designates the place of each entity (x,y), ApDevice that adds the responsibilities and features of a wireless access point, MobileDevice that separates the concerns of user devices and fog servers to keep the features of IoT devices and users in their own class, MobileSensor that represents a set of sensors within a user device, MobileActuator to represent actuators, MigrationStrategy that models the migration strategy and MigrationPolicy which is used to implement the proposed migration policies. However, it does not support the execution of tasks in mobile devices.

10.13. YAFS (Yet Another Fog Simulator)

[178] is a python based simulation library for edge, fog and cloud computing ecosystems and which enables network design, billing management, allocation of resources and a dynamic control of network topology. It has robust, lightweight and highly configurable tools based on a discrete event simulator library and complex network theory. YAFS consists of seven classes only that offer absolute control by the implementation of several environment characteristics and customized policies. It needs to support the cellular network models, and a visual tool for displaying the network topology.

10.14. Discrete Event System Specification (DEVS)

[179] is used as a simulator for fog system, it consists of two layers which separate the simulator from the model. Furthermore, DEVS uses messages for communication among models. DEVS supports high level systems architecture like fog and cloud systems. It does not support the cellular network models for 3G, 4G, 5G, and 6G, in addition, it does not allow the execution of tasks in mobile devices.

10.15. NetSim

[180] is a network simulator and emulator that allows large-scale networks to be designed and implemented, such as 5G, IoT, VANET and fog computing. It offers a GUI for topology displaying, results dashboard that presents all statistics of the scenarios implemented by users, in addition it can display the packet animation to visually understand protocol working. However, it does not support the mobility of fog servers.

10.16. EmuFog

[181] is an emulation framework enables the design of fog computing systems and the emulation of real workloads and different applications scenarios. EmuFog allows researchers to design and implement the network topology of their use cases. In addition, it embeds fog nodes in the network topology and runs docker-based applications on different nodes connected in the network. However, it needs to support the cellular network models, and a visual tool for displaying the network topology.

10.17. Fogbed

[182] is a framework and toolset for the prototyping of fog environments. It enables the deployment of fog nodes as containers in different network configurations. Its design meets the flexible setup, compatibility with real technologies and allows the experiments and testing of fog computing components through standard interfaces. Nevertheless, it does not emulate the reliable distributed algorithms, also it does not support the mobility of edge servers and mobile devices.

11. Future Research Guidelines & Directions

Although the aforementioned advantages and existing solutions, various issues and challenges for a full integration of edge and fog computing on top of IoT applications. In the following, we discuss the major challenges, highlight new ideas and guidelines that need to be addressed seriously by the research community.

11.1. Scalability

Scalability is an indispensable factor for any network architecture in which the system must manage a wide number of demands, requests and services regardless of the growing number of clients, e.g. mobile devices in the edge network. Recently, the number of different connected things has increased rapidly, in which it may interrupt services and their quality and creates bottlenecks in the network caused by the enormous quantity of data generated by the connected things [6]. For this reason the edge servers should guarantee the scalability of the service in the fog layer by applying some mechanisms such as server clusters and load balancing.

11.2. High mobility support

Mobility is a very important aspect in the IoT due to the fact that most of the connected things, such as mobile devices, vehicles, and drones are highly mobile, which is a reason for frequent link failure among devices and edge of network. This problem leads to a decrease in the QoS and the security of the fog system, the high mobility device support is a very important issue and needs to be addressed in future generations of networks [184, 185]. For this reason, it is necessary to propose some algorithms, such as the prediction of link failures to ensure the path stability, detection of obstacles, etc.

11.3. Energy management

The fog computing architecture consists of different distributed systems, therefore the energy consumption is expected to be high, which increase the costs. Thus, a lot of works need to address this issue by optimizing and developing a new effective energy protocol in the fog computing systems, specifically in the virtual and the Adhoc fog systems, such as network and computing resource optimization, reliance on environmentally friendly energies (renewable energy) and so on [186]. And on the other hand, at the end users' level, and because are devices using battery which is limited energy save, this will cause some problems, such as the disconnection of devices during the upload/download of data in/from fog devices. Therefore, it is a big challenge to provide solutions for this problem, such as providing sources for charging devices in the streets, reliance on friction (movement of cars and people) as a power source for devices.

11.4. Security & Privacy

The security issue is one of the most important problems and which needs to be studied carefully in edge & fog computing networks. For this there are different mechanisms such as cryptography, hash functions, etc. It is necessary to be used to guarantee the security of communications in the IoT network with fog computing environment.

The privacy leakage of end users' information in the IoT networks, like location, usage and data, is attracting the attention of unauthorized users to threaten the privacy of end users, such as location privacy issue because most IoT devices are location-based, this allows the adversary to change the location of end users, which will affect the decision of fog servers that are based on the end users' location. The future work needs to address seriously the privacy issue by ensuring the security against MITM attacks and use the infrastructure of public key for data block encryption.

11.4.1. Authentication

Authentication is essential in the security of IoT devices. Regrettably the IoT devices don't have enough capacity such as CPU and memory to execute the cryptographic operations used for authentication. The IoT devices can use an outsourced storage and computations capacity in the fog devices to execute the authentication protocol. Therefore, the fog server will be considered as an authentication server of the IoT devices.

11.4.2. Data Protection

The huge volume of data generated by IoT devices must be protected at the communication level and at the processing level at the network edge or the cloud. The risk of data replication/sharing attack, data altering attacks, data loss at the edge or cloud level needs to be resolved to ensure data integrity by the encryption mechanisms, backup and recovery of data, policy enforcement, network monitoring.

11.4.3. Preventing cache attacks

The edge servers cache the contents sent from mobile devices and then forward to the cloud the cached data for storage. The cached data can be threatened by some malicious programs that can change or erase the cached data. For this, the security of data caching in fog computing is an important issue. Moreover, as a future challenge the data cached in fog servers need to be protected by several mechanisms, such as hash functions, cryptography and also ensure the data recovery and backup.

11.5. Machine Learning & Deep Learning in EC

Machine learning is a promising approach and an important methodology to resolve different problems, it helps the IoT in many fields [187] such as smart cities, smart homes, also in smart healthcare systems, and it can also used in game programming such as puzzles and video games.

The machine learning is structured in three categories: *Supervised learning* where the training set consists of input vectors (samples) together with their labels also known as corresponding appropriate target vectors, *Unsupervised learning* where for the training set no labels are required, and finally, *Reinforcement learning*, that deals with learning appropriate actions problems, in order to maximize the payoff, furthermore, the multi-agent reinforcement learning (MADRL) [3] [4] is a learning paradigm to control a system with a view to maximizing an objective over the long term. Thus, the Multi-Agent Systems interacting in mixed competitive and cooperative environments, it can be used in a variety of domains like economics, robotics, games, etc.

