

Received 23 May 2022, accepted 12 June 2022, date of publication 16 June 2022, date of current version 5 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183634

Edge Computing Technology Enablers: A Systematic Lecture Study

SALMANE DOUCH^{1,3}, (Graduate Student Member, IEEE),
MOHAMED RIDUAN ABID^{2,3}, (Member, IEEE), **KHALID ZINE-DINE⁴**, **DRISS BOUZIDI¹**,
AND DRISS BENHADDOU⁵, (Senior Member, IEEE)

¹National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University in Rabat, Rabat 30050, Morocco

²TSYS School of Computer Science, Columbus State University, Columbus, GA 31907, USA

³School of Science and Engineering, Al Akhawayn University, Ifran 53000, Morocco

⁴Faculty of Sciences (FSR), Mohammed V University in Rabat, Rabat 30050, Morocco

⁵Department of Engineering Technology, University of Houston, Houston, TX 77004, USA

Corresponding author: Salmane Douch (sa.douch@aui.ma)

This work was supported by the National Academy of Sciences (NAS)/United States AID (USAID) under the PEER Cycle 5 Project Grant 5-398, entitled “Towards Smart Microgrid: Renewable Energy Integration Into Smart Buildings.”

ABSTRACT With the increasing stringent QoS constraints (e.g., latency, bandwidth, jitter) imposed by novel applications (e.g., e-Health, autonomous vehicles, smart cities, etc.), as well as the rapidly increasing number of connected IoT (Internet of Things) devices, the core network is becoming increasingly congested. To cope with those constraints, Edge Computing (EC) is emerging as an innovative computing paradigm that leverages Cloud computing and brings it closer to the customer. “EC” refers to transferring computing power and intelligence from the central Cloud to the network’s Edge. With that, EC promotes the idea of processing and caching data at the Edge, thus reducing network congestion and latency. This paper presents a detailed, thorough, and well-structured assessment of Edge Computing and its enabling technologies. Initially, we start by defining EC from the ground up, outlining its architectures and evolution from Cloudlets to Multi-Access Edge Computing. Next, we survey recent studies on the main cornerstones of an EC system, including resource management, computation offloading, data management, network management, etc. Besides, we emphasized EC technology enablers, starting with Edge Intelligence, the branch of Artificial Intelligence (AI) that integrates AI models at resource-constrained edge nodes with significant heterogeneity and mobility. Then, moving on to 5G and its empowering technologies, we explored how EC and 5G complement each other. After that, we studied virtualization and containerization as promising hosting runtime for edge applications. Further to that, we delineated a variety of EC use-case scenarios, e.g., smart cities, e-Health, military applications, etc. Finally, we concluded our survey by highlighting the role of EC integration with future concerns regarding green energy and standardization.

INDEX TERMS Edge computing, cloud computing, fog computing, multi-access edge computing, edge intelligence, 5G, containerization.

I. INTRODUCTION

In recent years, the number of connected devices has grown tremendously, causing congestion issues that push to consider handling more data at the network’s Edge. According to Gartner [1], by 2025, 75% percent of data generated by enterprises will be processed at the Edge rather than the centralized Cloud. In addition to those network impracticability constraints, EC promises to provide an excellent service

(latency, throughput) that will encourage the evolution of this revolutionary paradigm. As IBM pointed out [2], moving the computing workload to the Edge will reduce data circulation time from 20 ms to 10 ms.

The added value of EC offers tremendous market opportunities to multiple market participants, counting Cloud providers, ISP (Internet service providers), and numerous intermediate hardware and software companies. Based on research done by Grand View Research [3], the EC market is forecast to extend from 3.4\$ US billions dollar in 2020 to 43.4\$ US billions dollar in 2027, with a growth of 37.4 %

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

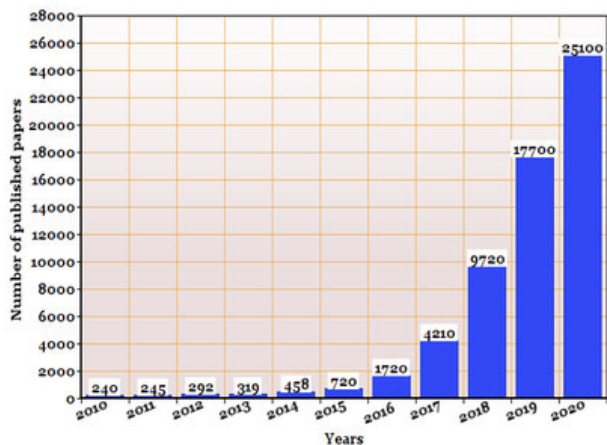


FIGURE 1. Edge computing number of published papers in Google Scholar [4], between 2010 to 2020.

percent each year. From a different perspective, EC has attracted much attention from academia in the last ten years, as evidenced by the exponential increase in published papers on topics ranging from EC architectures to deployment challenges, orchestration platforms, EC use-cases, and related technologies. Figure 1 depicts the evolution in terms of the number of EC papers published in Google Scholar [4].

In this effort, we present a survey on edge computing, beginning with an explanation of the novel computing concept and how EC can meet the growing demand for computing and memory resources with low latency communications, as well as how EC can solve the rising privacy problems associated with processing data at the cloud level. Alternatively, the collaboration of numerous heterogeneous devices at various levels of the network edge is what defines edge computing. EC enables those resources to be efficiently managed, scaled, and secured, allowing them to act as performing hosts for workloads received from end devices.

Nevertheless, EC is associated with several innovative technologies, notably the Internet of Things (IoT). Along with the fifth-generation networks (5g), EC is a vital solution for enabling the polarization of connected objects. Moreover, artificial intelligence (AI) and machine learning (ML) technologies are becoming more prevalent in novel applications. Consequently, there is a growing need for computing resources. Not only will EC address this requirement, but it will also adapt AI models to the network edge environment, promoting the idea of “Edge Intelligence.”

A. SURVEY ORGANISATION

The following is a summary of the rest of the paper, section II presents a definition of EC and a lecture study of the different related EC surveys, plus it underscores our unique contribution and novelty. Section III gives a brief history of the evolution of EC from Cloudlets to Multi-access Edge Computing (MEC) while also highlighting the differences in architectural design of each sub-EC concept. Further, Section IV discussed the recent advancements made in EC main pillars,

counting resource management, computation offloading, data management, network management, security and privacy, and EC pricing & billing. Next, in section V, we examined the three major enabling technologies of EC, which are Edge Intelligence, 5G, and Containerization, and in the process, we demonstrated how these technologies are crucial for EC’s success. Furthermore, in section VI, we presented the various scenarios and use cases in which EC is proving to be greatly useful, ranging from e-health to smart cities, and from entertainment to military applications. Succeeding that, in section VII, we discussed the future concerns facing Edge Computing, such as standardization and efficient green energy integration to EC. Lastly, in section VIII, we ended our work with a conclusion paragraph.

Fig. 2 shows the survey structure and a map for assisting the reader. Table 1 offers a helpful tool for defining the used acronyms and abbreviations in the survey.

II. RELATED SURVEYS

A. EC: DEFINITION

There is no standard definition of EC, but many researchers view EC as an abstracted computing paradigm that aims to move cloud computing and storage capabilities to the network edge near where the end-users reside. For two main reasons. The first reason is to meet the current need for quality of service (latency & bandwidth) imposed by the latest applications, and the second reason is to address the problem of core network up-growing pressure. Additionally, there are some notable definitions of edge computing, one of the first papers to use the terminology “Edge Computing” is [5], within, the authors do define the concept as “*the enabling technology that allows the computation to be performed at the edge of the network, on downstream data on behalf of cloud services, and upstream data on behalf of IoT services*”.

In order to completely comprehend the EC concept and its functions, the following sections tackle the main two questions in EC: where is the Edge located? And what exactly is the purpose of edge computing?

1) WHERE IS THE EDGE LOCATED?

As a term, the word “Edge” signifies the extreme part of any given network. In the case of a telecommunications network, it refers to the RAN (Radio Access networks) part. While in the case of the data network (the Internet), the Edge or, more precisely, the Edge Device (ED) is any extreme end-users or IoT device (mobile phones, cars, smartwatches, etc.) [6].

However, In EC, the devices responsible for executing computation tasks are referred to as Edge Servers (ESs) or Edge nodes (ENs). Those computing devices can exist in one hope or a few more from the edge devices. Nonetheless, processing data at the Edge means typically handling it before it crosses any WAN (Wide Area Networks), knowing that passing through any WAN denotes a significant data transfer delay. Supplementary, the nature of an Edge Server varies

TABLE 1. Acronym table.

Acronym	Definition	Acronym	Definition
5G	Fifth-Generation Wireless Technology	ISP	Internet Service Provider
4G	Fourth-Generation Wireless Technology	IoT	Internet of Things
AI	Artificial Intelligence	LORA	Long Range
AR	Augmented Reality	LSTM	Long Short-Term Memory
ASIC	Application Specific Integrated Circuit	LTE	Long Term Evolution
BS	Base Station	MBB	Mobile Broadband
BBU	Baseband unit	MCC	Mobile Cloud Computing
CAPEX	Capital Expenditure	MEC	Multi-Access/Mobile Edge Computing
CC	Cloud Computing	MIMO	Multiple Input Multiple Output
CDN	Content Delivery Networks	ML	Machine Learning
CP	Cloud Provider	MO	Mobile Operator
CPU	Central Processing Unit	MTC	Machine Type of communication
CNN	Convolutional Neural Networks	NFV	Network Function Virtualisation
CSI	Channel State Information	NIST	National Institute of Standards and Technology
C-RAN	Centralised Radio Access Networks	NOMA	Non-Orthogonal Multiple Access
DAG	Direct Acyclic Graph	NN	Neural Networks
DL	Deep Learning	OEC	Orbital Edge Computing
DNN	Deep Neural Networks	OFDMA	Orthogonal Frequency Division Multiple Access
DRL	Deep Reinforcement Learning	OMA	Orthogonal Multiple Access
DRX	Discontinuous Reception	OPEX	Operational Expenditure
EC	Edge Computing	QoE	Quality of Experience
ED	Edge Devices	QoS	Quality of Service
EI	Edge Intelligence	RAN	Radio Access Networks
EIP	Edge Infrastructure Provider	RL	Reinforcement Learning
EN	Edge Node	RNN	Recurrent Neural Networks
ES	Edge Server	RRU	Remote Radio Unit
ESP	Edge Service Providers	RSU	Road Side Unit
ETSI	European Telecommunication Standardization Institute	SDN	Software Defined Networks
EU	Edge User	SM	Smart Grid
FaaS	Function as a Service	TDMA	Time Division Multiple Access
FC	Fog Computing	TPU	Tensorflow Processing Unit
FFNN	Feed Forward Neural Networks	UAV	Unmanned Aerial Vehicle
FL	Federated Learning	uRLLC	Ultra Reliable Low Latency Communication
FPGA	Field Programmable	VEC	Vehicular Edge Computing
F-RAN	Fog Radio Access Networks	VFC	Vehicular Fog Computing
GPU	Graphic Processing Unit	VM	Virtual Machine
GRU	Gated Recurrent Unit	WIFI	Wireless Fidelity
ICT	Information Communication Technologies	WPT	Wireless Power Transfer
IP	Internet Protocol	WSN	Wireless Sensor Networks

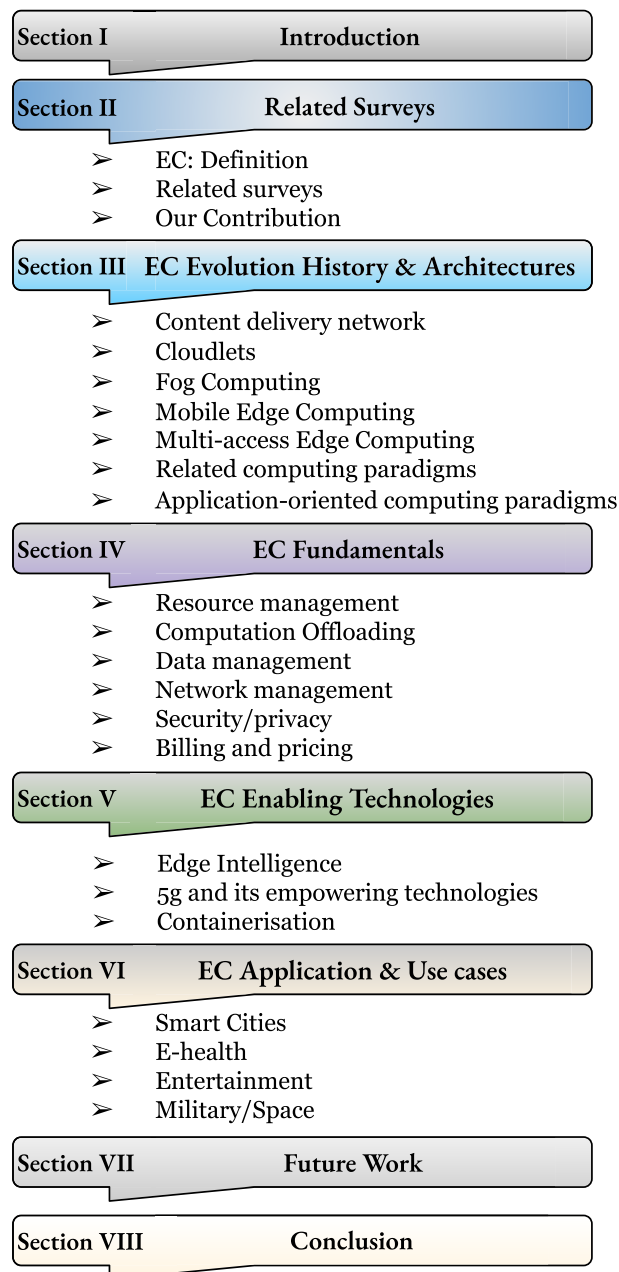


FIGURE 2. The survey structure.

depending on the architecture and context in which it was deployed.

2) WHY EDGE COMPUTING?

There are numerous benefits and drawbacks that drove the need for a new computing paradigm known as Edge Computing, which we divide into two categories: QoS and necessity.

- Quality of Service (QoS) is one of the most important characteristics of novel applications, and it consists of two elements: low latency and high bandwidth. The first element allows numerous novel applications (e.g., autonomous vehicles) to access cloud services with the lowest response time. For the second element, the data

is transferred in shorter paths between the Edge and the end-users, allowing for a higher bandwidth exchange between EC servers and end-users.

Shortly, many industries, homes, and hospitals will strive to own those performance requirements, and with EC, they will be able to effectively receive them while edge computing suppliers handle edge servers deployment and management.

- Necessity, due to the rapidly rising number of IoT devices (tens of billions) and the limited bandwidth, the more computing is performed locally, the better it is for preserving the network capacity, thus the necessity of Edge Computing. Further, another subject that does raise much concern today is privacy. Many users and companies are not self-insured about sending their data to the far Cloud. Therefore, EC, with its ability to keep the data close to where the users requested it, could be the perfect solution for this issue. EC has also been found in [7] to be more environmentally friendly than the Cloud. The video analysis experiment showed that computing at the Edge would reduce CO₂ injection by 50% compared to the Cloud.

B. RELATED SURVEYS

In the past few years, several EC surveys have been proposed. Table II outlines the most important ones of these surveys, as well as the taxonomy of the topics they covered, which includes the EC Concept and History, EC Architecture, EC Fundamentals, Enabling Technologies, Applications, and EC Challenges & Future Concerns.

Further, based on the published time and the main focus of the surveys, we divided the Edge Computing literature surveys into three groups.

Studies from 2012 to 2016, such as [39] and [38] are included in the first group; these reviews provided explanations of the EC concept as well as its implications on issues counting latency, bandwidth, privacy, etc.

Concerning the second group of surveys (2016-2020), the surveys often focused on explaining and comparing different EC architectures (MEC, FOG, Cloudlets, etc.). Moreover, many of them had fully or partially addressed some of EC's main pillars, including computation offloading, managing resources, managing data, and protecting confidential information [34], [37].

Furthermore, with the extensive explorations of the EC concept and its challenges, the number of publications also started to rise dramatically, as Fig. 1 demonstrates. Along with that, the road map of our targeted subject started to expand, and its branches grew tremendously, to a point where EC began to converge and touch other related technologies (e.g, IoT, 5G, EI, etc.). As a result, most surveys in the third group tend to focus on a single topic or a subbranch within EC, such as the recent resource placement survey in [12], resource scheduling in [13], or the work in [9] that cover security and privacy issues in EC.

TABLE 2. Related surveys table.