Different works have been proposed for machine learning and deep learning in Edge-IoT environment [188], where the machine learning provide a great number of services to the network edge to enhance the efficiency of the edge services. We believe that the future Edge-IoT architectures need to integrate distributed machine learning algorithms such as Deep Multi-Agent Reinforcement Learning to automatically make decisions in large scale networks.

11.6. Simulation Platform

Simulation is a process of modeling a real scenario using a mathematical formula and implementing it using programming language. The advantage of using simulation is because it helps us to understand the system and apply experiments at a cheap cost. The experiments and testing in real edge computing infrastructure require a lot of resources (finance and effort) which-can not be possible for all researchers. Instead of that, developing a simulation platform for edge computing will encourage researchers to implements their new ideas. For this, the future simulation platform must support the high mobility of end users and fog servers, also ensure task migration and provide a model of energy consumption for end users and fog servers. In addition a visual tool and detailed guide for using the simulator.

12. Conclusion

With the huge development of IoT networks, edge computing is becoming a solution to the complex challenges and difficulties of managing billions of connected devices/sensors, and the huge computing resources that they use. Compared to the cloud computing technology, the edge computing will offer a large data computation and storage at the network edge which is required by future applications, such as smart homes, smart cities, smart vehicles and so on. This manuscript presents a survey on edge computing for IoT, with recent research activities like task scheduling, SDN/NFV, security and privacy and the Blockchain. In addition, the future applications and use cases

[3]DeepMind: https://deepmind.com/

[4]OpenIA: https://openai.com/

are detailed, furthermore the most used simulators are summarized and finally we identify the open research challenges and highlight future research directions.

References

[1] P. K. Senyo, E. Addae, R. Boateng, Cloud computing research: A review of research themes, frameworks, methods and future research directions, International Journal of Information Management (IJIM) 38 (2018) 128–139. doi:10.1016/j.ijinfomgt.2017.07.007.

[2] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet of Things Journal (IoT-J) 5 (2018) 439–449. doi:10.1109/JIOT.2017.2767608.

[3] S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in: Workshop on mobile big data, ACM, 2015, pp. 37–42. doi:10.1145/2757384.2757397.

[4] M. Chiang, T. Zhang, Fog and IoT: An overview of research opportunities, IEEE Internet of Things Journal (IoT-J) 3 (2016) 854–864. doi:10.1109/JIOT.2016.2584538.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: Vision and challenges, IEEE Internet of Things Journal (IoT-J) 3 (2016) 637–646. doi:10.1109/JIOT.2016.2579198.

[6] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Communications Surveys & Tutorials (COMST) 20 (2017) 416–464. doi:10.1109/COMST.2017.2771153.

[7] P. Hu, S. Dhelim, H. Ning, T. Qiu, Survey on fog computing: architecture, key technologies, applications and open issues, Journal of Network and Computer Applications 98 (2017) 27–42. doi:10.1016/j.jnca.2017.09.002.

[8] J. Ni, K. Zhang, X. Lin, X. S. Shen, Securing fog computing for internet of things applications: Challenges and solutions, IEEE Communications Surveys & Tutorials (COMST) 20 (2017) 601–628. doi:10.1109/COMST.2017.2762345.

[9] A. C. Baktir, A. Ozgovde, C. Ersoy, How can edge computing benefit from software-defined networking: a survey, use cases, and future directions, IEEE Communications Surveys & Tutorials (COMST) 19 (2017) 2359–2391. doi:10.1109/COMST.2017.2717482.

[10] M. Mukherjee, L. Shu, D. Wang, Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges, IEEE Communications Surveys & Tutorials (COMST) (2018). doi:10.1109/COMST.2018.2814571.

[11] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the Internet of Things, IEEE Access 6 (2018) 6900–6919. doi:10.1109/ACCESS.2017.2778504.

[12] R. Mahmud, R. Kotagiri, R. Buyya, Fog computing: A taxonomy, survey and future directions, in: Internet of Everything, Springer, 2018, pp. 103–130. doi:10.1007/978-981-10-5861-5_5.

[13] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, S. A. Malik, Fog/Edge Computing-based IoT (FECIoT): Architecture, Applications, and Research Issues, IEEE Internet of Things Journal (IoT-J) (2018). doi:10.1109/JIOT.2018.2875544.

[14] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, A. Ahmed, Edge computing: A survey, Future Generation Computer Systems (FGCS) 97 (2019) 219–235. doi:10.1016/j.future.2019.02.050.

[15] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, B. Weihl, Globally distributed content delivery, IEEE Internet computing (IC) 6 (2002) 50–58. doi:10.1109/MIC.2002.1036038.

[16] S. Yangui, P. Ravindran, O. Bibani, R. H. Glitho, N. B. Hadj-Alouane, M. J. Morrow, P. A. Polakos, A platform as-a-service for hybrid cloud/fog environments, in: IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), 2016, pp. 1–7. doi:10.1109/LANMAN.2016.7548853.

[17] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, Y. Wang, A Survey of Internet of Things Communication using ICN: A Use Case Perspective, Computer Communications (ComCom) 142-143 (2019) 95–123. doi:10.1016/j.comcom.2019.05.010.

[18] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named data networking (NDN) project, Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC 157 (2010) 158.

[19] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, A. Ksentini, Named Data Networking in Vehicular Ad hoc Networks: State-of-the-Art and Challenges, IEEE Communications Surveys and Tutorials (2019). doi:10.1109/COMST.2019.2894816.

[20] B. Nour, K. Sharif, F. Li, H. Moungla, Y. Liu, A Unified Hybrid Information-Centric Naming Scheme for IoT Applications, Computer Communications 150 (2019) 103–114. doi:10.1016/j.comcom.2019.11.020.

[21] D. Raychaudhuri, K. Nagaraja, A. Venkataramani, Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R) 16 (2012) 2–13. doi:10.1145/2412096.2412098.

[22] R. K. Chahal, N. Kumar, S. Batra, Trust management in social Internet of Things: A taxonomy, open issues, and challenges, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.10.034.

[23] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Computer Communications (ComCom) 54 (2014) 1–31. doi:10.1016/j.comcom.2014.09.008.

[24] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future generation computer systems (FGCS) 29 (2013) 1645–1660. doi:10.1016/j.future.2013.01.010.

[25] D.-H. Shih, P.-L. Sun, D. C. Yen, S.-M. Huang, Taxonomy and survey of RFID anti-collision protocols, Computer communications (ComCom) 29 (2006) 2150–2166. doi:10.1016/j.comcom.2005.12.011.

[26] B. Sun, Y. Xiao, C. C. Li, H.-H. Chen, T. A. Yang, Security coexistence of wireless sensor networks and RFID for pervasive computing, Computer Communications (ComCom) 31 (2008) 4294–4303. doi:10.1016/j.comcom.2008.05.035.

[27] M. A. Mohandes, Mobile technology for socio-religious events: a case study of NFC technology, IEEE Technology and Society Magazine 34 (2015) 73–79. doi:10.1109/MTS.2015.2396122.

[28] H. Guo, Y. Zheng, X. Li, Z. Li, C. Xia, Self-healing group key distribution protocol in wireless sensor networks for secure IoT communications, Future Generation Computer Systems (FGCS) 89 (2018) 713–721. doi:10.1016/j.future.2018.07.009.

[29] S. Wan, Z. Gu, Q. Ni, Cognitive computing and wireless communications on the edge for healthcare service robots, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.10.012.

[30] H. Ning, Z. Wang, Future internet of things architecture: like mankind neural system or social organization framework?, IEEE Communications Letters (COMML) 15 (2011) 461–463. doi:10.1109/LCOMM.2011.022411.110120.

[31] P. Mell, T. Grance, et al., The NIST definition of cloud computing (2011).

[32] D. Zissis, D. Lekkas, Addressing cloud computing security issues, Future generation computer systems (FGCS) 28 (2012) 583–592. doi:10.1016/j.future.2010.12.006.

[33] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, A. Ghalsasi, Cloud computing—The business perspective, Decision support systems (DSS) 51 (2011) 176–189. doi:10.1016/j.dss.2010.12.006.

[34] G. Mateescu, W. Gentzsch, C. J. Ribbens, Hybrid computing—where HPC meets grid and cloud computing, Future generation computer systems (FGCS) 27 (2011) 440–453. doi:10.1016/j.future.2010.11.003.

[35] M. Cusumano, Cloud computing and SaaS as new computing platforms, Communications of the ACM (CACM) 53 (2010) 27–29. doi:10.1145/1721654.1721667.

[36] C. Pahl, Containerization and the PaaS cloud, IEEE Cloud Computing 2 (2015) 24–31. doi:10.1109/MCC.2015.51.

[37] S. Bhardwaj, L. Jain, S. Jain, Cloud computing: A study of infrastructure as a service (IAAS), International Journal of engineering and information Technology (IJEIT) 2 (2010) 60–63.

[38] E. Cavalcante, J. Pereira, M. P. Alves, P. Maia, R. Moura, T. Batista, F. C. Delicato, P. F. Pires, On the interplay of Internet of Things and Cloud Computing: A systematic mapping study, Computer Communications (ComCom) 89 (2016) 17–33. doi:10.1016/j.comcom.2016.03.012.

[39] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud

computing and Internet of Things: a survey, Future generation computer systems (FGCS) 56 (2016) 684–700. doi:10.1016/j.future.2015.09.021.

[40] G. Neagu, Ş. Preda, A. Stanciu, V. Florian, A Cloud-IoT based sensing service for health monitoring, in: IEEE E-Health and Bioengineering Conference (EHB), 2017, pp. 53–56. doi:10.1109/EHB.2017.7995359.

[41] L. Ismail, H. Materwala, Energy-Aware VM Placement and Task Scheduling in Cloud-IoT Computing: Classification and Performance Evaluation, IEEE Internet of Things Journal (IoT-J) (2018). doi:10.1109/JIOT.2018.2865612.

[42] N. Almolhis, A. M. Alashjaee, S. Duraibi, F. Alqahtani, A. N. Moussa, The security issues in iot-cloud: A review, in: IEEE International Colloquium on Signal Processing & Its Applications (CSPA), 2020, pp. 191–196. doi:10.1109/CSPA48992.2020.9068693.

[43] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet of Things Journal (IoT-J) (2017). doi:10.1109/JIOT.2017.2767608.

[44] B. Nour, S. Mastorakis, A. Mtibaa, Compute-Less Networking: Perspectives, Challenges, and Opportunities, IEEE Network (2020).

[45] M. Aazam, E.-N. Huh, Fog computing: The cloud-iot/ioe middleware paradigm, IEEE Potentials 35 (2016) 40–44. doi:10.1109/MPOT.2015.2456213.

[46] A. M. Goscinski, Z. Tari, I. A. Aziz, E. J. Alzahrani, Fog Computing as a Critical Link Between a Central Cloud and IoT in Support of Fast Discovery of New Hydrocarbon Reservoirs, in: International Conference on Mobile Networks and Management (MONAMI), Springer, 2017, pp. 247–261. doi:10.1007/978-3-319-90775-8_20.

[47] M. Chiang, S. Ha, I. Chih-Lin, F. Risso, T. Zhang, Clarifying fog computing and networking: 10 questions and answers, IEEE Communications Magazine (COMMAG) 55 (2017) 18–20. doi:10.1109/MCOM.2017.7901470.

[48] Y. Pan, P. Thulasiraman, Y. Wang, Overview of Cloudlet, Fog Computing, Edge Computing, and Dew Computing, in: International Workshop on Dew Computing, 2018, pp. 20–23.

[49] M. Satyanarayanan, The emergence of edge computing, IEEE Computer 50 (2017) 30–39. doi:10.1109/MC.2017.9.

[50] S. Yi, Z. Hao, Z. Qin, Q. Li, Fog computing: Platform and applications, in: IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), 2015, pp. 73–78. doi:10.1109/HotWeb.2015.22.

[51] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: A platform for internet of things and analytics, in: Big data and internet of things: A roadmap for smart environments, Springer, 2014, pp. 169–186.

[52] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Mobile cloud computing (MCC) workshop, ACM, 2012, pp. 13–16. doi:10.1145/2342509.2342513.

[53] B. Negash, A. M. Rahmani, P. Liljeberg, A. Jantsch, Fog Computing Fundamentals in the Internet-of-Things, in: Fog Computing in the Internet of Things, Springer, 2018, pp. 3–13. doi:10.1007/978-3-319-57639-8_1.

[54] M. Yannuzzi, F. van Lingen, A. Jain, O. L. Parellada, M. M. Flores, D. Carrera, J. L. Pérez, D. Montero, P. Chacin, A. Corsaro, et al., A new era for cities with fog computing, IEEE Internet computing (IC) 21 (2017) 54–67. doi:10.1109/MIC.2017.25.