EC Survey	Year	Definition History	EC Architectures				EC Fundamentals						Enabling Technologies			Applications	Challenges & Future Directions	Ref
			Cloudlets	Fog	MEC	App Oriented Architecture	Resource Management	Computation Offloading	Data Management	Network Management	Security Privacy	Pricing Billing	Edge Intelligence	5G	Containerization			
Babar et al	2021	✓	✓	×	×	×	✓	✓	×	✓	✓	×	×	×	×	✓	✓	[8]
Ranaweera et al	2021	✓	✓	✓	✓	×	✓	✓	×	✓	✓	×	×	✓	×	×	✓	[9]
Sharghivand et al	2021	✓	✓	✓	✓	×	✓	×	×	×	✓	✓	×	×	×	×	✓	[10]
Xu et al	2021	✓	✓	×	×	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	✓	[11]
Sonkoly et al	2021	✓	✓	✓	✓	✓	✓	✓	×	×	✓	×	×	×	×	×	✓	[12]
Luo2021 et al	2021	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓	✓	[13]
Khan et al	2020	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	[14]
Cao et al	2020	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×	×	✓	×	×	×	[15]
Hamdan et al	2020	✓	✓	✓	✓	×	✓	×	✓	✓	✓	×	×	×	×	✓	✓	[16]
Vhora et al	2020	✓	✓	✓	✓	×	×	✓	×	✓	×	×	×	×	×	✓	×	[17]
Salaht et al	2020	✓	✓	✓	✓	×	✓	×	✓	×	×	×	×	×	×	×	✓	[18]
Rafique et al	2020	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	[19]
Martinez et al	2020	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	[20]
Filali et al	2020	✓	×	×	✓	×	✓	✓	✓	✓	×	×	×	✓	×	×	×	[21]
Pham et al	2020	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓	✓	[22]
Habibi et al	2020	✓	✓	✓	✓	✓	✓	×	×	✓	✓	×	×	×	×	×	✓	[23]
Pang et al	2020	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×	×	✓	×	✓	✓	[24]
Narayanan et al	2020	✓	×	✓	✓	✓	✓	×	✓	✓	✓	×	✓	×	✓	×	✓	[25]
Qadir et al	2020	✓	✓	×	✓	✓	✓	✓	✓	×	✓	×	×	×	×	✓	✓	[26]
Mehrabi et al	2019	✓	×	✓	✓	×	✓	✓	✓	✓	✓	×	×	×	×	×	✓	[27]
Yousefpour et al	2019	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×	×	×	×	✓	✓	[6]
Puliafito et al	2019	✓	✓	✓	✓	×	×	×	✓	×	✓	×	×	×	×	✓	✓	[28]
Hong et al	2019	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×	×	×	✓	×	×	[29]
Liu et al	2019	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	×	✓	×	×	×	✓	[30]
Abbas et al	2017	✓	✓	✓	✓	×	×	✓	✓	✓	✓	✓	×	×	×	✓	✓	[31]
Baktir et al	2017	✓	✓	✓	✓	×	✓	✓	×	✓	×	×	×	✓	×	✓	×	[32]
Taleb et al	2017	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	✓	[33]
Mouradian et al	2017	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	×	×	✓	✓	[34]
Mao et al	2017	✓	✓	×	✓	×	✓	✓	×	✓	✓	×	×	✓	×	✓	✓	[35]
Mach et al	2017	✓	✓	✓	✓	×	✓	✓	×	✓	×	×	×	×	×	✓	✓	[36]
Wang et al	2017	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	×	×	×	✓	✓	[37]
Ahmed et al	2016	✓	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	✓	✓	[38]
Shaukat et al	2015	✓	✓	×	×	×	×	✓	✓	✓	✓	✓	×	×	×	×	×	[39]

1) OUR CONTRIBUTION

In our work, we drew inspiration from the excellent EC systematic mapping study built-in [40], recognizing the need to fill the gap and produce a systematic lecture study on the subject. We observe a lack of works covering EC technology enablers since the few works covering EC enabling technologies focused only on one enabling technology of EC. Based on our modest research, this is the first work to provide coverage of all EC primary technology enablers.

In order to distinguish ourselves from other similar surveys, we emphasize our contribution and novelty in the items below.

- 1) We present a complete and comprehensive Edge Computing survey, covering most topics related to Edge Computing.
- 2) We offer an essential and precise road map of EC, illustrating the concept's main branches and subbranches.
- 3) We provide an in-depth and recent lecture study about the advancements made in the various EC sub-domains.

- 4) This work is the first detailed overview of EC enabling technologies, showing how Edge Intelligence, 5G, and Containerization empower this revolutionary computing paradigm.

III. HISTORY AND ARCHITECTURES: THE EVOLUTION OF EDGE COMPUTING

The evolution of this computing paradigm has been carried through several stages since 1997. As shown in Fig. 3, each step influenced the concept directly or indirectly and changed the way EC is conceived.

A. CONTENT DELIVERY NETWORKS (CDNs)

Content delivery networks (CDNs) were developed at MIT by a group of researchers who were trying to solve the flash crowd problem [41], where an individual server is unable to serve a large number of requests. The MIT researchers recommend replicating the content on numerous intelligently spread edge servers. Hence, CDNs were the first technology capable of delivering memory resources at the Edge of the network [42].

In the last decade, CDNs usage has become increasingly popular among website owners, who cache their files (HTML, Images, Java-Scripts) at a CDN provider to offer a better web experience to their users. Fig. 4 describes the process of requesting content from a CDN provider.

B. CLOUDLETS

To get into the history of Edge Computing, one must first understand its parents computing paradigms, pervasive computing, and cloud computing [43]. Pervasive or ubiquitous computing inspires the idea of making computations accessible everywhere, where clients can access capable computers from any place and at any timestamp. Motivated by the ubiquitous concept, the Cloud Computing (CC) framework developed as a modern worldview that conveys computing and memory resources on a pay-as-you-consume premise. It makes computing assets a service instead of a product [44]. In 2006, CC got prevalent with AMAZON's "Elastic Compute." CC solidifies numerous heterogeneous servers for providing infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS), all in an adaptable and versatile design.

However, on the other hand, CC lacked one critical performance indicator: latency. Because centralized data centers were so far from the end-users, they could not guarantee short communication delays. In response to this urge, in 2009, Microsoft proposed Cloudlets [45], a concept in which Cloud users can request computing resources from micro-datacenters (from one to forty servers) called Cloudlets, which are widespread small data centers with virtualized infrastructure located closer to the end-users, thus offering low latency connection between them and Cloud users. In that process, with Cloudlets computing, users can request cached content like in CDNs, offload computation tasks to cloudlets, and, most importantly, get a response in a few milliseconds

(\ll Cloud WAN) [8]. Over the next few years, Cloudlets will be renamed afterward to edge servers as a computing component of EC.

C. FOG COMPUTING

The origin of fog computing is traced back to Cisco vision, a company that served as a bridge between the end-users and the Cloud. It saw an opportunity in 2012 to introduce a new computing paradigm known as FOG Computing (FC) [46]. FC aims to provide a continuous computing capability between IoT devices and the Cloud. FC is established on collaboration between multiple heterogeneous devices known as fog nodes [47]. Those devices may exist at different levels of the network (e.g., switches, commodities, servers, micro-data centers, etc.). Fig. 5 shows the fog computing paradigm's architecture and its computing elements. The fog computing paradigm differs from Cloudlets because it does not consider fog nodes as isolated devices but as part of a pool of computing resources that can be extended to the Cloud. Among Fog Computing's keywords is "Orchestration" [48], which is the essential mechanism responsible for automating and managing fog resources across multiple network levels.

Besides that, FC does require serious cooperation between different network and cloud provider entities, this lack of collaboration was the reason why two years later, Cisco canceled the fog computing project [49], as the workload was too much for Cisco to handle on its own. Additionally, as described in [50], there are two types of fog computing architectures, the hierarchical architecture, where nodes from different network layers can collaborate to perform tasks together, and the flat architecture, where nodes from the same layer join forces to perform fog computing.

D. MOBILE EDGE COMPUTING

From an alternative viewpoint, integrating rich computation capabilities into mobile devices has always been a significant concern, especially since the emergence of smartphones and their associated sophisticated applications (>2013). Those new requirements led to the creation of Mobile Cloud Computing (MCC), a computing model that extends CC capabilities to mobile applications [51]. In MCC, mobile applications can offload some of their intensive workloads to the Cloud for processing. However, this is no longer sufficient with the new imposed real-time communication restriction of mobile applications, plus the number of mobile devices has increased considerably, causing severe congestion in the core network.

Following those circumstances, in 2014, ETSI (European Telecommunications Standards Institute) and Industry Specification Group (ISG) proposed Mobile Edge Computing (MEC), "a computing paradigm that provides IT and cloud capabilities within the Radio Access Networks (RANs), in close proximity to mobile subscribers" [52]. Computing Nodes in Mobile Edge Computing are called Small Cell Clouds (SCC), which are servers used to enhance small network cells (SCeNBs), counting microcells, picocells, or femtocells. The idea of adding computation to RAN was first

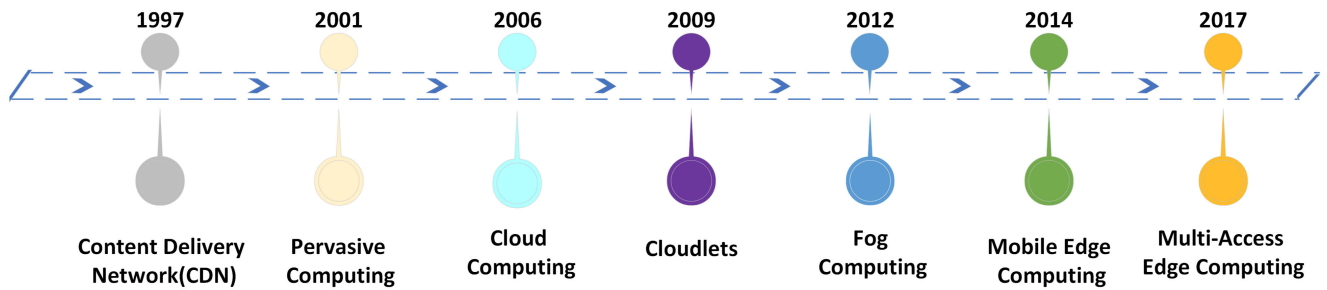


FIGURE 3. Edge computing evolution chronograph.

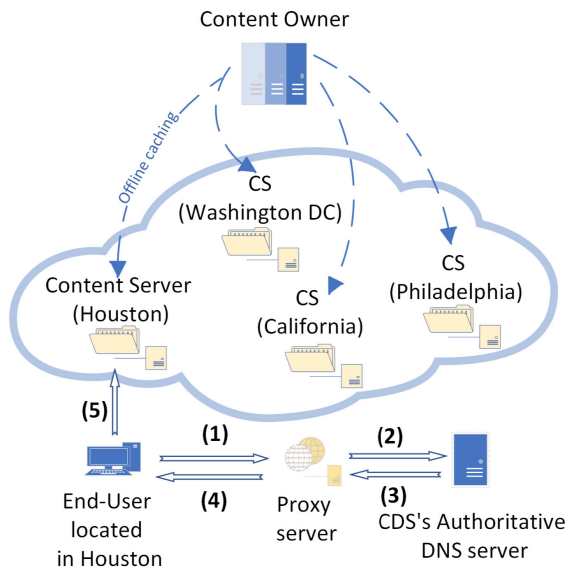


FIGURE 4. The steps of requesting a content in a CDNs.

proposed in 2012 by the European project TROPIC [36]. However, in MEC, those computing components are extended to host different edge applications. In addition, in 2014, Nokia and China Mobile performed a successful MEC test at a car race stadium, where 17000 users were connected to 95 LTE small cells receiving HD video from MEC servers [53].

E. MULTI-ACCESS EDGE COMPUTING

At MEC World Congress in 2016, the European Telecommunications Standards Institute (ETSI) officially changed its MEC name from Mobile Edge Computing to Multi-Access Edge Computing. Considering the enlargement of connected devices that are not mobile devices, ESTI decided to focus more on integrating MEC resources to non-cellular networks (WiFi) and fixed networks (physical cables) [54]. Fig. 6 represents a basic MEC architecture, where the MEC resources are accessed from different types of networks (Cellular, non-cellular and Fixed).

After the expansion, MEC can be deployed at various places in RAN [21], listed in the following:

- Base stations, including mobile base stations, cell towers, central office base stations.
- WiFi access points.

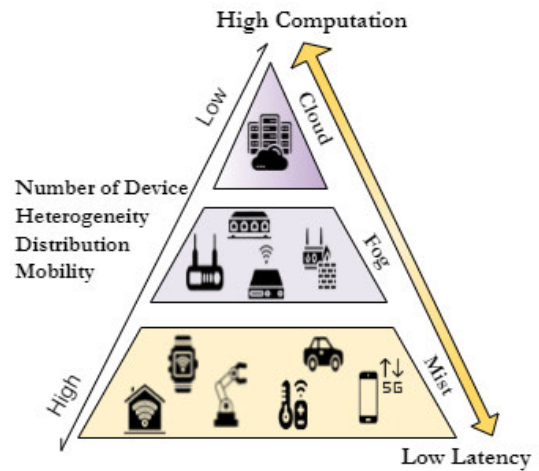


FIGURE 5. The different EC network levels.

- Radio Network Controller (RNC).
- Cable Modem Termination Systems (CMTS) in the case of fixed networks.
- PON OLT (Optical Cable Unit) for fiber, or the access points for other networks such as Zigbee, LoRa (Long Range), private LTE, etc.

F. OTHER HONORABLE RELATED COMPUTING PARADIGMS

Besides Cloudlets, Fog, and MEC, and in the sub-sections below, we discussed other relevant and notable distributed Edge Computing concepts and architectures

1) MIST COMPUTING

Mist computing is a computing paradigm that exploits the participation of multiple extreme edge components (such as Micro-controllers, mobile devices, sensors, etc.) to provide a computing platform that is based on the IoT devices themselves without relying on outsider computing nodes located at the Edge, Fog, or Cloud level [55].

2) DEW COMPUTING

Dew computing is a computation concept that was introduced in 2015 [56]. Dew computing focuses on the formation of a collaborative link between Cloud Computing components and end-personal computing devices. This collaboration

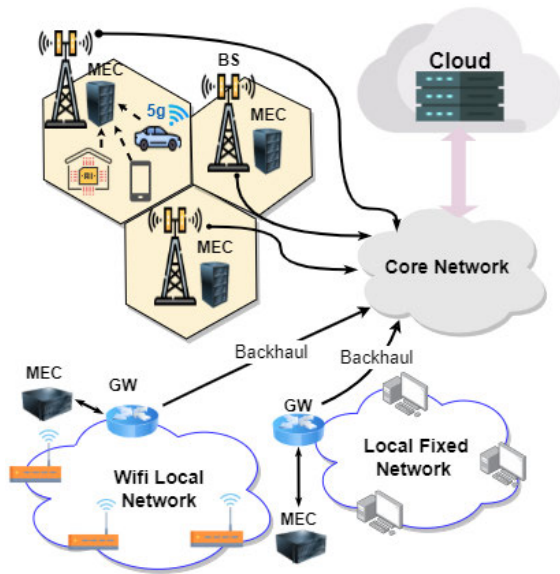


FIGURE 6. 5G-multi-access edge computing architecture.

allows resources to migrate between the two components depending on the network conditions.

3) OSMOTIC COMPUTING

Osmotic computing is a new computing archetype that supports the efficient execution of Internet of Things (IoT) services at the network edge. This paradigm is founded on the need to couple microservices deployed at the Edge with those services running on large skill data centers [57]. If the consolidation of multiple data centers creates CC, Osmotic Computing is characterized by connecting the Cloud, the Fog, and the Edge for the seamless and free microservices movement between them.

G. USE-CASES ORIENTED COMPUTING PARADIGMS

It is widely regarded that edge computing architectures like Cloudlets, MEC, and Fog, can serve a wide range of end-users using various computing resources. Nonetheless, there are other use-cases-oriented architectures in the literature. Those architectures are based on scenarios in which there are specifications on the type of edge nodes and the nature of end-users. For example, the following is a list of EC's application-oriented computing paradigms shown in Fig. 5.

- VEC (Vehicular Edge Computing) [58], in VEC, connected vehicles along with a group of computing units called RSUs (Road Side Units) collaborate to offer EC services primarily for improving the vehicular road system, making it more intelligent and safer.
- OEC (Orbital or Satellite Edge Computing) [59]: is a research area aimed at equipping satellites with computing power. Within, satellites utilize their collaborative coverage and one-hop connection with end-users to deliver EC services.
- "UAV-EC" (Unmanned Ariel Vehicle Edge Computing) [60]: is a concept in which a collection of unmanned

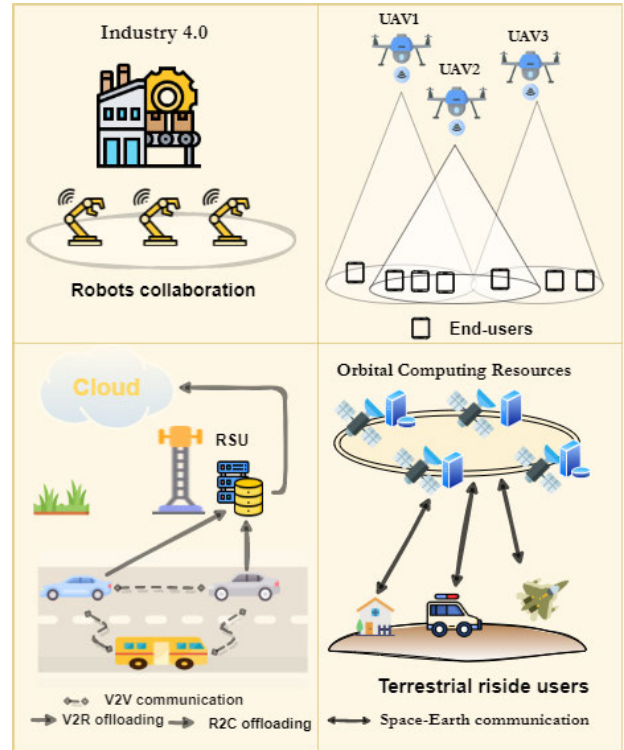


FIGURE 7. The different EC application oriented-architecture.

aerial vehicles (UAVs) swarm over a region to cover customers' needs for computing resources with low latency connections.