[55] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, Q. Yang, A hierarchical distributed fog computing architecture for big data analysis in smart cities, in: ASE BigData & SocialInformatics, ACM, 2015, p. 28. doi:10.1145/2818869.2818898.

[56] W. Hou, Z. Ning, L. Guo, Green survivable collaborative edge computing in smart cities, IEEE Transactions on Industrial Informatics (TII) 14 (2018) 1594–1605. doi:10.1109/TII.2018.2797922.

[57] J. Li, J. Jin, D. Yuan, M. Palaniswami, K. Moessner, EHOPES: Datacentered Fog platform for smart living, in: IEEE International Telecommunication Networks and Applications Conference (ITNAC), 2015, pp. 308–313. doi:10.1109/ATNAC.2015.7366831.

[58] N. Chen, Y. Chen, S. Song, C.-T. Huang, X. Ye, Smart urban surveillance using fog computing, in: IEEE/ACM Symposium on Edge Computing (SEC), 2016, pp. 95–96. doi:10.1109/SEC.2016.25.

[59] D. Bruneo, S. Distefano, F. Longo, G. Merlino, A. Puliafito, V. D'Amico, M. Sapienza, G. Torrisi, Stack4Things as a fog computing platform for Smart City applications, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016, pp. 848–853. doi:10.1109/INFCOMW.2016.7562195.

[60] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, Y. Zhang, Multitier fog computing with large-scale iot data analytics for smart cities, IEEE Internet of Things Journal (IoT-J) 5 (2018) 677–686. doi:10.1109/JIOT.2017.2724845.

[61] M. Bansal, I. Chana, S. Clarke, Enablement of IoT based context-aware smart home with fog computing, in: Fog Computing: Breakthroughs in Research and Practice, IGI Global, 2018, pp. 251–263. doi:10.4018/978-1-5225-5649-7.ch013.

[62] Y.-D. Chen, M. Z. Azhari, J.-S. Leu, Design and implementation of a power consumption management system for smart home over fog-cloud computing, in: IEEE International Conference on Intelligent Green Building and Smart Grid (IGBSG), 2018, pp. 1–5. doi:10.1109/IGBSG.2018.8393553.

[63] G. P. Rocha Filho, L. Y. Mano, A. D. B. Valejo, L. A. Villas, J. Ueyama, A low-cost smart home automation to enhance decision-making based on fog computing and computational intelligence, IEEE Latin America Transactions (LATAM) 16 (2018) 186–191. doi:10.1109/TLA.2018.8291472.

[64] F. Y. Okay, S. Ozdemir, A fog computing based smart grid model, in: IEEE International Symposium on Networks, Computers and Communications (ISNCC), 2016, pp. 1–6. doi:10.1109/ISNCC.2016.7746062.

[65] S. Zahoor, S. Javaid, N. Javaid, M. Ashraf, F. Ishmanov, M. K. Afzal, Cloud–Fog–Based Smart Grid Model for Efficient Resource Management., Sustainability (2071-1050) 10 (2018). doi:10.3390/su10062079.

[66] A. Fatima, N. Javaid, M. Waheed, T. Nazar, S. Shabbir, T. Sultana, Efficient Resource Allocation Model for Residential Buildings in Smart Grid Using Fog and Cloud Computing, in: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Springer, 2018, pp. 289–298. doi:10.1007/978-3-319-93554-6_26.

[67] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, K. Makodiya, Fog data: Enhancing telehealth big data through fog computing, in: Proceedings of the ASE BigData & SocialInformatics, ACM, 2015, p. 14. doi:10.1145/2818869.2818889.

[68] B. Negash, T. N. Gia, A. Anzanpour, I. Azimi, M. Jiang, T. Westerlund, A. M. Rahmani, P. Liljeberg, H. Tenhunen, Leveraging fog computing for healthcare iot, in: Fog Computing in the Internet of Things, Springer, 2018, pp. 145–169. doi:10.1007/978-3-319-57639-8_8.

[69] P. Verma, S. K. Sood, Fog Assisted-IoT Enabled Patient Health Monitoring in Smart Homes, IEEE Internet of Things Journal (IoT-J) (2018). doi:10.1109/JIOT.2018.2803201.

[70] M. Bhatia, S. K. Sood, Exploring Temporal Analytics in Fog-Cloud Architecture for Smart Office HealthCare, Mobile Networks and Applications (2018) 1–19. doi:10.1007/s11036-018-0991-5.

[71] C. S. Nandyala, H.-K. Kim, From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals, International Journal of Smart Home (IJSH) 10 (2016) 187–196. doi:10.14257/ijsh.2016.10.2.18.

[72] A. Monteiro, H. Dubey, L. Mahler, Q. Yang, K. Mankodiya, FIT A Fog Computing Device for Speech TeleTreatments, IEEE International Conference on Smart Computing (SMARTCOMP) (2016). doi:10.1109/SMARTCOMP.2016.7501692.

[73] O. Fratu, C. Pena, R. Craciunescu, S. Halunga, Fog computing system for monitoring Mild Dementia and COPD patients-Romanian case study, in: IEEE International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), 2015, pp. 123–128. doi:10.1109/TELSKS.2015.7357752.

[74] Z. Xu, H. Gupta, U. Ramachandran, STTR: A System for Tracking All Vehicles All the Time At the Edge of the Network, in: ACM International Conference on Distributed and Event-based Systems, 2018, pp. 124–135. doi:10.1145/3210284.3210291.

[75] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: A viewpoint of vehicles as the infrastructures, IEEE Transactions on Vehicular Technology 65 (2016) 3860–3873. doi:10.1109/TVT.2016.2532863.

[76] S. K. Datta, C. Bonnet, J. Haerri, Fog computing architecture to enable consumer centric internet of things services, in: IEEE International Symposium on Consumer Electronics (ISCE), 2015, pp. 1–2.

doi:10.1109/ISCE.2015.7177778.

[77] K. Abas, K. Obraczka, L. Miller, Solar-powered, wireless smart camera network: An IoT solution for outdoor video monitoring, Computer Communications (ComCom) 118 (2018) 217–233. doi:10.1016/j.comcom.2018.01.007.

[78] S. Sudhakar, V. Vijayakumar, C. S. Kumar, V. Priya, L. Ravi, V. Subramaniyaswamy, Unmanned Aerial Vehicle (UAV) based Forest Fire Detection and monitoring for reducing false alarms in forest-fires, Computer Communications (ComCom) 149 (2020) 1–16. doi:10.1016/j.comcom.2019.10.007.