- Robotics Edge Computing [61]: is a field that entails the merging of different robotic/industrial resources to enhance production processes and robot-human interactions.

In the following chapters of our survey, if the type of EC architecture is not specified, we will refer to any node in Fog, MEC, or Cloudlets as an edge server or an edge node.

IV. EDGE COMPUTING FUNDAMENTALS

A. RESOURCE MANAGEMENT

The task of resource management is the act of providing the right and comfortable scale of edge resources (CPU, memory, I/O) to any requesting edge application while also optimizing the usage of the exciting pool of resources. In CC, a good resource management strategy gives the cloud provider a flexible and efficient way to manage his IT resources, making it an essential element of any successful cloud business. The management of resources is one of the essential pillars in EC, as it represents the ability that enables the consolidation of multiple dynamic, heterogeneous, and dispersed edge nodes [62].

Resource management can be broken down into many connected phases, as shown in Figure 6, including generating and distributing a pool of resources, monitoring those resources and provisioning them ahead of time, and lastly, allocating those resources to forthcoming demands.

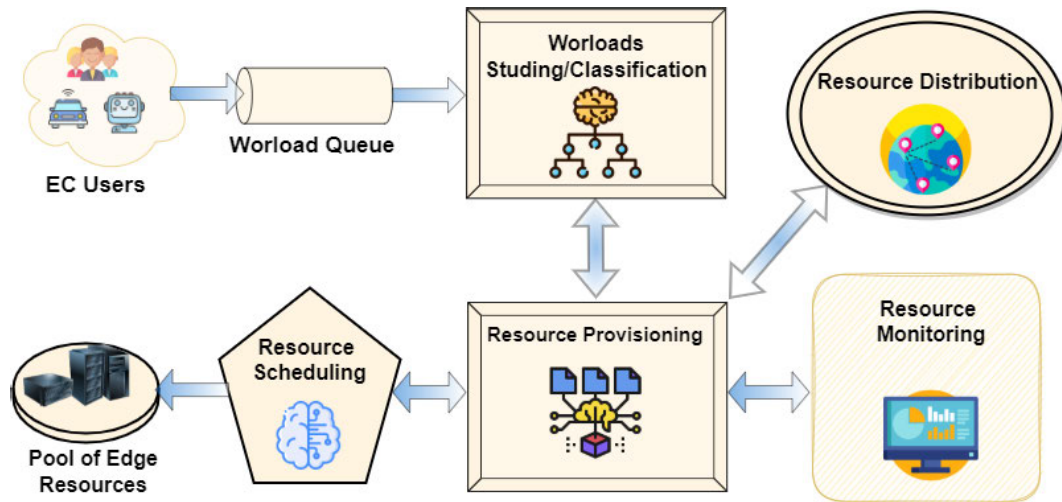


FIGURE 8. Resource management elements in edge computing.

1) RESOURCE ALLOCATION & SCHEDULING

Allocating or scheduling resources entails assigning each upcoming workload to the best and most appropriate edge server (physical or virtual) to host it. Since QoS is a critical differentiator between the Edge and the Cloud, knowing the exact quantity and quality of resources to allocate for a pending request is critical to the success of Edge Computing [13]. In EC, resource scheduling is a complex function because there are numerous factors and conditions to consider. Some of those are discussed further below:

- Cost-driven, in resource scheduling, one of the goals is to reduce computation and bandwidth costs while maintaining the required response time. Numerous studies have been conducted to achieve a reasonable balance between reducing the consumed energy and minimizing the latency, [63] and [64] are some of them.
- Crowd management, whenever a workload is mapped to the closed-edge resources, it must be aware of the load percentage of nearby edge devices. Therefore, overcrowding a specific host can be avoided [65]. Additionally, busy requests are one of the overloading issues that may affect a particular server or a region. The effort [66] provides a collaborative edge-to-edge method to address the issue of busy requests, in which the authors recommend dispersing requests to other surrounding nodes or regions.
- Dynamic demand, applications, and services at the Edge are often characterized by dynamic changes in the quantity and quality of resources they require over time and space. Thus, the scheduler should take these changes into account and recalculate and adjust the allocated resource dynamically in order to avoid any degradation of QoS or under-utilization of resources [67]. The mobile augmented reality applications provide an excellent example of resource demand fluctuation [68].
- Priorities, EC Environments feature concurrency, where many users are fighting over a spot in an edge node.

In this case, the scheduler should be as fair as possible with all users' requests. The study in [69] compares different scheduling algorithms according to different priorities (for instance, the first coming, the type of client, the nature of the task, etc.). As an example, analytical queries follow a type of scheduling selection based on the type of the tasks. Within, the tasks assigner prioritize edge nodes with the most relevant data statistics for receiving the analytical tasks [70].

- Agile learning, based on its past scheduling performance, the resource allocation optimizer can learn to make better decisions in the future, as demonstrated in the platform Deft [71]. Alternatively, predicting upcoming workloads in the space-time continuum provides the edge scheduler with valuable information on which to base his future-aware scheduling decisions. Regression models [72], LSTM [73], and Bayesian learning [74] are among machine learning models that have been used in literature for predicting upcoming workloads.
- Fault-tolerance, because edge nodes can lose power or connectivity at any time, offering backup copies of any scheduled application would improve the overall fault-tolerance of EC services [75]. Another reason for duplicating service instances on several edge servers is to avoid software multi-tenancy architecture [76], in which one server serves numerous users; this might result in a significant reduction in data throughput, hence violating the EC latency requirement.

2) RESOURCE PLACEMENT AND MIGRATION

The placement of resources refers to designing and engineering the optimal distributing strategy of physical and virtual edge servers. Since edge/fog computing is still in its infancy in terms of real-world deployment, many contemporary studies are now attempting to determine what is the best placement strategy for edge nodes, taking into account a variety of criteria (latency, reliability, user preferences),

and considering a variety of scenarios (metropolitan network, vehicular network, etc.).

People and companies in metropolitan areas can utilize EC for their computation and memory needs. In this scenario, edge nodes are dispersed in a similar way to antennas, which means that as the density of a region increases, more edge servers are needed to satisfy demand [77]. Meanwhile, a more precise approach is to consider not only the number of users in a region but also the degree to which the end-users are interested in using latency-aware applications [78]. Moreover, in the case of multi-access edge computing, MEC servers are distributed around cells. In practice, it is advisable that when two or more users from different cells interact (play virtual reality games together, stream with each other), they should be served by the same MEC server to avoid a third-party aggregation server, which typically exists in the cloud [79].

Further, many researchers suggest placing edge servers as close as possible to access points to minimize latency. However, this approach raises concerns about spending CAPEX (capital) and OPEX (operational) funds. In remedying those costs, the authors of [80] investigated the best balance between QoS offering and cost reduction in 5G-MEC servers placement.

Furthermore, an important criterion to take into consideration when distributing edge resources is robustness [81]. This last is defined as the ability of a system to survive or function normally despite multiple edge nodes failing or being attacked. The resource distribution must be robust so that if an edge node dies, there should be another one in that region that can replace him. Depending on budget constraints, this placement approach may compromise users' coverage for failure resilience [82]. Additionally, another safety factor to consider when placing edge resources is uncertain or unexpected workload handling. To address this issue, the authors of [80] suggested learning about workload patterns before deciding on edge server placement strategies.

Besides physical resource placement, virtual machine placement is equally important. A simple technique to arrange VMs is to use fewer physical edge servers to place as many VMs (virtual machines) as possible in a few physical edge servers to minimize the number of active servers and therefore reduce the consumed energy [83]. However, this approach can create more congestion on the network because this procedure will lead to more VM migrating to follow widespread demand and users' mobility [84].

In addition to resource placement, resource migration is a key mechanism for balancing the load on edge nodes and accommodating the mobility challenges that exist in EC. When migrating VMs, the researchers in [85] propose using artificial intelligence models to predict user mobility, allowing VMs to migrate proactively before new workloads arrive. In addition, the following are the main three items to be considered when VMs are migrated with user mobility:

- The handover effect, one technique to lessen the frequency of this effect in a VEC situation is to apply an

intelligent server placement strategy, in which vehicle resources are transferred based on the user's movements, thereby enhancing resource availability [86].

- The task deadlines and workload should not just be shifted to the closest server but also to the strongest one that can help maintain the deadlines [87].
- The cost of migration, the edge service provider should study the users' paths to place services in a way that reduces the overall communication costs [88].

3) RESOURCE PROVISIONING

Resources provisioning is the technology that binds the quantity and quality of resources with users' desired quality of service. In EC, provisioning resources requires planning, estimating, and pooling the necessary amount of physical and virtual machines, along with their exact customization in terms of processor, memory, and network interfaces, all before they are passed to the scheduler to be used by the upcoming request [89].

However, unlike Cloud Computing, in which the costs of the resources are likely to stay steady, the edge environment is well known for its spatial-temporal variation in prices; thereby, resource provisioning actions must meet the recent changes and better balance the QoS with the costs [90]. A good example of spatial changes can be seen in the case of vehicular edge computing, where resources should be provisioned according to traffic [91]. To illustrate the importance of resource provisioning, consider the case of an edge application provider that rents servers from an edge infrastructure provider (EIP). In renting edge infrastructure, on one side, an over-provisioning case can result in a loss of energy and money. On the other side, an under-provisioning case can have a destructive impact on the offered QoS and will also result in a variety of overflow accidents [92]. Therefore, resource planning is highly dependent on knowing the available budget while estimating the demand by understanding edge client behavior patterns [93]. Additionally, an EIP may run out of resources in some locations or at certain times. To overcome these constraints, collaborative resource provisioning across various EIPs can make edge services available everywhere [94]. Another option for dealing with edge infrastructure constraints is to employ public cloud backup services [95], which can keep an application running even if the edge infrastructure runs out of capacity.

Furthermore, the resource provisioning strategy must be aware of the preparedness of resources. In fog/edge, nodes are characterized by high variability in terms of connectivity and availability. Given that, a continuous resource monitoring technique should be adopted [96], within which the aliveness and readiness of edge nodes are predicted by analyzing parameters such as the battery level, the movement patterns of edge nodes, etc. The monitoring of resources is conducted by a group of edge nodes that aggregate information about the states of the surrounding ENs [97]. One of the resource monitoring techniques is overlay gossip [98], which is widely used in wireless mesh networks [99], where a set of nodes

distribute data to the whole network without overloading the system, and the nodes' correct reaction to that data is interpreted as a sign of their aliveness.

4) RESOURCE POOLING

Creating a pool of resources that can be provisioned or scheduled according to the coming requests is known as resource pooling. Resources pooling aims to group heterogeneous edge nodes and arrange them into a coherent community by allowing them to interact and use each other resources (computation & networking) [100].

One of the ways of creating a sufficient pool of resources is to encourage multiple edge infrastructure providers (EIPs) to collaborate. In light of that, the work [101] proposes a game theory cooperative approach. The game's goal is to reduce the overall service latency by creating a coalition of multiple EIPs and rewarding the coalition members based on their contributions. Besides EIP/EIP collaboration, a MEC/Cloud collaborative was explored by [102], where the MEC provider could buy resources from the Cloud when he had an over-abundance demand. In the counterpart, the cloud provider could buy MEC resources to offer premium QoS to his client. Also, resource sharing is crucial for some edge computing architectures, for example, vehicular edge computing (VEC), where there is always a need for portal vehicles to lease their resource for the benefit of all, as they are being rewarded in return [237].

In Fog Computing, resource discovery and selection mechanisms need to be implemented to create a pool of resources from the massive Edge heterogeneous nodes. Resource discovery or node discovery helps locate new resources and add them successfully to the pool [104]. The thesis [105] is an excellent work that covers discovering fog nodes in the surrounding using customized WiFi beacons techniques. Another way to discover fog nodes is to provide them with a metadata description that makes them known to the other fog nodes [106].

B. COMPUTATION OFFLOADING

Computation offloading is a branch in computer science that deals with whether to run a process locally or send it to be processed by a commodity server outside. Computational offloading gained popularity with the rise of mobile cloud computing [51]. An incapacitated mobile device always requires more resources to run sophisticated applications, like Google Assistant or Apple Siri. As a result, those voice recognition tasks are offloaded to the Cloud.

Nevertheless, as interest in edge computing has grown, the question of offloading has become more prevalent than ever, as well as it did take on new forms. With EC, offloading is not only vertical or unidirectional but also horizontal, from IoT device to IoT device, from the edge server to edge server, and from any IoT or end device to any destination server in the continuum mist-fog-cloud. In [107], the authors presented a literature review answer to the central questions

in computation offloading, which are: When and where to offload? And according to what measurement should the decision be taken? In that, offloading entails selecting appropriate resources, filtering them, and deciding which ones are the most suitable for that giving task [108].

EC recognizes four types of offloading directions, listed below:

- End-device-to-End-device, End devices close to one another can collaborate, as IoT and end-user devices are becoming more powerful. Tasks are executed locally if possible or forwarded to light-loaded collaborative IoT devices in the surrounding [109].
- End-device-to-Cloud, because not all tasks are time-sensitive, incorporating the Cloud into the offloading equation can significantly increase the system capacity [110].
- End-device-to-Edge-to-Cloud, also known as hierarchical offloading [111], is a technique in which an end-device sends requests to the most appropriate edge servers, and the ES makes the decision on which parts to execute and which to offload to the Cloud.
- Vertical and horizontal offloading, end-devices can simultaneously transfer tasks vertically (to edge/fog/cloud) and horizontally (to neighboring nodes). This offloading type is well illustrated in VEC (Vehicular Edge Computing) [112].

The following sections examine the different aspects that can influence the offloading choice, ranging from task studying to the decision-making process to various used offloading algorithms.

1) WORKLOAD STUDYING

Before a task can be offloaded, it must first be understood and studied. Almost any task can be represented as a DAG (Direct Acyclic Graph) with multiple interdependent sub-tasks (see Fig. 9). Given the limited resources of edge nodes, effective task partitioning methods are highly valued in EC. The ultimate goal of task partitioning is to reduce the main task executing latency [113], which can be accomplished by creating as many parallel subtasks as possible. However, reducing latency in task partitioning should be accompanied by lowering communication costs, as distributing subtasks on many edge nodes may cause network congestion, as well as an increase in the probability of bits errors in the data transmission process [114].

Besides, while most theoretical studies assume that the complexity of a task is known, in practice, it is usually unknown before the task is executed. Consequently, the offloading brain should always estimate the runtime of each task before carrying it [115]. Additionally, because tasks are divided into interdependent subtasks, the offloading process should take those dependencies into account to reduce transfer delays between subtasks, as well as to give priority to subtasks that are required to complete other subtasks [116].

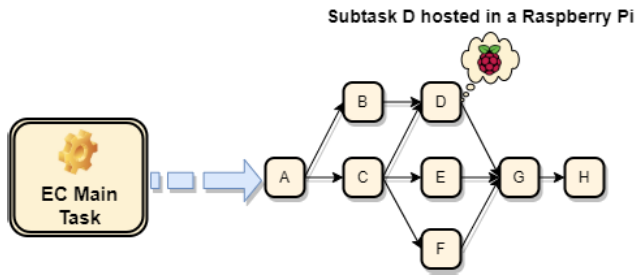


FIGURE 9. A task divided into a DAG, the result of both executing C and D is required to run the subtask D in a Raspberry pi.

2) DECISION MAKING

Making the correct offloading decision is a complex problem, and it should only be opted for if necessary since sending requests to the edge/cloud can cause transfer delays. When offloading a deep learning inference task, the effort [117] recommends employing an intermediate layer to measure the accuracy of a neural network (NN), where the authors suggest to stake to it if it is larger than a threshold, else send the rest of the NN to the Cloud to be processed by a more extensive neural network. Similarly, the reference [118] proposes using an estimator to determine whether a small NN was sufficient or a larger one was required. Meanwhile, when outsourcing a NN inference to the Edge, the decision should be taken when the NN is at a layer with a small number of neurons to reduce data transfer costs [119]. Moreover, one of the criteria that influences the offloading decision in the network environment is sending packages to congested networks may add extra delays to the EC tasks [120]. Within, using data compression techniques can help reduce network congestion when offloading [121], although they may add extra latency charges due to compression and decompression delays.

Generally, one of the most challenging aspects of FC is dealing with uncertainties when the system operates in a black-box environment. The task offloading assigner is unaware of the computing capabilities of the surrounding fog nodes. In that scenario, [122] proposes a Coded Computing approach based on the map-reduce model [123], which splits jobs into sub-jobs, each of which is sent to multiple edge servers, with the first completed replica of a sub-job being the only one taken into account, preventing the system from encountering some slow or untrustworthy servers. Additionally, [124] discusses another study that aimed to perform well in those uncertain environments (unknown state of nodes, lack of feedback from the environment), where a reinforcement learning approach was used to learn to adapt to those uncertainties.

Computation offloading can benefit significantly from resource monitoring, knowing that without a resource orchestrator that tracks, detects, and selects the appropriate quantity of resources, the offloading decision will remain unidirectional and unaware of dynamic changes in resource pool [125]. In that process, several efforts focus on joining computation offloading with resource scheduling [126].

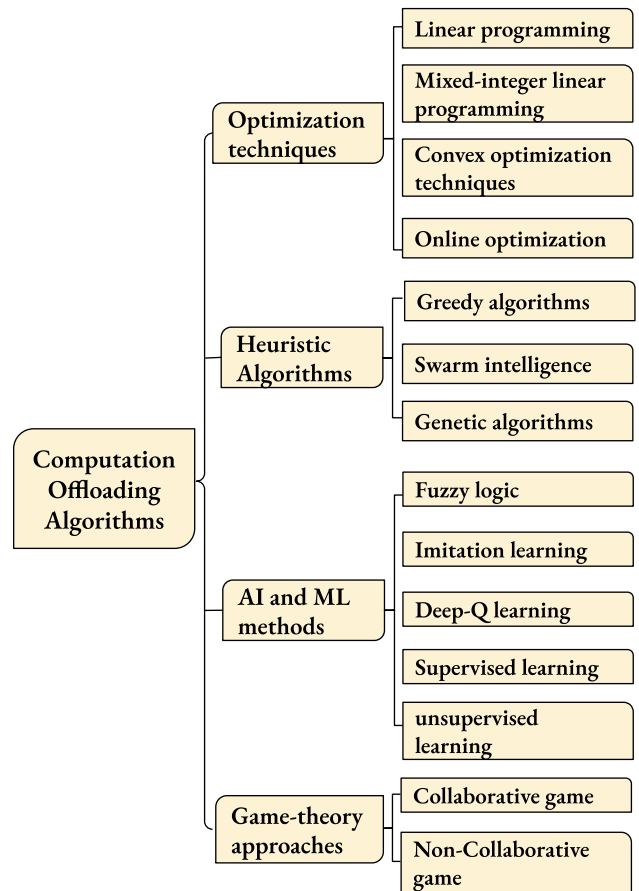


FIGURE 10. Computation offloading algorithms diagram.

Meanwhile, the offloading decision-maker is responsible not only for improving performance but also for maintaining system stability [127]. This stability is measured by the queue state of the multiple edge nodes that receive workload; when a receiving queue of an edge node is exhausted (hectic or converging to a situation where received tasks cannot be organized), the assigner must bypass those types of nodes.

Furthermore, the quality of experience (QoE) is a crucial criterion that needs to be addressed in offloading decision-making process. For example, offloading a video stream job is assessed by the low latency and high throughput that an Edge Service Provider (ESP) can supply [128]. One of the approaches used to increase user QoE is predictive workload [129]. Many edge-enabled apps use predictive information about upcoming workloads to perform parts of their tasks even before the user asks for them.

3) OFFLOADING ALGORITHMS

When taking the offloading decision, many mathematical algorithms and methods have been proposed in the literature [12]; we highlighted them in the diagram Fig. 10.

C. DATA MANAGEMENT

The act of acquiring, storing, distributing, and using data is referred to as data management. Data management aims to

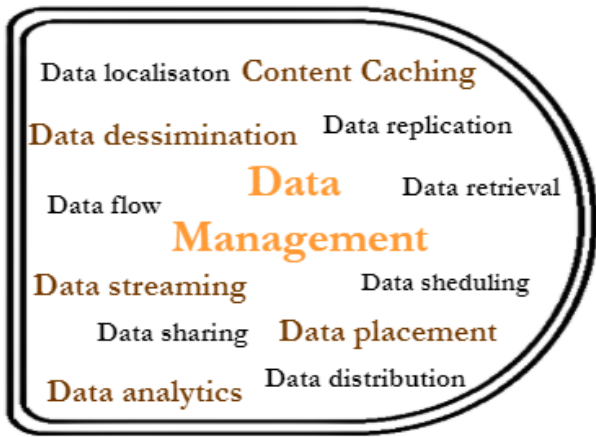


FIGURE 11. Data management keywords.

assist edge nodes in placing, sharing, analyzing, and retrieving data from one another to make decisions and take actions that maximize their overall utility while minimizing the end-to-end latency. However, data management in EC is more complicated than it is in CC. This difficulty is due to the large number of heterogeneous and widely distributed edge nodes that can adhere to any spatial topology distribution.

As a result of reviewing various and recent research related to data management in EC, we listed several keywords and terminologies related to data management in Fig. 11.

1) CONTENT CACHING

Content replication, also known as content caching, is an old and essential concept that encapsulates the idea of delivering content close to where the user requested it. In content caching, several servers were installed at the network’s Edge, creating what is referred to as a CDN (content delivery network) [42]. Further, caching content aims to optimize cache hit reward, measured by the number of times users request data stored on edge servers while lowering the cost of using those servers [130]. The caching problem is modeled as a multi-objective optimization problem with many parameters, as illustrated in Table III. This problem is difficult to solve in general. However, it can be approximated to a single objective optimization problem by considering the costs as constraints or by employing extra weight variables to control the optimization preferences between the rewards and the costs.

In Edge Computing, the main two factors that influence the caching policy are content popularity and edge environment capacity.

In an edge environment, ENs are defined by memory and connectivity constraints. As a result, the caching policy at the Edge should consider load balancing among different edge servers to improve fairness in terms of exploitation and avoid exhausting a single node, or a group of nodes [131]. Additionally, when deciding on a caching policy for MEC-based caching, it is critical to consider the connectivity capabilities of base stations as well as bandwidth limitations [132]. Another capacity criterion is stability. In [133], the cache hit

TABLE 3. Caching parameters in edge computing.

Caching Parameters	Description
System component	The number of ES, and the number of users covered by each ES. The number of files, and the size of each file.
Constraints	The Memory and bandwidth restriction imposed by each ES
Decision variable	The Binary variables refers to the decision of caching a file(s) in a giving EN(s) at a time slot.
Cost	Costs associated with the use of a bench of ESs in giving period of time.
Reward	The reward gained from users requesting files from ESs.

and system stability is optimized concurrently to maximize cache capacity and improve the overall system robustness.

Moreover, there is no doubt that the rewards obtained from cache hits are directly proportional to content popularity. Therefore, many recent studies have focused on predicting the popularity of content using machine learning and deep learning models. Some of those studies include K-mean [134], GRU (Gated Recurrent Unit) [135], and reinforcement learning methods like the Multi-armed bandit that balance data exploration in finding the liked content with data exploitation in caching the in-demand content [136].

In the last 20 years, the telecommunication industry has grown tremendously, and one of the main reasons for its success is infrastructure sharing [137]. As a result of this collaboration, mobile users enjoyed a better QoE, and the mobile infrastructure was exploited to its full potential. Similarly, MEC researchers have begun studying the possibility of sharing MEC infrastructure between different MEC providers (Fig. 12 depicts a simple architecture for a MEC providers’ data caching collaboration). In this collaborative caching scenario, a MEC provider’s added value is the cache hit gained from serving users subscribed to other MEC providers [138].

2) DATA DISSEMINATION

Data dissemination, or data circulation, is a fundamental element of the Internet of Things domain. Considering the case of wireless sensor networks (WSNs), [139], wherein a massive amount of data is generated each second, finding the best strategy to circulate data between the Edge and the Cloud is a challenging task. On the one hand, the strategy should avoid crowding the network. A solution to that is presented in [140], in which the authors introduce a new broadcasting protocol that uses neighbor knowledge around each EN to prevent redundant broadcasting. Alternatively, on the other hand, if the circulating data is an emergency one, it should

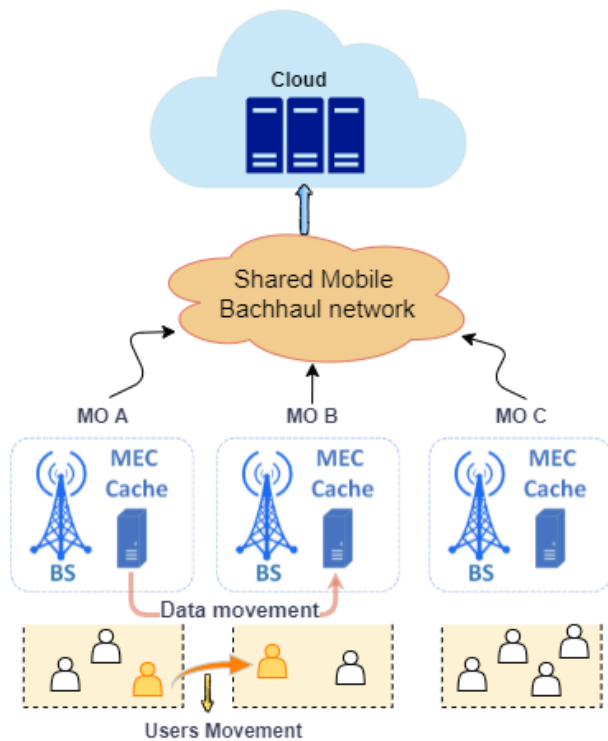


FIGURE 12. Sharing MEC infrastructure between multiple mobile operators.

be kept away from overcrowded parts of the network [141]. Supplementary data disseminated at the Edge must also avoid packet loss, for which the work [142] envisions a parallel push technique.

One of the fundamental aims of data dissemination is to maintain data availability, which is especially important when edge nodes enter and leave the network at any moment and intermittent wireless connections are the norm. In response to these circumstances, [142] proposed a strategy for regulating data transmission in fog computing using epidemic models. In addition, network stability is another goal of data dissemination. For example, in an Internet of Vehicles (IoV) scenario, according to the study in [143], the disseminated data between vehicles should use a restricted number of hops before being transferred to a roadside unit to ensure stability.

3) DATA STREAMING

As of 2020, video streaming accounted for 71% of all downstream traffic according to Comcast Cable [144]. Just on Twitch, 17 billion hours of live streams have been viewed. As a result, streaming is now more prominent than ever. Currently, hierarchical streaming from the Cloud to the Edge and then from the Edge to the covered users is the most popular architecture, particularly in companies like Netflix, which have as a primary goal the prevention of central servers from becoming bottlenecks [145].

In edge data streaming, users' QoE (Quality of Experience) is influenced by video quality (high, medium, low), latency, and bit rate variance. In [146], the authors present Elephanta,

an adaptive bit rate algorithm that adapts to the end user's preferences regarding average bit rate, rebuffering, switching, and buffer occupancy to select the appropriate bit rate. Likewise, to avoid the video flicker effect caused by changes in video bit rate, and in the context of vehicle fog computing (VFC), the authors of [147] modeled bit selection optimization as an actor-critic reinforcement learning (ACRL) problem.

Even though Edge Computing helps reduce upstream traffic to the Cloud, the need to upload data to the Cloud persists, particularly for applications requiring high-performance computing. [148] proposes a new stream sampling technique that only uploads to the Cloud what is required for incremental learning (IL). IL is defined as the ability of a model to learn from the newest data continuously. Alternatively, in downstream traffic, edge devices are expected to extract important features from streamed data without repeatedly processing it [149]. One of the objectives of stream processing at the Edge is to discover anomalies and novelties, such as determining the skyline sets in data streams [150], those represent the most significant data points or data objects.

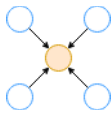
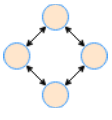
Furthermore, in EC, transcoding data streams into different formats and different quality levels is essential because of the heterogeneity of IoT and edge devices. Transcoding can be done proactively at the Edge to improve system efficiency, as demonstrated in [151]. Nonetheless, reducing transcoding time at the Edge is also critical for providing high-quality services. In that scope, [152] put forward a new method for reducing transcoding time by extracting information from the encoding time and saving it as meta-data, which is then used to reduce transcoding time at the Edge. Besides, [153] proposed the novel concept of collaborative live stream transcoding, in which viewers can transcode using their own devices and are rewarded in return. This method is a low-cost solution for reducing delays caused by cloud transcoding.

4) DATA ANALYTIC

Nowadays, vast amounts of data are generated every second, all over the place. There is a huge demand for data analytical tools; hence different data processing frameworks gained popularity, counting Hadoop [154], and Spark [155], due to their ability to extract information from large chunks of data. However, those tools require a lot of computing power and memory, making them more suited to the Cloud than the Edge.

With today's trend toward EC, Edge Analytics is a key technology that will meet the requirement of keeping data at the Edge. In order to maintain real-time performance and reduce congestion in the core network [156]. However, edge analytics faced numerous challenges regarding accuracy decline due to insufficient computational resources and data decentralization. Table IV illustrates the advantages and disadvantages of centralized and distributed analytics. Additionally, some studies such as [157] focus on balancing analytic accuracy and bandwidth consumption based on the degree of data decentralization. Similarly, another bandwidth-accuracy

TABLE 4. The characteristics of centralised and distributed analytic.

	Advantages	Disadvantages
 <p>Centralised Analytic</p>	<ul style="list-style-type: none"> • High Accuracy • Simple to deploy • Low bandwidth consumption • High execution time 	<ul style="list-style-type: none"> • High costs • Privacy risks • High latency • Lack of mobility support
 <p>Distributed Analytic</p>	<ul style="list-style-type: none"> • Low cost • Low latency • High mobility support • Low privacy risk 	<ul style="list-style-type: none"> • Deployment complexity • High communication resources • Accuracy degradation • Synchronisation requirement

trade-off occurs in cloud-edge analysis. Where edge devices can compress images before sending them to the Cloud to save bandwidth, but this can reduce accuracy, especially for highly detailed images [158].

Furthermore, edge analytics must reduce energy consumption when transferring data to edge nodes, especially when dealing with large data streams. Within that, the authors of [159] advance a technique for circumventing the limitations of ENs by simultaneously adjusting the video configuration (frame sampling rate, frame resolution) with the bandwidth allocation. Alternatively, [160] investigated a data discretization and sketching solution to overcome the bandwidth limitation issue.

Further to that, the main technology underlying edge analytics is Edge Intelligence, particularly collaborative Edge Intelligence [161] since many analytical tools rely on AI models, usually deep learning ones. Edge Intelligence is discussed in detail in section V.1,

5) DATA PLACEMENT

Data placement or data scheduling is the art of finding the best place to store data. Placing the data in an Edge-Cloud environment depends on the service’s nature that requires the data. In the case of edge application data, the ultimate goal of a placement strategy is to keep the latency demand while also reducing the cost of transferring the data [162]. Meanwhile, since the sensors and IoT devices are becoming more powerful, they can handle a reasonable amount of data, making it more appropriate to store the data at the mist/fog level rather than the cloud [163]. Additionally, [164] conducted a comparative study of three different algorithms in different network topologies, concluding that the algorithm that favors placing data at the Edge outperforms the one that

chooses a fog mapping or the standard cloud data placement. Besides, the effort [164] suggested the use of reinforcement learning for data scheduling in VEC, where the RL model can intelligently decide whether to place data locally, transfer it to a RSU, or forward it to a collaborative vehicle.

Before placing data, the placement strategy should anticipate data retrieval by shortening the retrieval routing path [165]. A good data location service is required to retrieve data; [166] proposes HDS (Hybrid Data Sharing), a fast data location service adapted to the MEC environment.

Further, data placement must account for the heterogeneity and mobility of edge nodes. Vehicle Edge Computing (VEC) is a good example, where mobility is the norm and heterogeneity influences how data is placed based on the type of vehicle (public, private, or emergency) and the importance of the content [164].

D. NETWORK MANAGEMENT

Network management encapsulates the advancements made in network infrastructure and architectures that adapt the network parameters to the new computing paradigm. The network management function focuses on monitoring, analyzing, and dynamically adjusting network status as needed. For edge computing to be successful, it must enable resilient and cost-effective network management methods, ranging from access control to traffic engineering to the adaptation of the newest network technologies. References [35] and [58] are two of the most notable publications that have surveyed the networking elements of EC. The following sections will discuss notable network technologies that assisted EC, counting network abstraction (SDN, NFV), radio access networks (F-RAN, C-RAN), and radio-resources allocation.

1) SDN

SDN (Software Defined Network) is an abstraction technique that decouples the network control from the data transmission by logically centralizing the network command functions in a NOS(Network Operation System) or an SDN controller. The NOS instructs the network’s forwarding devices on handling data packets (where and when to transfer data). For that to be possible, the Data Plan devices must support programmable switches that use the revolutionary OpenFlow protocol [167]. Overall, SDN allows the network to be more flexible and programmable. Fig. 13 illustrates the two layers of SDN architecture in fog computing. Besides, by designing the optimal path and employing the best packet forwarding procedures, SDN can be paired perfectly with the EC environment [168]. By taking all of these factors into account, [169] studied the characteristics and limits of SDN technology for edge-cloud computing.

Many scaling functions in edge computing, such as computation offloading or load balancing, necessitate the collusion of multiple network data plane components. In this regard, SDN architecture is helpful in EC because it enables the SDN controller to distribute bandwidth resources optimally across different data flows [170]. Additionally, because of SDN’s

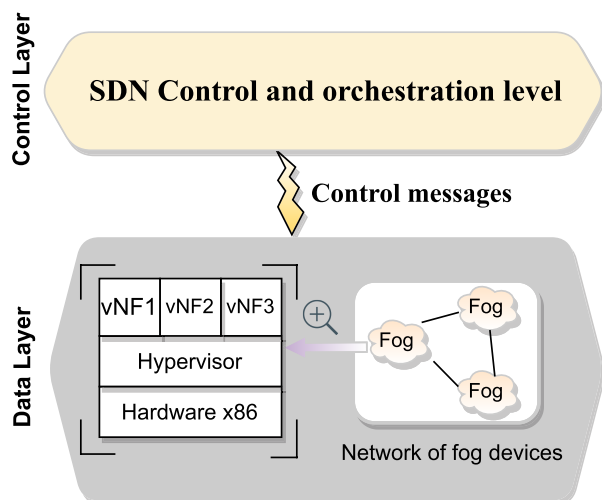


FIGURE 13. SDN/NFV architecture.

ability to have a global overview of network topology and users' movement, SDN controllers aid in determining the best edge server destination to host migrated services [171]. Other specifications, such as throughput or user preferences, can also be implemented in a mobility scenario to guide the SDN controller in performing the optimal handover for edge services [172]. Aside from mobility, a networking load balance mechanism can be performed by configuring Software Defined Network (SDN) switches across several edge servers [173], to ensure network resilience and protect against traffic spikes [174].