[79] D. Jiang, The construction of smart city information system based on the internet of things and cloud computing, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.10.035.

[80] H. Li, K. Ota, M. Dong, Learning IoT in edge: deep learning for the internet of things with edge computing, IEEE Network 32 (2018) 96–101. doi:10.1109/MNET.2018.1700202.

[81] B. Tang, H. He, Q. Ding, S. Kay, A parametric classification rule based on the exponentially embedded family, IEEE transactions on neural networks and learning systems (TNNLS) 26 (2015) 367–377. doi:10.1109/TNNLS.2014.2383692.

[82] B. Tang, S. Khokhar, R. Gupta, Turn prediction at generalized intersections, in: IEEE Intelligent Vehicles Symposium (IV), 2015, pp. 1399–1404. doi:10.1109/IVS.2015.7225911.

[83] H. Luo, H. Cai, H. Yu, Y. Sun, Z. Bi, L. Jiang, A short-term energy prediction system based on edge computing for smart city, Future Generation Computer Systems (FGCS) (2019). doi:10.1016/j.future.2019.06.030.

[84] Cisco, Cisco Global Cloud Index: Forecast and Methodology, 2014–2019, 2014.

[85] B. Nour, K. Sharif, F. Li, H. Moungla, A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study, in: IEEE Global Communications Conference (GLOBECOM), Singapore, 2017, pp. 1–6. doi:10.1109/GLOCOM.2017.8255022.

[86] L. Y. Mano, B. S. Faiçal, L. H. Nakamura, P. H. Gomes, G. L. Libralon, R. I. Meneguete, P. R. Geraldo Filho, G. T. Giancristofaro, G. Pessin, B. Krishnamachari, et al., Exploiting IoT technologies for enhancing Health Smart Homes through patient identification and emotion recognition, Computer Communications (ComCom) 89 (2016) 178–190. doi:10.1016/j.comcom.2016.03.010.

[87] D.-M. Han, J.-H. Lim, Design and implementation of smart home energy management systems based on zigbee, IEEE Transactions on Consumer Electronics 56 (2010). doi:10.1109/TCE.2010.5606278.

[88] A. Aliyu, A. H. Abdullah, O. Kaiwartya, Y. Cao, J. Lloret, N. Aslam, U. M. Joda, Towards video streaming in IoT Environments: Vehicular communication perspective, Computer Communications (ComCom) 118 (2018) 93–119. doi:10.1016/j.comcom.2017.10.003.

[89] J. Bhatia, R. Dave, H. Bhayani, S. Tanwar, A. Nayyar, SDN-based realtime urban traffic analysis in VANET environment, Computer Communications (ComCom) 149 (2020) 162–175. doi:10.1016/j.comcom.2019.10.011.

[90] K. Ansari, Cooperative Position Prediction: Beyond Vehicle-to-Vehicle Relative Positioning, IEEE Transactions on Intelligent Transportation Systems (TITS) (2019). doi:10.1109/TITS.2019.2902572.

[91] J. Shi, Z. Yang, H. Xu, M. Chen, B. Champagne, Dynamic Resource Allocation for LTE-based Vehicle-to-Infrastructure Networks, IEEE Transactions on Vehicular Technology (2019). doi:10.1109/TVT.2019.2903822.

[92] L. Noel, G. Z. de Rubens, J. Kester, B. K. Sovacool, The Technical Challenges to V2G, in: Vehicle-to-Grid, Springer, 2019, pp. 65–89. doi:10.1007/978-3-030-04864-8_3.

[93] M. Laroui, B. Nour, H. Moungla, H. Afifi, M. Cherif, Mobile Vehicular Edge Computing Architecture using Rideshare Taxis as a Mobile Edge Server, in: IEEE Annual Consumer Communications Networking Conference (CCNC), IEEE, Las Vegas, NV, USA, 2020, pp. 1–2. doi:10.1109/CCNC46108.2020.9045741.

[94] A. Gregoriades, A. Sutcliffe, Simulation-based evaluation of an invehicle smart situation awareness enhancement system, Ergonomics (2018) 1–28. doi:10.1080/00140139.2018.1427803.

[95] J. Singh, K. Singh, Congestion Control in Vehicular Ad Hoc Network: A Review, in: Next-Generation Networks (NGN), Springer, 2018, pp. 489–496. doi:10.1007/978-981-10-6005-2_49.

[96] K. Qin, A. Sun, M. Xue, K. Zhong, C. Jia, Traffic alert methods in an intersection and vehicle multi-functional ambient light systems, 2018. doi:USPatentApp.15/673,956.

[97] J. Liu, J. Li, L. Zhang, F. Dai, Y. Zhang, X. Meng, J. Shen, Secure intelligent traffic light control using fog computing, Future generation computer systems (FGCS) 78 (2018) 817–824. doi:10.1016/j.future.2017.02.017.

[98] T. Lehr, Smart Cities: Vision on-the-Ground, in: Smart Cities, Springer, 2018, pp. 3–15. doi:10.1007/978-3-319-59381-4_1.

[99] J. A. Jimenez, Smart Transportation Systems, in: Smart Cities, Springer, 2018, pp. 123–133. doi:10.1007/978-3-319-59381-4_8.

[100] N. Komninos, E. Philippou, A. Pitsillides, Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures, IEEE Communications Surveys & Tutorials (COMST) 16 (2014) 1933–1954. doi:10.1109/COMST.2014.2320093.

[101] S. S. Sk, G. Thatiparthi, R. S. Gunturi, Cloud and intelligent based SCADA technology, International Journal of Advanced Research in Computer Science (IJARCS) and Electronics Engineering (IJARCSEE) 2 (2013) pp–293.

[102] M. M. Hussain, M. S. Alam, M. S. Beg, Feasibility of Fog Computing in Smart Grid Architectures, in: International Conference on Communication, Computing and Networking (ICCCN), Springer, 2019, pp. 999–1010. doi:10.1007/978-981-13-1217-5_98.

[103] T. N. Gia, I. B. Dhaou, M. Ali, A. M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease, Future Generation Computer Systems (FGCS) 93 (2019) 198–211. doi:10.1016/j.future.2018.10.029.

[104] M. M. Mahmoud, J. J. Rodrigues, K. Saleem, J. Al-Muhtadi, N. Kumar, V. Korotaev, Towards energy-aware fog-enabled cloud of things for healthcare, Computers & Electrical Engineering 67 (2018) 58–69. doi:10.1016/j.compeleceng.2018.02.047.