Moreover, SDN can help improve the performance of many edge applications such as CDNs (content delivery networks), where the SDN controller can take advantage of its network aggregated information to shortest paths between users and the content provider server [175]. Additionally, UAV air-ground communication is another application enabled by SDN abstraction [176], in which the SDN controller can utilize the predicted traffic load to perform data forwarding with the highest throughput efficiency. Aside from network decisions, the SDN controller can perform a variety of other required decision-making tasks, including a ML inference task [177], in which the NOS, based on the accuracy, can decide whether to transmit the ML task to the Edge or to keep it at the level of IoT devices.

Furthermore, one of the improved functions of using SDN architecture is traffic classification. Traffic classification is the study of categorizing data flows, encrypted or not, into multiple categories based on the packet byte information to differentiate video surveillance traffic from e-health and email traffic. Numerous studies have been conducted on the traffic classification problem, with many of them employing machine learning techniques [178]. With the help of traffic classification technologies, SDN and its integration with MEC could provide better network congestion handling [179]. When MEC servers benefit from their

communication with the SDN controllers, they can store the delayed tolerant traffic (for example, email traffic), and then redirect them after a reasonable delay. Alternatively, in the case of latency-critical tasks, the SDN controller can select reliable links that are less prone to failure for using them in critical traffic, such as e-health forwarding schema [180].

In the last few years, there has been an increased demand for SDN abstraction at the network's edges, which can be accomplished with the use of an SDN controller that tracks edge nodes in the data plan [181]. Traditionally, SDN controllers were deployed in the Cloud to provide a global view of the network. Although, With the shift to eURLL (enhanced ultra-reliable low latency) in 5G, SDN controllers have been relegated to the Edge (e.g., MEC servers, gateways, etc.) to facilitate efficient edge infrastructure provisioning. Additionally, ETSI sees MEC as a location for many SDN-based services [181], such as edge packet service management, data plane IP forwarding, adaptive routing for specific applications, etc. The Internet of Vehicles (IoV) is an excellent example of this expansion [182]; if the vehicles and roadside units (RSUs) are added to the data plan, the SDN controller can obtain agile information about vehicle movement changes, improving vehicle-to-X communication.

2) NFV

NFV (network function virtualization) is a network abstraction technique initiated by ETSI [183]. Traditionally, network companies used specific types of hardware for each component of the network (Firewall, Switch, Load Balancer). However, this approach was costly and inflexible. Now with NFV, network functions (NF) can be deployed as software applications on the top of a VM or a container hosted by a Blade server, and many NF can be hosted on one server (as illustrated in Fig. 13), making the deployment of network functions (NF) more flexible and scalable with fewer expenses.

Similar to how virtualization enabled CC, NFV is the technology that enables Fog Computing. Within fog commodity servers, network functions are deployed along with edge applications in VMs or Containers. NFV provides fog users with the ability to make service placement strategies that meet the end user's requirements [184].

Moreover, the main challenging problem in NFV is the placement of network functions (NFs) on the most appropriate fog nodes. The work [185] modeled this problem as an optimization one, where an NF deployed in a fog node is represented as a binary decision variable. The variables are chosen to reduce the time it takes to deploy, process, and communicate network functions. Additionally, As studied in [186], the NFs placement challenge can be paired with the optimal physical placement of fog nodes for offering the best network/computing services.

3) RADIO ACCESS CONTROL

In 4G/5G networks, the radio access network comprises two main components: the remote radio unit (RRU), which

performs radio frequency receiving/transmission tasks, and the baseband unit (BBU), which performs signal processing functions.

C-RAN (Cloudified/Centralized Radio Access Network) is a network architecture proposed by China Mobile in 2010. C-RAN group BBUs from different antennas in a remote central office, known as a BBU hotel, away from their correspondent Remote Radio Units (RRUs), the grouped BBUs are distinguished using internal routers that exist inside the BBU hotel. The C-RAN architecture supports lower power consumption, efficient operation, and higher reliability. However, one of the disadvantages of C-RAN is the long distance between the RRUs and the BBU pool, which causes extra latency between the end-users and the BBU pool. This issue gave birth to a new RAN architecture called F-RAN [187], in which each BBU hotel is constructed by grouping only a small number of RRUs, giving EC users more options for deciding on their optimal BBU hotel to run their services [187]. Likewise, the reference [188] conducted a comparison study between F-RAN and C-RAN, demonstrating that the F-RAN provides a faster response but at a higher cost. Fig. 14 represents the architectural differences between F-RAN and C-RAN.

4) RADIO RESOURCES ALLOCATION

Radio resource allocation aims to dynamically joint edge-users with their most adequate radio resources, allowing them to select the best transmission channel to maximize their throughput while augmenting their SINR (Signal to Interference plus Noise Ratio). In that, multiple orthogonal multiple access (OMA) techniques are exploited, including OFDM (Orthogonal Frequency Division Multi-access), in which every Edge User (EU) uses an orthogonal range of frequencies, and TDMA (time division multiple access), within EU access periodically a sub-channel. Moreover, studying the radio resource allocation problem is crucial in EC, especially when multiple edge users request EC resources. In that, the wireless channel condition has a direct impact on the QoS requirement of EC applications [189].

E. SECURITY & PRIVACY

Since the network edge environment is much different, EC security issues are considered one of its biggest challenges. In CC, data is stored in multiple large data centers, which are very well protected physically, with many guards, fences, and security protocols. In addition to physical security, Cloud providers are heavily investing in cybersecurity. In contrast, the conditions in edge computing differ; physical edge devices are much more dispersed and heterogeneous; this makes them more vulnerable to physical attacks such as the cooling system attacks [190], in which the attackers inject extra thermal load on the cooling systems. Additionally, the high data offloading and circulation at the network Edge made ESs (Edge servers) prone to cyber vulnerabilities [191]. Although, since the data is kept close to the end-users, EC provides better privacy protection than the Cloud.

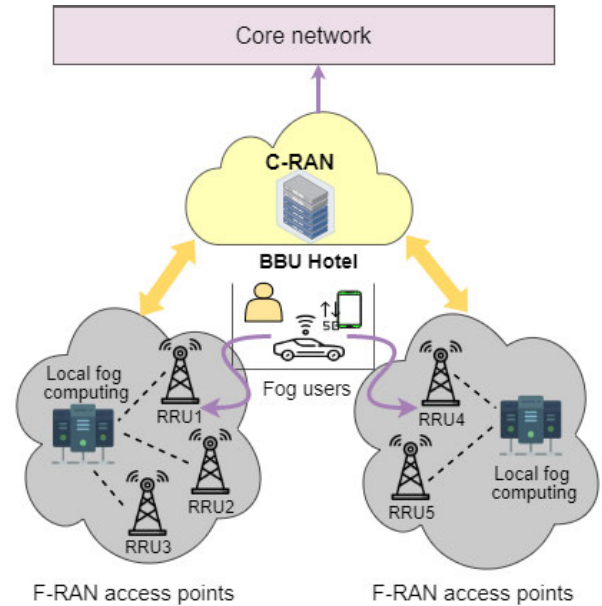


FIGURE 14. F-RAN architecture.

The following sections will cover the progress made in detecting and defending against EC cyberattacks.

1) ATTACK DETECTION AND DEFENSE

Attack detection is the study of multiple adversarial attacks and vulnerabilities that target edge devices and servers, particularly embedded ones. Because some of those ES are lightweight, they cannot use defensive tools such as anti-viruses or firewalls; therefore, there is a need to develop software anomalies detectors and testing techniques adapted for those edge nodes [192].

One of the most well-known threats to the edge servers is DDOS (Distributed Denial of Service). The DDOS attack aims to overwhelm the server's capacity by requesting the server by thousands of zombie machines. These attacks are significantly more efficient at the Edge than in the Cloud. In the CC, solutions like CDNs are utilized to spread content across numerous servers to relieve the load on a single server. On the other hand, Edge users will be unable to use this approach since they must connect to the nearest edge server. Nonetheless, there are some proposed defenses against DDOs attacks, counting the collaborative edge nodes solution in [193]; this approach suggests allowing the targeted server to redistribute the upcoming requests to his neighbors, reducing pressure on him.

Moreover, in order to detect network attacks, an intrusion detection system (IDS) is required, which can detect anomalous packet transmissions by analyzing historical data from packet transfers, processors, and memory [194]. Many machine learning models are being investigated in the literature to discover those hidden intrusion patterns. Among those are [195] and [196].

Meanwhile, due to the fact that most edge devices are fog/mist ones, those have minimal energy resources, some

attacks take advantage of this weakness by targeting ES batteries and causing them to consume a significant amount of energy. The Stretch attack [197], for example, sends data packets with headers that contain extended routing and looping paths, forcing these edge devices to consume energy from unnecessary data transmission routes. Another attack is Droplet attacks [198], in which the adversary sends an 802.15.4 data frame and then stops, putting the receiving edge server in continuous reception mode.

Further, a clone node attack is another type of attack, mainly exploited in wireless sensor networks (WSNs) [199], where the attackers create a clone node of a sensor using its ID. If the cloning attack is not detected, it can cause a false data injection attack [200]. In order to recognize the clone nodes, studying the Channel State Information (CSI) is commonly used as an effective defense [201].

Furthermore, edge computing is tightly coupled with computation offload since it is one of its main pillars. However, offloading makes edge devices vulnerable to attacks, for example, the Byzantine attack [202], in which a malicious receiver of the offload can corrupt the operation or change values as he sees fit. One of the proposed solutions to detect these types of attacks is homomorphic hash functions [202].

2) DATA INTEGRITY

Verifying the integrity and consistency of data that is distributed over a network is known as data integrity. Ensuring data integrity in EC is substantial because many edge devices can be manipulated or corrupted intentionally by adversaries or accidentally due to sensor malfunctions or transmission errors [203]. For verifying data integrity, the work [204] put forward a protocol called EDI-V. Their approach is based on giving each data block a tag before storing it in an edge node and then having a robust and trusted third-party server audit the data changes by comparing the initial tags with the newest ones. However, adding a third party is not adequate for privacy issues. In that, the efforts [205] proposed a distributed and lightweight auditing approach based on Merkle Hash Trees.

Moreover, an excellent example of data integrity issues is well illustrated in the case of data replications in CDNs, where consistency could be neglected for performance. In this case, the writers of [206] proposed a technique for locating corruption incidents that relied on generating signatures during inspection time and comparing them to the signature of the original data. Another method for ensuring data integrity is to use Blockchain-based architectures, which are well-known for their integrity and traceability protection [207].

3) ACCESS CONTROL

Access control is a process that allows only authorized users to access a given edge server. Since the authentication time adds to the total latency, EC authentication challenges lie in making access control as light as possible [208]. Supplementary, users' mobility and the wide distribution of

edge servers make the authentication process more challenging in EC.

One of the first user authentication work in fog/edge computing is [209], where edge users receive a master key that allows them to access any edge server using this master key along with the targeted Edge server public key, then having the edge server decrypt it to check the user's authorized status. Another approach for controlling access is the implementation of user tokens [210]. The edge server receives tokens, then analyzes its cryptography representation and compares them to those in the database.

Meanwhile, effective authentication is known for resisting and countering many attacks, counting privileged-insider and man-in-the-middle attacks [211].

4) PRIVACY

Today, user privacy is a controversial topic, especially with the expansion of camera surveillance systems and the exploitation of confidential users' data by social media platforms. Connected edge devices collect information about people to provide services that may jeopardize their privacy. In CC and EC, the journey of processing data passes by three main stages: data cleaning, data aggregation, and data analysis. The cleaned data is often less representative, with a smaller number of attributes than the rest. For privacy reasons, [212] put forward a distributed data cleaning algorithm that only asks users for data representation without transferring the actual data to the Cloud.

After cleaning, data analysis necessitates transferring to a centralized server(s) to run analytic models. Many undertaking approaches were given in the literature to prevent this process from violating privacy rules; lightweight encryption of data before transferring it is one of them [213], or queries encryption to protect the content of data [214]. Additionally, [215] suggests sending data with noise and training ML models with noised data. Similarly, [216] proposes multiplying the data with projection matrices before sending it to the Edge/Cloud for training. Besides the methods that change or de-identified data before analyzing it, differential privacy emerged as a powerful technique that helps defend against re-identification attacks [217].

For analyzing data, Federated Learning (FL) is one of the suitable leading solutions that help extract information from data without centralizing it [218]. FL allows end-users to share model parameters without sharing private users' data; each user trains the ML model locally and then offloads parameters updates to the exterior. A good example of using FL is in vehicular edge computing, in which car owners refuse to share their data with others [219].

5) EDGE APPLICATION SAFETY

Edge computing enables a wide range of applications, each vulnerable to a different type of attack depending on its nature. In the case of EC content caching, for example, some attackers request unpopular content regularly in order

to deplete the reservoir of caching servers and force normal users to request unpopular files from the cloud [220].

Meanwhile, many edge applications are AI-based, which are subject to high-level attacks that plan to fool them without any virus injection of a network intrusion. In today's world, AI base applications face a variety of adversarial attacks. Those attacks are divided into two types, white box attacks [221], within which the attacker has access to the model parameters, and black-box attacks [222], in which the attacker does not know the model parameters but generates adversarial inputs from similar models.

Overall, these attacks are not limited to AI-based edge applications, but they can also target AI models that were used to facilitate the primary functions of EC, such as AI-based offloading mechanisms [223] and Network Intrusion Detection models [224].

Further, some of the proposed approaches for preventing adversarial attacks include adversarial learning [225], in which NN is trained on the discovered adversarial examples. Another technique for increasing the NN is to train neural networks with data that has been perturbed by a small amount of noise with labels similar to standard clean data [226].

Furthermore, a positive indicator about the safety and the robustness of AI at the Edge has been addressed in the works [227], [228], in which neural networks adapted to the Edge using compression techniques (see section V.1) such as quantization or distillation are found to be more robust than their non-compressed counterparts.

F. BILLING AND PRICING

Pricing or billing of edge computing services is an important concern for any ESP (Edge Service Provider). In the EC market, there are four main market players: clients (individuals or businesses), ISPs (internet service providers), clouds (cloud providers), and ESP. These players are interconnected and can affect directly or indirectly one another [229]. The following section will discuss the various mechanisms for pricing EC services.

1) EDGE SERVICE PRICING

The ultimate goal of EC pricing is to find a strategy that maximizes edge services prices while also taking the concurrence and client's willingness to pay into account [230]. EC prices are regulated dynamically based on the supply and demand situation, with the supply usually known but the demand having to be estimated [231]. In pricing edge resources, the effort [232] suggests that the pricing of edge resources should be based on allocated resources rather than used ones in order to maximize profits. Overall, pricing aids in mitigating the abuse of edge resources, and it is an essential factor when selecting the best hosting ES [233].

Moreover, achieving full EC service coverage by a single entity is extremely difficult. As a result, collaboration among EIP providers is the norm to serve multiple users in a wide range of areas. Therefore, pricing policies that regulate

cooperations are required. For that, the authors of [234] propose a peer-to-peer payment system in fog computing, based on virtual coins, in which each fog nodes owners has a budget of coins, and whenever he requests some resources from his neighboring fog node, he pays them using his coin, those coins can then be transferred to real money. In a similar manner, a Blockchain credit values exchange for resources was adopted in [235] to regulate multiple edge nodes' cooperation.

In contrast to users paying ESP, in VEC, vehicle owners are compensated for providing their vehicle resources. In that, the offloading decisions from users to vehicles should be based on jointly minimizing the cost of running servers locally, as well as the cost of running them in a collaborative vehicle [236]. The amount of money the vehicle owners are paid during this process is determined by the MEC services demand [103]; if there is an increase in demand, MEC providers should raise the remuneration price to entice more vehicle owners.

2) AUCTION BASED PRICING

In an auction pricing system, each user or agency bids or submit a request for a quantity & quality of edge resources. The objective behind the auction is to design a payment system for all truthful users that reduce the difference between their valuations of the allocated resource and the proposed prices.

One of the famous auction methods is Vickrey–Clarke–Groves (VCG) [238], where competitors give valuations of an item or a service without knowing each other's bids. Based on VCG, many auction pricing methods have been proposed for edge computing, including [239], where the aim was to maximize users' rational valuation without considering any envy intention. Another auction method is McAfee auction [240], used in an environment with multiple ES (Edge Services) sellers and ES buyers. Additionally, as demonstrated in [241], the auction matching system can be used in computation offloading, where ENs accept or deny offloaded tasks not only based on their computation and communication resource but also based on the proposed bid by the users; if no edge node accepts the task, the user must augment their bid to be competitive.

3) MARKET ANALYSIS

The pricing of edge services is determined not only by the number of users in a market but also by the competition mechanism among multiple sellers, which has resulted in the pricing problem usually being modulated as a game, typically the Stackelberg game [242]. Because there are three main players (ISP, ESP, EU) in the EC pricing game, the reference [243] designing two nested Stackelberg games, one between the ISP and EU and another one between ISP and the ESP. The EU (Edge Users) subscribe to an ISP for radio resources allocated by access points, plus edge services, while ISP pays the ESP for leasing their given MEC resources.

In 5G, many private operators plan to enter the market to meet the demand for MEC services in a region. To achieve

TABLE 5. EC pricing parameters.

Pricing parameters	Definition	Ref
Offloading ratio	The Edge/Cloud pricing determine which tasks portions should be offloaded to the edge and which should be completed in the Cloud.	[246]
Users' reputation	Discount on edge services, based on users' reputation.	[247]
Stability	The pricing strategy must remain stable over the long run, avoiding selfishness of players.	[248]
Time delay	When a task got delayed the users get discount.	[249]

that, they will need to purchase and operate MEC servers and rent backhaul connections from legacy telecom providers. In order to have a good return on investment, private operators must have a reasonable pricing strategy adjusted to QoS offered to their clients [244]. Besides, to dynamite the EC market, intermediate platforms are widely initiated, like the suggested one in [245], inside which Edge services owners can put to rent part of their edge resources that they do not need for the users who request them, in exchange, the platform receiving a commission fee on each transaction.

4) PRICING PARAMETERS

Many measurements influence edge computing service pricing. Those parameters are represented in the table V.

V. ENABLING TECHNOLOGIES

A. EDGE INTELLIGENCE

Edge Intelligence (EI) is the study of the co-convergence of AI and EC. EI field highlights and studies the optimization methods that allow AI models to run efficiently in the edge environment in one direction and reviews the various methods that help adapt the Edge to AI application in the other direction. There are many challenges that the EI field is attempting to overcome in order to allow a much happier and more satisfying union between the two technologies (EC & AI). Our survey focused on presenting the various EI methods that enable AI models to be adequate for EC conditions [250], counting pruning, quantification, knowledge distillation, hardware acceleration, etc.

1) A BRIEF INTRODUCTION TO AI

AI (Artificial Intelligence) refers to any program that can act like a human when faced with complex tasks. Machine learning (ML) is a subset of AI in which intelligence is acquired by repeatedly interacting with the environment. A machine learning model is a system that improves by learning and correcting its mistakes as it goes. Machine learning is classified into supervised, semi-supervised (reinforcement learning), and unsupervised. Meanwhile, Deep Learning (DL) is a subfield of ML that employs neural networks (NN) as the base of its models [251]. NNs are made up of many neurons with weighted connections connecting them (see Fig. 15). During the learning process, the network attempts to modify the weights using an optimizer such as gradient descent [251], to improve model performance. There are several well-known neural network architectures, including CNNs (Convolutional Neural Networks), which are used primarily in image processing and object recognition, and RNNs (Recurrent Neural Networks), which are used to analyze time series.

2) PRUNING

Pruning in EI refers to the act of removing or deleting elements from an AI model, generally in the context of deep learning, elements such as neuron(s), weight(s), filter(s), and even entire layer(s) from a NN, for the reason that those elements have a negligible impact on the Neural Network construction or performance. Fig. 15 shows a case of pruning neurons and weights in a FFNN, plus filters in CNN. As a result of pruning, neural networks' computation and memory sizes are reduced, making them more adaptable to edge nodes. To answer the question, what should be pruned? Many studies intend to discard weights or neurons based on their magnitude or influence on the loss function; typically, this magnitude is calculated mathematically using the derivative of the loss function on a given weight or neuron. Then, various norms, such as the standard L1 norm [252], or the nuclear norm that gives more sparsity [253] can be used to measure and compare those values; after comparing the various normalized magnitudes, the goal is to discard a given number of the least important ones.

Meanwhile, due to the changes brought by pruning, it is necessary to calibrate the portion of the NN connected to the pruned elements. The rectifications in the case of a pruned neuron are represented by deleting the weights connected to that neuron. Unfortunately, pruning also results in a burst degradation of accuracy. In this case, to preserve previous NN knowledge, [254] proposed redistributing the pruned part of the NN parameters (weights, biases), using linear regression in a way that makes the sum of the outputs of the pruned layer approximate to the total outputs of the old layer (before pruning a neuron). Alternatively, to avoid training the model again after pruning, it is more effective to prune the NN during the training phase [255].

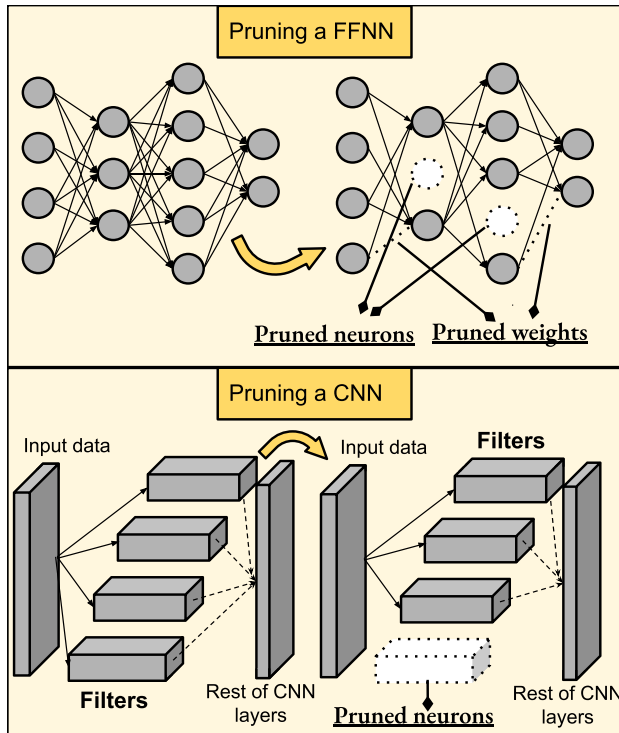


FIGURE 15. Pruning.

In CNN, convolutional layers or filtering layers perform most of the computation (more than 85%). As a result, they are the best-targeted elements during the pruning process. Within that, the effort [256] modeled the filter pruning as an optimization problem that seeks to minimize the cross-entropy loss function while having as a variable the binary decision vector of whether to keep a filter or not. An improvement in [257] suggests considering decision vectors as Bernoulli probabilities to facilitate the solving of the pruning optimization problem. Additionally, in [258], an advanced pruning method is used to address the goal of selecting a pruning strategy that achieves the best accuracy/speed trade-off.

3) SPARSIFICATION

Sparsification refers to the act of neglecting some NN values because they do not have enough impact on the NN inference result. Sparsification is a term used to describe the process of pushing a neural network variable to 0 if it is less than a certain threshold. Pruning and sparsification are similar but not identical. Although pruning targets entire elements, sparsification, on the other hand, typically targets values within elements. Moreover, the work [259] presented a sparsification method based on a probability formulation that depends on the weight magnitude. Additionally, the similarity to other values is another justification for the sparsification of some values. In CNN, the effort [260] introduced a kernel values sparsification technique within if two filters have similar absolute values in the same channel, the value of the one with the most negligible magnitude is set to 0.

Meanwhile, sparsification can benefit greatly from weight matrix factorization, as demonstrated in [261]. By decomposing the weight matrix into SVD (singular value decomposition) ($W = USV$) and employing a sparsification technique that allows many singular values to converge to 0 because they are unimportant in comparison to the others, this feature maximizes the number of multiplications by 0, and thus reduce the computation time.

4) QUANTIZATION

Quantization is the process of downsizing the bits' representations of a neural network's elements (weights, neurons, and activation) from a high precision representation (32 bits) to a lower precision representation (8 bits). Quantization's goal is to reduce the hardware execution time of a NN training/inference phase. In quantization, real variables (with 32 bits) are typically rounded to the nearest number in the new low representation; however, rounding to the nearest is not always optimal, particularly during the training phase, because it can disrupt the NN learning process, as demonstrated in [262].

Quantization can be classified into three types. The first type is a fixed bit variable quantization [263], in which the required number of fixed bits to represent all numbers is calculated based on the highest and lowest float numbers that may exist in the NN. The second type of quantization is uniform quantization [264], with each variable represented in the space \mathbb{R} based on its size using a Uniform Distribution. The third type is non-uniform quantization, which is similar to uniform quantization, but it uses a different distribution, typically a Gaussian one [265].

Meanwhile, one disadvantage of quantization is that it reduces neural network accuracy and makes the training difficult due to non-differentiability. In [266], the authors proposed a progressive quantization approach that starts the NN training with low precision quantization and gradually increases the precision, allowing control over the trade-off between resources consumption and performance. Similarly, an adaptive learning method was introduced in [267], which enables the number of bits representation to be modeled as a learnable hyperparameter.

5) KNOWLEDGE DISTILLATION

Knowledge distillation is a transfer learning technique that transmits knowledge from a complex model (teacher) to a simpler model (student) with fewer parameters. Knowledge distillation is an efficient solution for bringing high-performance neural networks to the Edge. Some examples of knowledge distillation applications include increasing image resolution [268], fast person re-identification in a camera surveillance environment [269] and visual dialog comprehension [270].

6) TINY MODELS CONSTRUCTION

Besides using compression or reduction techniques, other approaches suggest building a small neural network from

the start, knowing that not many intelligent tasks require a large NN. Many efforts in the literature have followed this direction, including [271], where the authors focused on reducing the dimension of filters in a way that does not affect the accuracy of the CNN model. Similarly, tiny TRU-Net was constructed in [272], and it was built using one GRU (Gated recurrent unit) cell. Aside from reducing NN size, changing the optimizer in the learning process can save a lot of computation time, as in [273], in which back-propagation is avoided in favor of a computation-friendly automatic adjustment of the NN weights based on the loss function. Likewise, some work like [274] even propose to bring back multi-layer perceptron, which require far less computation than the newest NNs.

Meanwhile, another method of reducing AI inference time is to combine two or more AI models into one, as was done in SSD [275], where the model does object boxing and recognition at the same time, saving much time in comparison if the task was done using two separate NNs. Along with model parameters reduction, input compression is considered a valuable variant for reducing NN training time, as seen in [276].

7) DISTRIBUTED EDGE COMPUTING

Aside from the compression and reduction techniques discussed earlier, the edge environment is well known for the large spread-out numbers of edge nodes that perform the computing tasks. As a result, distributed computing is an excellent solution for enabling EC to perform AI tasks. In this section, we will cover both distributed training and distributed inference.

There are two ways of distributing a deep NN inference, either vertically or horizontally. In the case of vertical inference distributing, also known as the IoT-edge-cloud collaboration, to execute a model inference, usually, the IoT device starts computing the first small part of the neural network for privacy issues, then send a small part of the rest to the Edge, and then the large rest to the cloud [277].

In the case of horizontal inference distributing, one of the first proposed distributed inference frameworks is MoDNN [278], a map-reduce mechanism that makes partitions of the input feature according to the number of ENs, where each node does a part of the inference calculation, and an aggregation layer comes to reduce the results. In addition, In the case of CNN, Fused Tile Partitioning (FTP) is proposed by [279], inspired by the idea that in CNN, the result of the dot product of the input data with each filter depends only on a specific region in the input data. Thus, good parallelism computing could be achieved if each edge device took a portion of the input feature. Moreover, in the case of an FFNN (Feed Forward Neural Network), the heavy calculation exists in the multiplication of weight matrices with activation layers. Therefore, the work in [280] suggests dividing the weight matrix into multiple sub-matrices so that each edge node is responsible for doing the multiplication of the small sub-matrix with the corresponding part from the activation

layer. Meanwhile, according to [281], while developing a distributed inference process, it is necessary to consider not just reducing the overall latency but also memory limits and communications costs related to parallelism computing.

Unlike distributed inference, distributed training is not a time-critical task for ML applications because the training phase is generally completed offline. Although most deep neural networks are trained in the Cloud (public, private), Edge Intelligence comes with the new perspective of training ML models at the Edge, following the fundamental goal of keeping the data at the Edge for better privacy and less network congestion. The standard method for training a ML model is to send the whole input data to a centralized server and then perform the learning process there. However, in EC, this approach is impractical since a single ES does not have enough memory and computing power to handle the training of an entire NN. Thereby, A new ML training paradigm has emerged called Federated learning (FL) [282]. FL considers distributing a NN' input data across multiple nodes, where each node trains based on his local data and transfers the outcome parameters to an aggregator. Next, the aggregator broadcasts the parameter changes for all collaborative nodes.

Further, Federated learning is plagued by high communication costs, and there is an urgent need to reduce data transfer overhead between collaborative training devices. Following that, [283] recommends sharing only the essential values of the gradient matrix since sharing the entire gradient matrix is an exhausting communication task. Alternatively, a different approach is to advise the aggregator to only receive updates from a small subset of ENs in order to reduce his throughput limitations. The work [284] chooses to select those collaborative nodes in a way that the staleness of the non-selected edge nodes at each timestamp is reduced.

8) ADAPTING THE EDGE TO AI

The second part of edge intelligence is about adapting the Edge to AI. In this section, we highlight various technologies that allow AI algorithms to be well received at the Edge; we mainly focus on the edge hardware adaptation for AI models since the software part is covered in the containerization section IV.3. According to the recent survey [285], the most frequently asked question in AI hardware adaptation is what type of edge hardware is optimal for hosting AI model inference and training? Is it CPU, GPU, FPGA, or ASIC (application-specific integrated circuit)?

Starting with the CPU (Central Processing Unit), they are the computer's brain; a CPU is known to be more flexible since it can do a variety of jobs without intentionally favoring one over another. Alternatively, GPUs (Graphics Processing Units) differ from CPUs in that they have more transistors in their arithmetic logic units, making them more powerful than the CPU in doing math calculations. Further, unlike a CPU, a GPU can have thousands of cores, allowing the GPU to perform well in parallelism tasks like training a neural network.

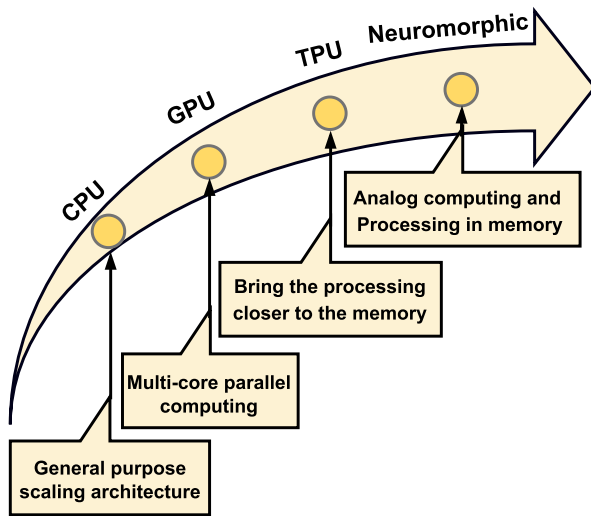


FIGURE 16. AI hardware evolution.

Aside from GPUs and CPUs, FPGA (Field Programmable Gate Array) is a candidate integrated circuit for hosting AI inference. By structuring, configuring, and interconnecting a group of logic blocks, FPGA makes it possible to perform well in some targeted logical functions. FPGA can accelerate AI inference by creating registers for NN input and weights values and multiplication and arranging addition blocks in a way that couples memory and computation and reduces data flow latency inside the circuit, thus speeding up the inference [286]. Because FPGAs are known for their high customization and future profiling, if designed well, they are regarded as one of the best hardware to host efficient model inference. However, FPGAs suffer when it comes to training AI models [287], especially in comparison to GPUs, due to their limited memory space. Although, as [288] pointed out, training in an FPGA is possible with the help of low-precision quantization and compression methods.

Further, ASIC (application-specific integrated circuit) is a hardware non-programmable architecture that is used for specific applications. ASIC chips are one of the best hardware for running AI inference at maximum efficiency due to their high design customization. Regrettably, the disadvantage of ASIC is that they are hard to design, especially when it comes to integrating and supporting multiple DNNs (deep neural networks) [289]. Some good examples of ASIC chips are Tesla D1 Dojo chips [290], and Microsoft TPU (Tensor flow processing unit) [291] which is specifically designed for the well-known TensorFlow framework models.

In comparison, many types of edge hardware can be used to accelerate AI training or inference fully or partially, ranging from high flexibility to precise adaptation. Based on the benchmarking done in [292], in terms of training, ASIC (TPU) and GPU outperform other types of hardware, and for neural network inference, FPGA and ASIC are proving to be the best. Besides electric hardware, photonic AI accelerators are now making an appearance as candidate hardware for hosting AI models [293].

In addition to digital hardware, analog circuits, including Neuromorphic ones, are emerging as a powerful alternative. Neuromorphic Hardware is a set of electrical circuits that mimic the human biological brain [294]. They are built with silicon-based artificial physical neurons. The primary electrical component in neuromorphic hardware is the memristor [294]. A Cross bare memristor perfectly replicates the analog operation of two matrix multiplication [295]. If neuromorphic hardware overcomes all of its current challenges, it will be the most advanced and adequate edge hardware to host an AI model since Neuromorphic computing is all about computing in memory [296]. Fig. 16 shows the evolution of AI hardware accelerators.