[105] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, R. Buyya, HealthFog: An Ensemble Deep Learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in Integrated IoT and Fog Computing Environments, Future Gerenation Computer Systems (2019). doi:10.1016/j.future.2019.10.043.

[106] P. H. Vilela, J. J. Rodrigues, P. Solic, K. Saleem, V. Furtado, Performance evaluation of a Fog-assisted IoT solution for e-Health applications, Future Generation Computer Systems (FGCS) 97 (2019) 379–386. doi:10.1016/j.future.2019.02.055.

[107] S. Li, J. Huang, Energy Efficient Resource Management and Task Scheduling for IoT Services in Edge Computing Paradigm, in: IEEE International Conference on Ubiquitous Computing and Communications, IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA/IUCC), 2017, pp. 846–851. doi:10.1109/ISPA/IUCC.2017.00129.

[108] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, M.-T. Zhou, MEETS: Maximal Energy Efficient Task Scheduling in Homogeneous Fog Networks, IEEE Internet of Things Journal (IoT-J) (2018). doi:10.1109/JIOT.2018.2846644.

[109] S. Kabirzadeh, D. Rahbari, M. Nickray, A hyper heuristic algorithm for scheduling of fog networks, in: IEEE Conference of Open Innovations Association (FRUCT), 2017, pp. 148–155. doi:10.23919/FRUCT.2017.8250177.

[110] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, C. Liu, Fog Computing for Energy-aware Load Balancing and Scheduling in Smart Factory, IEEE Transactions on Industrial Informatics (TII) (2018). doi:10.1109/TII.2018.2818932.

[111] K. Liang, L. Zhao, X. Chu, H.-H. Chen, An integrated architecture for software defined and virtualized radio access networks with fog computing, IEEE Network 31 (2017) 80–87. doi:10.1109/MNET.2017.1600027NM.

[112] V. Cardellini, V. Grassi, F. L. Presti, M. Nardelli, On QoS-aware scheduling of data stream applications over fog computing infrastructures, in: IEEE Symposium on Computers and Communication (ISCC), 2015, pp. 271–276. doi:10.1109/ISCC.2015.7405527.

[113] T. Choudhari, M. Moh, T.-S. Moh, Prioritized task scheduling in fog computing, in: Annual ACM Southeast Conference (ACMSE), 2018, p. 22. doi:10.1145/3190645.3190699.

[114] Q. Liu, Y. Wei, S. Leng, Y. Chen, Task scheduling in fog enabled In-

ternet of Things for smart cities, in: IEEE International Conference on Communication Technology (ICCT), 2017, pp. 975–980. doi:10.1109/ICCT.2017.8359780.

[115] D. Hoang, T. D. Dang, FBRC: Optimization of task scheduling in Fog-based Region and Cloud, in: IEEE International Conference on Big Data Science and Engineering, and IEEE International Conference On Embedded Software And Systems (Trustcom/BigDataSE/ICESS), 2017, pp. 1109–1114. doi:10.1109/Trustcom/BigDataSE/ICESS.2017.360.

[116] N. B. Truong, G. M. Lee, Y. Ghamri-Doudane, Software defined networking-based vehicular adhoc network with fog computing, in: IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, pp. 1202–1207. doi:10.1109/INM.2015.7140467.

[117] L. Huang, G. Li, J. Wu, L. Li, J. Li, R. Morello, Software-defined QoS provisioning for fog computing advanced wireless sensor networks, in: IEEE SENSORS, 2016, pp. 1–3. doi:10.1109/ICSENS.2016.7808814.

[118] S. Tomovic, K. Yoshigoe, I. Maljevic, I. Radusinovic, Software-defined fog network architecture for IoT, Wireless Personal Communications 92 (2017) 181–196. doi:10.1007/s11277-016-3845-0.

[119] H. Wang, J. Gong, Y. Zhuang, H. Shen, J. Lach, Healthedge: Task scheduling for edge computing with health emergency and human behavior consideration in smart homes, in: IEEE International Conference on Big Data (Big Data), 2017, pp. 1213–1222. doi:10.1109/BigData.2017.8258047.

[120] S. Bitam, S. Zeadally, A. Mellouk, Fog computing job scheduling optimization based on bees swarm, Enterprise Information Systems (EIS) 12 (2018) 373–397. doi:10.1080/17517575.2017.1304579.

[121] 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 Transactions on Computers (TC) 65 (2016) 3702–3712. doi:10.1109/TC.2016.2536019.

[122] X.-Q. Pham, E.-N. Huh, Towards task scheduling in a cloud-fog computing system, in: IEEE Asia-Pacific Network Operations and Management Symposium (APNOMS), 2016, pp. 1–4. doi:10.1109/APNOMS.2016.7737240.

[123] M. Verma, N. Bhardwaj, A. K. Yadav, Real time efficient scheduling algorithm for load balancing in fog computing environment, Int. J. Inf. Technol. Comput. Sci 8 (2016) 1–10. doi:10.5815/ijitcs.2016.04.01.

[124] J. Fan, X. Wei, T. Wang, T. Lan, S. Subramaniam, Deadline-Aware Task Scheduling in a Tiered IoT Infrastructure, in: IEEE Global Communications Conference (GLOBECOM), 2017, pp. 1–7. doi:10.1109/GLOCOM.2017.8255037.

[125] X.-Q. Pham, N. D. Man, N. D. T. Tri, N. Q. Thai, E.-N. Huh, A cost- and performance-effective approach for task scheduling based on collaboration between cloud and fog computing, International Journal of Distributed Sensor Networks (IJDSN) 13 (2017) 1550147717742073. doi:10.1177/1550147717742073.

[126] N. C. Luong, Z. Xiong, P. Wang, D. Niyato, Optimal auction for edge computing resource management in mobile blockchain networks: A deep learning approach, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422743.

[127] Z. Xiong, S. Feng, D. Niyato, P. Wang, Z. Han, Optimal pricing-based edge computing resource management in mobile blockchain, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422517.

[128] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, A. Alamri, A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography, IEEE Access 5 (2017) 22313–22328. doi:10.1109/ACCESS.2017.2757844.

[129] Q. Huang, Y. Yang, L. Wang, Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things, IEEE Access 5 (2017) 12941–12950. doi:10.1109/ACCESS.2017.2727054.

[130] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, X. Yao, Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things, IEEE Internet of Things Journal (IoT-J) 4 (2017) 1143–1155. doi:10.1109/JIOT.2017.2659783.