B. 5G AND ITS EMPOWERING TECHNOLOGIES

5G refers to the fifth generation of telecommunications networks, the latest group of advancements made over the previous 4G LTE (Long-Term) networks. 5G promises three primary services: eMBB (enhanced Mobile Broadband), mMTC (Massive Machine Type Communication), and URLLC (Ultra-Reliable Low Latency Communication). Certainly, EC is one of the most important technologies enabling 5g, as it is recognized by ETSI as an essential component of 5g [24]. With the help of MEC services, 5g networks can provide URLLC services to clients by hosting their services at the edge/access level of the network. Subsequently, MEC resources are utilized for hosting many 5G functions, for instance, cellular traffic prediction [297].

1) MIMO

5g is empowered by multiple technologies, including massive MIMO (multiple-input, multiple-output). MIMO is a radio technology that incorporates different transmitter and receiver antennas. MIMO employs spatial diversity to reconstruct signals received from multiple receivers or transmit a signal using multiple antennas. Integrating MIMO technology with edge computing allows multiple edge users to send computation requests simultaneously and in high efficiency [298].

2) IRS (INTELLIGENT REFLECTING SURFACES)

IRS (Intelligent reflecting surfaces) are signal-reflection surfaces that control the signal's transfer angle. IRS focuses energy by creating a beam directed toward a receiver. IRS can be deployed in various locations between the transmitter (antennas) and the receiver (mobile device), but when the IRS is deployed at signal source in base stations, it creates what is known as holographic beamforming. By reconfiguring the wireless propagation environments, MIMO technologies improve the offloading links (throughput, data rate) between edge devices and MEC servers by intelligently phase shift parameters of the IRS [299].

3) 5G OPTIMIZATION OF COMMUNICATION RESOURCES

5g networks are well-known for supporting massive IoT communication via cellular networks; however, this effort

can only be achieved by optimizing machine-to-machine and machine-to-x communications, referred to as energy-draining communication types. Consequently, multiple efforts in g focused on developing DRX (Discontinuous Reception) techniques for MTC (machine type communication) [300], those methods allow end-devices to save power by sleeping whenever there is no packet to receive. Some of this work is based on modeling packet arrival time using statistical distributions. Alternatively, as illustrated in [300], machine learning models can be used to predict future arrivals, allowing machine-type end-devices to schedule the ideal sleeping times. Additionally, the machine learning models for DRX could be hosted on MEC servers. Within [301] introduced Discontinuous Mobile Edge Computing (D-MEC), a DRX technique adopted for MEC servers.

Further to that, avoiding the execution of redundant or identical tasks is another technique to optimize ED resources. This technique is accomplished by reusing or partially reusing previously completed compute tasks, exploiting what is known by Computer Reuse Architecture [302].

Besides, the end devices at the mist level are energy-draining and battery-depleted. The groundbreaking WPT (Wireless power transfer) systems emerged as one of the solutions to this energy problem. Some of the efforts about integrating WPT and MEC were documented in [303]. The harvested energy from surrounding antennas within the IoT/mobile devices range can be used in offloading works to MEC servers associated with those antennas.

4) NETWORK SLICING

The 4G LTE networks are known for the concept of “one network fits all,” All users share a single network with the same performance. However, this architecture wasn’t ideal for many applications. Thus the 5G network intended to improve on 4G by introducing a new concept called Network Slicing. Network slicing aims to break a network into many slices, each of which is tailored to serve a specific group of applications with specified requirements (Latency, Peak data rate, Cell throughput, etc.). All of this is to provide adequate resource sharing in 5G networks [304]. A network slice is an end-to-end connection that contains elements from different network parts (RAN, Core-network, transport, and so on). Fig. 17 demonstrates how a network can be partitioned into several slices, each with its own set of customizations.

One of the most challenging aspects of network slicing is automating network slices based on the requests [62]. Further, to combine edge computing with network slicing, physical MEC resources are typically divided into multiple VMs or containers, each with its own set of capabilities designed for a specific slice. However, the drawback of having multiple VMs in a physical machine will reduce the throughput, therefore, lowering the latency of applications hosted in those VMs [305]. Also, network slicing contributed to the difficulty of scheduling MEC resources since in 5G networks, provisioning and resource selection should be combined by picking the most appropriate network slice [306].

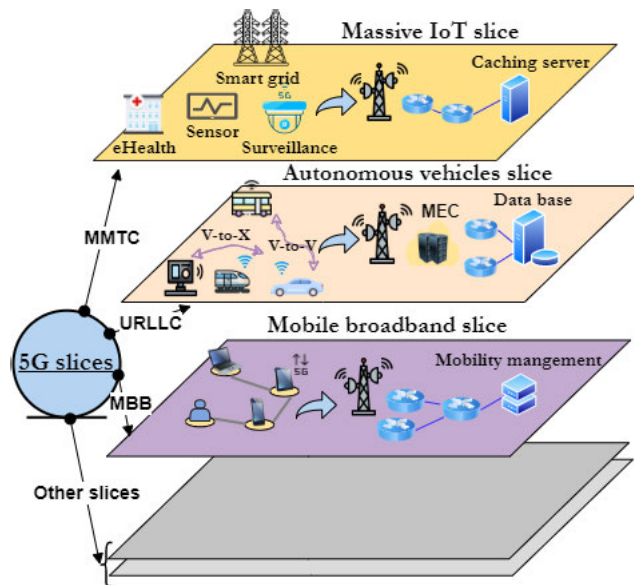


FIGURE 17. Network slicing illustration.

5) COGNITIVE RADIO: NOMA

In 4g legacy networks, radio resource allocation is produced via OMA (orthogonal multiple access) procedures (OFDMA, TDMA, etc.), in which users take turns accessing radio resources (frequency and time). However, with the rising pressure on spectrum utilization, OMA methods are becoming insufficient and inefficient. Consequently, the future of telecommunication networks (5G and 6G) is moving toward a non-orthogonal multiple access (NOMA) strategy. NOMA allows numerous devices to send data on the same band of resources (frequency & time), maximizing spectrum efficiency. NOMA is based on the superposition of numerous signals at the transmitter side, followed by interference cancellation via successive interference cancellation (SIC) at the receiver side.

NOMA is considered the leading technology that helps optimize the workload offloading process to MEC servers, as it enables multiple machine-type end-users to access 5g-MEC resources simultaneously. However, the main challenge in NOMA and EC is the establishment of the right radio/MEC resource assigning strategy that respects task delays while lowering total offloading energy [307].

6) 6G

The sixth network generation moves toward the polarization of intelligence usage in the network from the core to the Edge. The 6g networks promise ultra-smart and robust network functions; however, those functions will be accompanied by large (computing and memory) consumption. Along with Fog and Cloud computing, Edge computing will play a critical role in hosting 6g functionalities while providing them with ultra-low latency. The 6g evolution is characterized by the ‘softwarization’ of many parts of the network, which leads to what is known as the convergence of computing and networking [308].

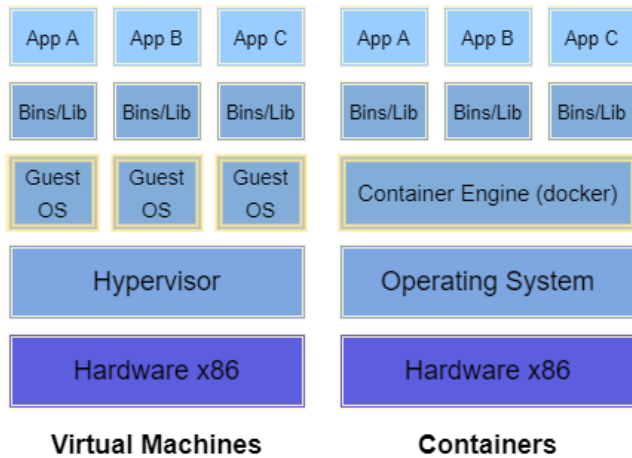


FIGURE 18. Containers and VMs architectures.

C. VIRTUALISATION, CONTAINERISATION

Virtualization is a powerful technology that enables cloud computing. Today, every data center uses virtualization to create large pools of resources (CPUs, memory, disks, network). For offering them to customers as scalable, consolidated VMs. Virtualization has changed the way people think about computing and communication resources. In cloud computing with VMs, computation has evolved into a service rather than a product. Similarly, in 5G, with NFV and SDN, network components are becoming network services. This section provides a brief introduction to containers and container orchestration and explains why containerization is one of the key enabling technologies of EC.

1) CONTAINER

Over the recent years, containers' popularity increased as they emerged as a promising alternative to virtual machines, leveraging the lightweight implementation of virtualization. Containers are as old as the Linux Kernel is. In 2008, Linux Control Groups (Cgroups) and Namespaces were combined to develop Linux containers (LXC). LXC aimed to create a complete OS-level virtualization technology that became a prominent Linux kernel feature. This Linux kernel feature was incorporated into many projects/organizations, and the most known of them is Docker. Fig. 18 depicts the differences between VMs and containers in terms of architecture.

2) CONTAINER VS VIRTUAL MACHINE

The main advantages of using VMs and containers are consolidation and elasticity. Consolidating workloads reduces hardware, power, and space requirements. Elasticity allows dynamic allocation of resources that are needed [309]. With VMs, companies are no longer required to own physical servers and accommodate peak demands whenever they occur. Additionally, yet importantly, due to their lightweight dependencies, containers offer higher portability than VMs. Alternatively, VMs provide a high level of isolation and thus better security [310]. Despite VMs on a physical machine

sharing resources, mechanisms such as virtual paging are implemented to ensure that each VM's resource is entirely isolated. Finally, as shown in Figure. 18, containers share a standard operating system, different from VMs, where each one could run under a different operating system, adding overhead in memory and storage.

3) CONTAINERS ORCHESTRATION

Orchestration is the automated configuration, management, and coordination of computer systems and services [311]. The goal of orchestration is to help manage complex tasks and workflows more efficiently. Container orchestration automates the deployment, management, scaling, and network of containers. There are many orchestration tools to choose from [311]. The orchestration tool manages the life-cycle of the running containers according to the specifications laid out in the container's definition file.

Kubernetes, Greek for the helmsman, is an open-source container orchestration tool developed by Google in 2008. Kubernetes's main responsibility is making sure that all the containers that execute various workloads are scheduled to run in physical or virtual machines [312]. Kubernetes is comprised of many components, but the main ones are:

- Cluster is a collection of nodes containing at least one master node while the rest are worker nodes.
- Node, also known as a minion, is a single host whose job is to run pods.
- Master is responsible for the overall cluster-level scheduling of pods and handling of events.
- Pods are an important feature and the basic unit of work in Kubernetes. Each pod contains one or more containers.
- Deployments, replicas, and ReplicaSets. A deployment is a YAML object that defines the pods and the number of container instances, called replicas.

4) CONTAINERS AT THE EDGE

Recent research shows that using containers at the Edge has many advantages. The advantages stem primarily from the low deployment time [313] and the quick migration time [314] provided by containerization technology. Container orchestration allows consolidating multiple IoT devices with heterogeneous hardware for an increased quality of service at the edge [315]. In that, containerization perfectly matches the edge environment where mobility and constrained resources are the norms. In addition, containers also help expand the elasticity and resilience of the edge Eco-System, as their advanced task recovery methods allow tasks to run uninterruptedly at the edge [316]. Moreover, when edge nodes send data to the Cloud, they do not typically need to send raw data streams. Instead, the node only sends critical information. This event-driven approach in EC can be tackled by forking containers by the orchestrator whenever needed.

Furthermore, with their lightweight and portability characteristics, containers are considered the best run time for edge/mist devices like SBCs (Single Board Computers) [317].

Meanwhile, in a MEC system hosting edge application, base station Handover needs to be coordinated with container migration as it was presented in [318]. However, one of the issues that containers at the Edge suffer from is the cold start problem [319]; it refers to the needed time to bring up a new container when there is no warm container available. The solution to this issue is to have warm containers available for usage by the event-driven application, except that this method may add extra energy consumption [319].

D. THE MOVE TOWARD FUNCTION AS A SERVICE (FaaS)

FaaS (Function as a Service) refers to the ability to decouple an application into a group of interconnected pieces of computation unit (code) known as functions. It is the next abstraction layer above software as a service in CC, and its goal is to provide cloud clients with a platform for running their software without regard for OS or any virtualization dependencies. The ability to create event-driven apps is one of the benefits of serverless platforms. With that, some functions such as sensing and abnormalities detection can be activated based on events, making serverless ideal for a wide range of edge-enabled applications, including precision agriculture, image processing, etc.

Meanwhile, present serverless platforms are more suited to the Cloud and far from being viable at the level of an edge node with limited computation resources. However, recent efforts such as [320] research minimize the issues in deploying serverless edge computing.

VI. EDGE COMPUTING APPLICATIONS

A. SMART CITIES

Intelligent and connected cities are the image of our near future. Smart cities encompass a wide range of subdomains, including smart buildings, smart farms, smart roads, smart banks (Blockchain), and so forth.

1) SMART BUILDINGS

Smart buildings are a sub-branch of smart cities that make buildings more efficient, smart, and dynamic. Smart buildings' appliances (kitchens, light fixtures, TVs) are intelligently monitored and controlled based on users' preferences. Controlling the building environment can be computationally expensive, and most modern intelligent functions require little delay interaction with the user; this makes EC a valuable tool for meeting the fast response and low-cost requirements of smart buildings [321]. Some advanced smart building functionalities include room occupancy estimation [322], video surveillance on outdoor [323] and person tracking and identification [324], etc.

Meanwhile, intelligent buildings have been accused of consuming much energy, although they also contribute to energy

savings by doing energy-saving actions like turning off lights or heaters [325].

2) SMART FARMS

Smart farming (SF), or precision agriculture, as a component of smart cities, employs the most recent and advanced ICT technologies to improve farm sustainability and profitability. Smart farming is focused on controlling actuators (motors, pumps, light regulators, and so on) based on various aggregated data from sensors (temperature, humidity, brightness, etc.). Moreover, UAV (Unmanned Aerial Vehicle) Computing is a type of application-oriented edge computing that is widely used in agriculture today [326]. The UAVs are deployed in the form of a swarm, outfitted with cameras and computation resources, where they fly over large farms to monitor crop health and plant stress. Aside from crop rentability, EC can be used to analyze farm animal behavior [327], which is vital for animal welfare and health.

Meanwhile, one of the challenges agricultural areas face is isolation and a lack of solid and reliable connections to the data network. Private edge computing and communication infrastructure present a valuable solution to enrich agricultural areas to address this issue. Another reason to rely on edge/fog computing is to protect the core network from congestion [328], as transferring all camera records and sensor data from different farms to the centralized Cloud will be a tiresome network task in the future.

Alternatively, one of the well-known low-energy communication technologies is LORA (Long Range), which is used widely in precision agriculture [329]. It consists of two components: Lora Gateways, which are responsible for transferring data from sensors, and a LORA central unit, which functions as an edge node for processing the acquired data.

3) SMART INDUSTRY

EC has perfectly aligned with the most recent industry 4.0 requirements; this integration is also known as industrial edge computing [330]. Predictive maintenance is an approach that new industries rely on to reduce their CAPEX and OPEX expenses. In it, the machine is equipped with various IIoT (industrial IoT) sensors counting temperature, vibration, and pressure sensors, which gather data that is then transferred to fog nodes to be processed there for predicting machines failures and errors [331].

Further, the fourth industrial generation intends to incorporate artificial intelligence (AI) into its manufacturing processes. Because industrial companies cannot transfer their private data (for example, videos from the production scenes) to the Cloud, they must rely on the edge [332]. Object recognition with robots [333], automated guided vehicles (AGV) [334] and human pose estimation [335] are some examples of how industry 4.0 is working to improve itself using EI.