[131] S. He, B. Cheng, H. Wang, X. Xiao, Y. Cao, J. Chen, Data security storage model for fog computing in large-scale IoT application, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2018. doi:10.1109/INFCOMW.2018.8406927.

[132] J. Ni, K. Zhang, Y. Yu, X. Lin, X. S. Shen, Providing task allocation and secure deduplication for mobile crowdsensing via fog computing, IEEE Transactions on Dependable and Secure Computing (TDSC) (2018). doi:10.1109/TDSC.2018.2791432.

[133] R. Casado-Vara, F. de la Prieta, J. Prieto, J. M. Corchado, Blockchain framework for IoT data quality via edge computing, in: Proceedings of the 1st Workshop on Blockchain-enabled Networked Sensor Systems (BlockSys), ACM, 2018, pp. 19–24. doi:10.1145/3282278.3282282.

[134] W. Abdul, Z. Ali, S. Ghouzali, B. Alfawaz, G. Muhammad, M. S. Hossain, Biometric security through visual encryption for fog edge computing, IEEE Access 5 (2017) 5531–5538. doi:10.1109/ACCESS.2017.2693438.

[135] J. Cui, L. Wei, J. Zhang, Y. Xu, H. Zhong, An Efficient Message-Authentication Scheme Based on Edge Computing for Vehicular Ad Hoc Networks, IEEE Transactions on Intelligent Transportation Systems (TITS) (2018). doi:10.1109/TITS.2018.2827460.

[136] B. Mukherjee, R. L. Neupane, P. Calyam, End-to-End IoT Security Middleware for Cloud-Fog Communication, in: IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 151–156. doi:10.1109/CSCloud.2017.62.

[137] A. A. Diro, N. Chilamkurti, N. Kumar, Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing, Mobile Networks and Applications 22 (2017) 848–858. doi:10.1007/s11036-017-0851-8.

[138] Q. Wang, D. Chen, N. Zhang, Z. Ding, Z. Qin, PCP: A Privacy-Preserving Content-Based Publish-Subscribe Scheme With Differential Privacy in Fog Computing, IEEE Access 5 (2017) 17962–17974. doi:10.1109/ACCESS.2017.2748956.

[139] D. He, Y. Qiao, S. Chan, N. Guizani, Flight Security and Safety of Drones in Airborne Fog Computing Systems, IEEE Communications Magazine (COMMAG) 56 (2018) 66–71. doi:10.1109/MCOM.2018.1700916.

[140] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, Z. Ye, FOCUS: A fog computing-based security system for the Internet of Things, in: IEEE Annual Consumer Communications & Networking Conference (CCNC), 2018, pp. 1–5. doi:10.1109/CCNC.2018.8319238.

[141] F. Y. Okay, S. Ozdemir, A secure data aggregation protocol for fog computing based smart grids, in: IEEE International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG), 2018, pp. 1–6. doi:10.1109/CPE.2018.8372598.

[142] X. Wang, L. Wang, Y. Li, K. Gai, Privacy-Aware Efficient Fine-Grained Data Access Control in Internet of Medical Things based Fog Computing, IEEE Access (2018). doi:10.1109/ACCESS.2018.2856896.

[143] G. C. Sih, E. A. Lee, A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures, IEEE transactions on Parallel and Distributed systems (TPDS) 4 (1993) 175–187. doi:10.1109/71.207593.

[144] H. Topcuoglu, S. Hariri, M.-y. Wu, Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE transactions on Parallel and Distributed systems (TPDS) 13 (2002) 260–274. doi:10.1109/71.993206.

[145] J. Li, S. Su, X. Cheng, Q. Huang, Z. Zhang, Cost-conscious scheduling for large graph processing in the cloud, in: IEEE International Conference on High Performance Computing and Communications (HPCC), 2011, pp. 808–813. doi:10.1109/HPCC.2011.147.

[146] R. L. Kadri, F. F. Boctor, An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case, European Journal of Operational Research (EJOR) 265 (2018) 454–462. doi:10.1016/j.ejor.2017.07.027.

[147] M. Nouiri, A. Bekrar, A. Jemai, S. Niar, A. C. Ammari, An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem, Journal of Intelligent Manufacturing 29 (2018) 603–615. doi:10.1007/s10845-015-1039-3.

[148] S. Mirjalili, Ant Colony Optimisation, in: Evolutionary Algorithms and Neural Networks, Springer, 2019, pp. 33–42. doi:10.1007/978-3-319-93025-1_3.

[149] T. Zhu, S. Chen, W. Zhu, Y. Wang, Optimization of sound absorption property for polyurethane foam using adaptive simulated annealing algorithm, Journal of Applied Polymer Science 135 (2018) 46426.

doi:10.1002/app.46426.
[150] B. Liu, L. Wang, Y.-H. Jin, An effective PSO-based memetic algorithm for flow shop scheduling, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 37 (2007) 18–27. doi:10.1109/TSMCB.2006.883272.
[151] C. Adjiman, I. Androulakis, C. Floudas, Global optimization of MINLP problems in process synthesis and design, Computers & Chemical Engineering 21 (1997) S445–S450. doi:10.1016/S0098-1354(97)87542-4.
[152] C. S. Pawar, R. B. Wagh, Priority based dynamic resource allocation in cloud computing with modified waiting queue, in: IEEE International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, pp. 311–316. doi:10.1109/ISSP.2013.6526925.
[153] A. Marphatia, A. Muhnot, T. Sachdeva, E. Shukla, Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing, IOSR Journal of Computer Engineering (IOSR-JCE) (2013) 2278–0661. doi:10.9790/0661-1050105.
[154] A. V. Lakra, D. K. Yadav, Multi-objective tasks scheduling algorithm for cloud computing throughput optimization, Procedia Computer Science 48 (2015) 107–113. doi:10.1016/j.procs.2015.04.158.
[155] J. Xie, D. Guo, Z. Hu, T. Qu, P. Lv, Control plane of software defined networks: A survey, Computer communications (ComCom) 67 (2015) 1–10. doi:10.1016/j.comcom.2015.06.004.
[156] D. Qi, S. Shen, G. Wang, Towards an efficient VNF placement in network function virtualization, Computer Communications (ComCom) 138 (2019) 81–89. doi:10.1016/j.comcom.2019.03.005.
[157] N. H. Motlagh, T. Taleb, O. Arouk, Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives, IEEE Internet of Things Journal (IoT-J) 3 (2016) 899–922. doi:10.1109/JIOT.2016.2612119.
[158] A. Acar, H. Aksu, A. S. Uluagac, M. Conti, A survey on homomorphic encryption schemes: Theory and implementation, ACM Computing Surveys (CSUR) 51 (2018) 79. doi:10.1145/3214303.
[159] B. Nour, A. Ksentini, N. Herbaut, P. A. Frangoudis, H. Moungla, A Blockchain-Based Network Slice Broker for 5G Services, IEEE Networking Letters (2019). doi:10.1109/LNET.2019.2915117.
[160] M. Conti, S. Kumar, C. Lal, S. Ruj, A survey on security and privacy issues of bitcoin, IEEE Communications Surveys & Tutorials (COMST) (2018). doi:10.1109/COMST.2018.2842460.
[161] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, K. Zheng, Survey on blockchain for Internet of Things, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.01.006.
[162] N. Islam, Y. Faheem, I. U. Din, M. Talha, M. Guizani, M. Khalil, A blockchain-based fog computing framework for activity recognition as an application to e-Healthcare services, Future Generation Computer Systems (FGCS) 100 (2019) 569–578. doi:10.1016/j.future.2019.05.059.
[163] Y. Jiao, P. Wang, D. Niyato, Z. Xiong, Social welfare maximization auction in edge computing resource allocation for mobile blockchain, in: IEEE International Conference on Communications (ICC), 2018, pp. 1–6. doi:10.1109/ICC.2018.8422632.
[164] P. Dütting, Z. Feng, H. Narasimhan, D. C. Parkes, Optimal auctions through deep learning, arXiv preprint arXiv:1706.03459 (2017).
[165] K. Suankaewmanee, D. T. Hoang, D. Niyato, S. Sawadsitang, P. Wang, Z. Han, Performance analysis and application of mobile blockchain, in: IEEE International Conference on Computing, Networking and Communications (ICNC), 2018, pp. 642–646. doi:10.1109/ICCNC.2018.8390265.
[166] B. Wickremasinghe, R. N. Calheiros, R. Buyya, Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications, in: IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, pp. 446–452. doi:10.1109/AINA.2010.32.
[167] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software: Practice and experience (SPE) 41 (2011) 23–50. doi:10.1002/spe.995.
[168] W. Chen, E. Deelman, Workflowsim: A toolkit for simulating scientific workflows in distributed environments, in: IEEE International Conference on E-science (e-science), 2012, pp. 1–8. doi:10.1109/eScience.2012.6404430.
[169] P.-O. Ostberg, H. Groenda, S. Wesner, J. Byrne, D. S. Nikolopoulos, C. Sheridan, J. Krzywda, A. Ali-Eldin, J. Tordsson, E. Elmroth, et al., The CACTOS vision of context-aware cloud topology optimization and simulation, in: IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2014, pp. 26–31. doi:10.1109/CloudCom.2014.62.
[170] T. T. Sá, R. N. Calheiros, D. G. Gomes, CloudReports: An extensible simulation tool for energy-aware cloud computing environments, in: cloud computing, Springer, 2014, pp. 127–142. doi:10.1007/978-3-319-10530-7_6.
[171] B. Louis, K. Mitra, S. Saguna, C. Åhlund, Cloudsimdisk: Energy-aware storage simulation in cloudsim, in: IEEE/ACM International Conference on Utility and Cloud Computing (UCC), 2015, pp. 11–15. doi:10.1109/UCC.2015.15.
[172] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio, M. M. Freire, CloudSim plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness, in: IFIP/IEEE International Symposium on Integrated Network Management (IM), 2017, pp. 400–406. doi:10.23919/INM.2017.7987304.
[173] D. Kliazovich, P. Bouvry, S. U. Khan, GreenCloud: a packet-level simulator of energy-aware cloud computing data centers, The Journal of Supercomputing (TJSC) 62 (2012) 1263–1283. doi:10.1007/s11227-010-0504-1.
[174] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, I. M. Llorente, iCanCloud: A flexible and scalable cloud infrastructure simulator, Journal of Grid Computing (JGC) 10 (2012) 185–209. doi:10.1007/s10723-012-9208-5.
[175] C. Sonmez, A. Ozgovde, C. Ersoy, EdgeCloudSim: An environment for performance evaluation of Edge Computing systems, in: IEEE International Conference on Fog and Mobile Edge Computing (FMEC), 2017, pp. 39–44. doi:10.1002/ett.3493.
[176] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, R. Buyya, iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments, Software: Practice and experience (SPE) 47 (2017) 1275–1296. doi:10.1002/spe.2509.
[177] M. M. Lopes, W. A. Higashino, M. A. Capretz, L. F. Bittencourt, Myifogsim: A simulator for virtual machine migration in fog computing, in: International Conference on Utility and Cloud Computing (UCC), ACM, 2017, pp. 47–52. doi:10.1145/3147234.3148101.
[178] C. G. Isaac Lera, YAFS (Yet Another Fog Simulator), https://pypi.org/project/yafs/, 2018.
[179] M. Etemad, M. Aazam, M. St-Hilaire, Using DEVS for modeling and simulating a Fog Computing environment, in: IEEE International Conference on Computing, Networking and Communications (ICNC), 2017, pp. 849–854. doi:10.1109/ICCNC.2017.7876242.
[180] NetSim v12, https://www.tetcos.com/, 2019.
[181] Emufog, https://github.com/emufog/emufog, 2017.
[182] Fogbed, https://github.com/fogbed/fogbed, 2018.
[183] J. Wilkes, More Google cluster data, Google research blog, Nov (2011).
[184] S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.10.014.
[185] R. Wang, Y. Cao, A. Noor, T. A. Alamoudi, R. Nour, Agent-enabled task offloading in UAV-aided mobile edge computing, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.10.021.
[186] N. Piovesan, A. F. Gambin, M. Miozzo, M. Rossi, P. Dini, Energy sustainable paradigms and methods for future mobile networks: A survey, Computer Communications (ComCom) 119 (2018) 101–117. doi:10.1016/j.comcom.2018.01.005.
[187] M. Anbarasan, B. Muthu, C. Sivaparthipan, R. Sundarasekar, S. Kadry, S. Krishnamoorthy, A. A. Dasel, et al., Detection of flood disaster system based on IoT, big data and convolutional deep neural network, Computer Communications (ComCom) (2019). doi:10.1016/j.comcom.2019.11.022.
[188] S. Zafar, S. Jangsher, O. Bouachir, M. Aloqaily, J. B. Othman, QoS enhancement with deep learning-based interference prediction in mobile IoT, Computer Communications (ComCom) 148 (2019) 86–97. doi:10.1016/j.comcom.2019.09.010.