Outside of industries, E-commerce enterprises are now in desperate need of real-time interaction with their customers [336], as delivering quickly with EC is one of the

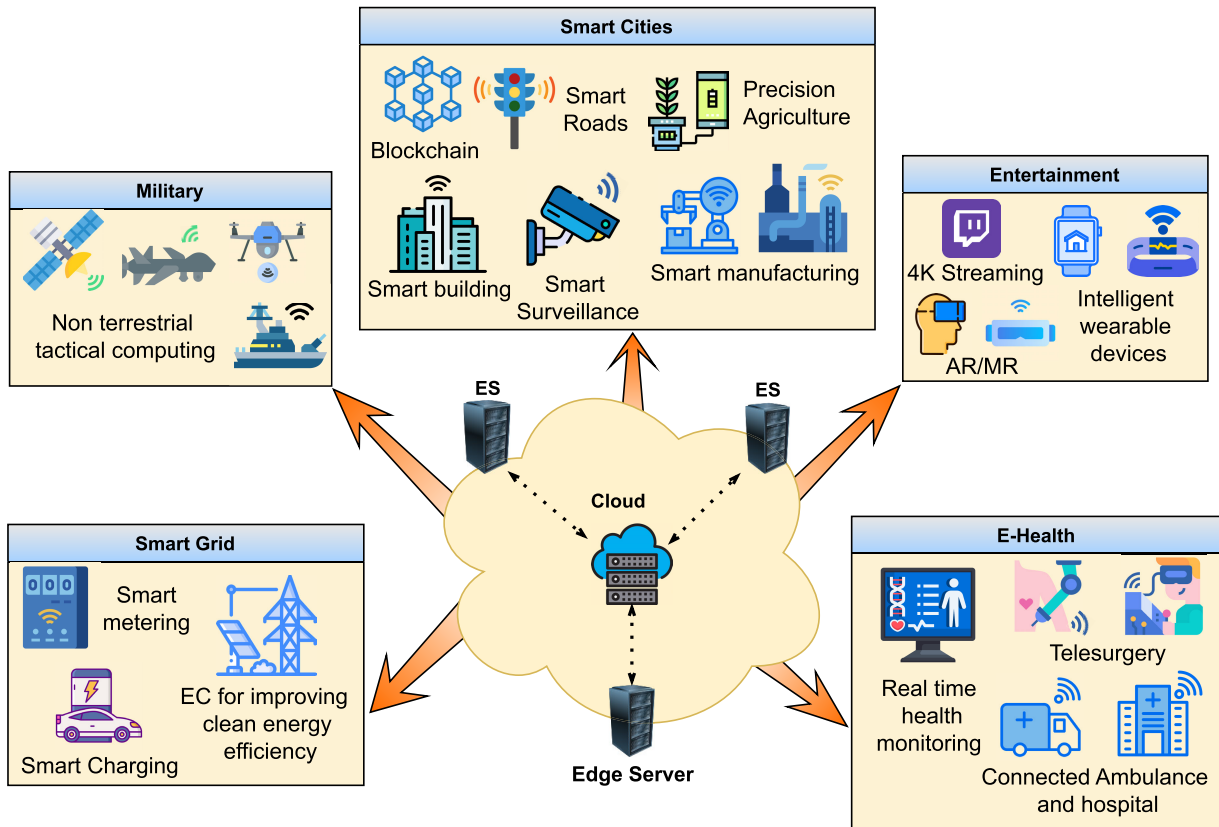


FIGURE 19. EC applications and use-cases.

best browser experiences they can provide to their customers. Likewise, EC can help secure in-store payment stations by upgrading their cameras with computer vision capabilities [337].

4) SMART GRID

Smart grid (SG) enhances electrical energy with efficient, flexible control of grid components through information and communication technologies (ICT). SG has three main branches, including smart grid metering and monitoring, intelligent control of grid functions, and effective integration of renewable energy sources into the grid [338].

Smart grids interact with edge computing via their energy management systems, which can be hosted in edge servers in a scalable and distributed fashion. One of the smart grid's primary functions is monitoring. Smart grids can monitor their grid equipment (voltage alarms, cables, towers, etc.) using EC. In the event of an overhead line failure, the smart grid can quickly restore power via its low communication with ES [339].

One of the primary smart grid energy measurement approaches is load forecasting, also known as electricity consumption prediction. The prediction is usually performed using ML models, and the training of those models can be accomplished using FL methods, in which each smart meter is connected to an ES that does the training on its electrical

data [340]. In addition, malfunctioning smart meters is one of the issues EC helps to resolve in SG [331]; with EC, smart meters can be empowered with intelligence at the Edge, allowing them to detect whether or not their monitored data is erroneous.

Moreover, in the intelligent grid control phase [341], the edge nodes can play a critical role in communicating with the various grid sensors and power sources for then sending the appropriate commands to generators and actuators in real-time.

Renewable energy integration is a component of the Smart grid; one of the services that VEC provides to EVs (Edge Vehicles) is the use of MEC servers represented by roadside units to calculate the best time and location for charging electric vehicles based on tariffs and battery level [342].

5) SMART ROADS

Edge computing made the new smart cities' roads and transportation systems safer and more intelligent. Smart roads, as surveyed in [343], aid in the spread of global awareness among road elements by employing an intelligent traffic management system, where vehicles assisted by EC can communicate with one another to maintain road traffic safety and equilibrium [344]. Using this Vehicle-to-Thing communication, it is also possible to treat special road scenarios more efficiently, for example, in the case of an accident or

emergency vehicles (ambulance, police car), where vehicles are commanded proactively to free up some road lines.

Based on the aggregated data from the road environment, one of the functions that will be hosted in ES is vehicle collision detection [345]; within ES can instruct and calibrate in real-time vehicle speeds and trajectories, as well as lighting systems to avoid collision accidents. Additionally, FC is used to enhance road cameras with functionalities such as vehicle detection and tracking [346]. Another issue in road safety is surface condition; the effort [347] proposed deploying a crowded surface sensing system with the assistance of vehicles, in which data is aggregated and analyzed using fog nodes.

6) SMART BANKS (BLOCKCHAIN)

Blockchain is an electronic transaction system invented in 2008, and it is free of any third-party controller (Banks, government, etc.). Miners, who work on solving a mathematically and computationally difficult problem known as the Proof of Work, perform transaction verification in Blockchain. Blockchain technology is the driving force behind smart banks. Because of the computationally intensive nature of mining tasks, Blockchain cannot be directly integrated into IoT devices. As a result, offloading mining tasks to the Cloud, Fog, or Edge represents a valuable solution [348]. Moreover, one of the challenges in Blockchain is pricing collaborative miners, as there is a need to improve the revenue of ECSP (Edge/Cloud service providers) while also protecting miners' investment gains from offloading to the Edge/Cloud [349].

B. E-HEALTH

By this century, electronic and information systems had already infiltrated the health sector, resulting in what we now call "e-Health." This new health paradigm is characterized by the use of electronic devices for diagnosing and treating patients and the widespread use of computers for collecting and analyzing health records. EC is here to benefit e-health in a variety of ways, including medical records storage, dealing with privacy concerns, and coping with long retrieval delays from the cloud [350]. Moreover, using fog/edge for continuous remote health monitoring of patients [351] will save a massive amount of network bandwidth. EC is essential for next-generation e-health applications, including Telesurgery operation [352], in which the doctor from their homes commands actions to be performed by robots in operation rooms; this communication requires a very low latency that can only be achieved with EC.

Further, with the increasing use of AI in all human activities, e-health is now widely benefiting from this intelligence in many tasks, as the survey in [353] highlighted the EI different deployment use cases in health-care systems. Further, some of those e-health-enabled tasks include arrhythmia detection [354], in which electrocardiogram sensors are enabled by Edge Intelligence to classify heartbeat. Another application is electroencephalogram

monitoring [355], or human brain activity classification, where recent advancements in embedded brain-computer interface (BCI) combined with EI allows for brain seizure detection to be performed [356].

C. ENTERTAINMENT

There is no doubt that edge computing, in conjunction with 5G networks, will transform the gaming experience, particularly VR and AR games. The two criteria gamers despise the most are higher pings caused by high latency and poor FPS linked to low computing resources. In the case of augmented reality, EC promises to improve these two gaming quality requirements by connecting with low latency gamers' AR goggles and mobile devices to high resourceful edge computing nodes [357]. In addition to AR goggles, another type of wearable gaming device is haptic ones [358], which provide sensing information to users whenever an action occurs within a virtual game, to come close to instant human reaction when touching a fire, those require Ultra-low latency, which can only be achieved using edge Computing. Additionally, With the mean of smart wearable devices, EC promises to enhance and improve edge users' entertainment experience [359].

Aside from gaming, EC enhances mixed reality applications, such as limited and visually impaired people assistance [360].

D. MILITARY & SPACE

A trend toward edge computing is a trend toward Tactical-Edge Computing. Data cannot be backhauled to a central office in military operations because network infrastructure is one of the first tactical targets in a war, rendering the Cloud, fog, or MEC non-existent or out of service. In a war scenario, the military is left with distributed edge nodes that must be consolidated and scaled up in a high fault tolerance environment before being used to augment other military equipment with computation resources [361]. Given that the EC favors decentralization, the work in [362] investigated the case of a Swarm of Drones as an effective source provider of FOG computing services.

Smart wearable devices (e.g., smart clothing, smart-watches) powered by EC have proven to be extremely useful on battlefields, as demonstrated by Microsoft's sale of the HoloLens smart glass to the US Army as part of a 22 billion dollar contract [363]. Another EC-powered military device is ground penetrating radar [364], which is typically carried by drones or low-flying aircraft. Adding local intelligence to these radars allows for an instant operation against underground detected objects. Similarly, EI can play an essential job in the surveillance of remote desert borders [365]. In a different scenario, maritime can perform rescue operations in the middle of the ocean using UAVs equipped with object detection models by using UAV-based Edge Computing [366].

Furthermore, because terrestrial edge computing (TEC) is vulnerable to disasters or tactical attacks, orbital edge computing (OEC) [367] is becoming a valuable backup in many

scenarios, particularly with the breakthroughs advancement made in enhancing space-ground data rate transformation and lessening of satellite manufacturing and operating costs. Nonetheless, as [368] highlights, OEC still faces numerous challenges, such as high-speed movement and channel condition changes. Meanwhile, many studies, including [369] and [370], propose outfitting satellites with embedded processing boards to perform tasks like data analysis and AI inference.

VII. EDGE COMPUTING CHALLENGES AND FUTURE CONCERNS

A. GREEN ENERGY

Climate change is one of the most pressing issues of the current decade. Climate change has compelled the globe to rely more on clean and sustainable energy. However, with today's increased electricity usage due to novel edge applications, the demand for integrating renewable energy as the primary source of EC energy consumption is at its all-time high. Although EC promises to reduce energy consumption pushed by cloud data centers, there is an increasing need for powering ES with clean energy and harvested energy approaches [371]. On the other hand, many efforts are working on reducing ENs energy consumption while also making offloading decisions favoring EC powered by clean energy [372].

B. STANDARDIZATION

Edge Computing has emerged as a compelling and vital paradigm for industry and research. Several standardization institutions have put up a lot of effort to create recommendations and references on how to integrate EC either from the cloud-Edge side or from the MEC-5g standards network side [373]. In terms of cloud architectures, the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have put in a lot of work to define Cloud/Edge technical architectures, software platforms, virtual machine and container management, and orchestration. ETSI is a prominent player in the 5g-MEC sector, and their numerous white papers have helped to standardize Multi-access Edge Computing platforms [54].

VIII. CONCLUSION

The world is undergoing a massive shift toward digital services. As a result, computing and memory resources are in high demand. Furthermore, novel applications like smart cities, e-health, smart grid, and others require resources (computing & memory) with low latency services and a stable network free of security and privacy issues. EC has emerged to provide all of this. We presented a survey on the evolution and construction of this computing paradigm as part of this work. We discussed how related technologies such as 5G, Edge Intelligence, and containerization had pushed the evolution toward keeping and handling data at the Edge. Finally, we investigated how EC will respond to future concerns such as green energy and standardization.

REFERENCES

- [1] Gartner: *Edge Computing Promises Near Real-Time Insights and Facilitates localized Actions*. Accessed: Jan. 2022. [Online]. Available: <https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders/>
- [2] *5G Network Transforms the Industrial, Transportation and Commercial Landscape With Real-Time AI and Accelerated Response Times—IBM Bus. Operations Blog*. Accessed: Mar. 31, 2022. [Online]. Available: <https://www.ibm.com/blogs/internet-of-things/iot-5g-transforms/>
- [3] *Edge Computing Market Share & Trends Report, 2021–2028*. Accessed: Mar. 31, 2022. [Online]. Available: <https://www.grandviewresearch.com/industry-analysis/edge-computing-market>
- [4] *Number of Papers Related to Edge Computing 2010–2020 | Statista*. Accessed: Jan. 2022. [Online]. Available: <https://www.statista.com/statistics/1193762/worldwide-edge-computing-google-scholar-paper>
- [5] W. Shi, J. Cao, S. Member, Q. Zhang, and S. Member, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Jun. 2016.
- [6] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, 2019, doi: 10.1016/j.sysarc.2019.02.009.
- [7] B. Ramprasad, A. Da Silva Veith, M. Gabel, and E. De Lara, "Sustainable computing on the edge: A system dynamics perspective," in *Proc. 22nd Int. Work. Mob. Comput. Syst. Appl. (HotMobile)*, 2021, pp. 64–70, doi: 10.1145/3446382.3448607.
- [8] M. Babar, M. S. Khan, F. Ali, M. Imran, and M. Shoaib, "Cloudlet computing: Recent advances, taxonomy, and challenges," *IEEE Access*, vol. 9, pp. 29609–29622, 2021, doi: 10.1109/ACCESS.2021.3059072.
- [9] P. Ranaweera, A. D. Jurcut, and M. Liyanage, "Survey on multi-access edge computing security and privacy," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1078–1124, 2021, doi: 10.1109/COMST.2021.3062546.
- [10] N. Sharghivand, F. Derakhshan, and N. Siasi, "A comprehensive survey on auction mechanism design for cloud/edge resource management and pricing," *IEEE Access*, vol. 9, pp. 126502–126529, 2021, doi: 10.1109/ACCESS.2021.3110914.
- [11] D. Xu, T. Li, Y. Li, X. Su, S. Tarkoma, T. Jiang, J. Crowcroft, and P. Hui, "Edge intelligence: Empowering intelligence to the edge of network," *Proc. IEEE*, vol. 109, no. 11, pp. 1778–1837, Nov. 2021, doi: 10.1109/jproc.2021.3119950.
- [12] B. Sonkoly, J. Czentye, M. Szalay, B. Nemeth, and L. Toka, "Survey on placement methods in the edge and beyond," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2590–2629, Jul. 2021, doi: 10.1109/COMST.2021.3101460.
- [13] Q. Luo, S. Hu, C. Li, G. Li, and W. Shi, "Resource scheduling in edge computing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2131–2165, Aug. 2021, doi: 10.1109/comst.2021.3106401.
- [14] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020, doi: 10.1109/JIOT.2020.2987070.
- [15] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020, doi: 10.1109/ACCESS.2020.2991734.
- [16] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for Internet of Things applications: A survey," *Sensors*, vol. 20, no. 22, pp. 1–52, 2020, doi: 10.3390/s20226441.
- [17] F. Vhora and J. Gandhi, "A comprehensive survey on mobile edge computing: Challenges, tools, applications," in *Proc. 4th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2020, pp. 49–55, doi: 10.1109/ICCMC48092.2020.ICCMC-0009.
- [18] F. A. Salaht, F. Desprez, and A. Lebre, "An overview of service placement problem in fog and edge computing," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–35, May 2021, doi: 10.1145/3391196.
- [19] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT services through software defined networking and edge computing: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1761–1804, 2020, doi: 10.1109/COMST.2020.2997475.

- [20] I. Martinez, A. S. Hafid, and A. Jarray, "Design, resource management, and evaluation of fog computing systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2494–2516, Feb. 2021, doi: [10.1109/JIOT.2020.3022699](https://doi.org/10.1109/JIOT.2020.3022699).
- [21] A. Filali, A. Abouamar, S. Cherkaoui, A. Kobbane, and M. Guizani, "Multi-access edge computing: A survey," *IEEE Access*, vol. 8, pp. 197017–197046, 2020, doi: [10.1109/ACCESS.2020.3034136](https://doi.org/10.1109/ACCESS.2020.3034136).
- [22] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020, doi: [10.1109/ACCESS.2020.3001277](https://doi.org/10.1109/ACCESS.2020.3001277).
- [23] P. Habibi, M. Farhoudi, S. Kazemian, S. Khorsandi, and A. Leon-Garcia, "Fog computing: A comprehensive architectural survey," *IEEE Access*, vol. 8, pp. 69105–69133, 2020, doi: [10.1109/ACCESS.2020.2983253](https://doi.org/10.1109/ACCESS.2020.2983253).
- [24] F. Fang and X. Wu, "A win-win mode: The complementary and coexistence of 5G networks and edge computing," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 3983–4003, Mar. 2021, doi: [10.1109/JIOT.2020.3009821](https://doi.org/10.1109/JIOT.2020.3009821).
- [25] A. Narayanan, A. S. D. Sena, D. Gutierrez-Rojas, D. C. Melgarejo, H. M. Hussain, M. Ullah, S. Bayhan, and P. H. J. Nardelli, "Key advances in pervasive edge computing for industrial Internet of Things in 5G and beyond," *IEEE Access*, vol. 8, pp. 206734–206754, 2020, doi: [10.1109/ACCESS.2020.3037717](https://doi.org/10.1109/ACCESS.2020.3037717).
- [26] J. Qadir, B. Sainz-De-Abajo, A. Khan, B. Garcia-Zapirain, I. De La Torre-Diez, and H. Mahmood, "Towards mobile edge computing: Taxonomy, challenges, applications and future realms," *IEEE Access*, vol. 8, pp. 189129–189162, 2020, doi: [10.1109/ACCESS.2020.3026938](https://doi.org/10.1109/ACCESS.2020.3026938).
- [27] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, "Device-enhanced MEC: Multi-access edge computing (MEC) aided by end device computation and caching: A survey," *IEEE Access*, vol. 7, pp. 166079–166108, 2019, doi: [10.1109/ACCESS.2019.2953172](https://doi.org/10.1109/ACCESS.2019.2953172).
- [28] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog computing for the Internet of Things: A survey," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–41, May 2019, doi: [10.1145/3301443](https://doi.org/10.1145/3301443).
- [29] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms," *ACM Comput. Surv.*, vol. 52, no. 5, pp. 1–37, Sep. 2019. [Online]. Available: <http://dl.acm.org/citation.cfm?doi=3362097.3326066>
- [30] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proc. IEEE*, vol. 107, no. 8, pp. 1–24, Jun. 2019, doi: [10.1109/JPROC.2019.2920341](https://doi.org/10.1109/JPROC.2019.2920341).
- [31] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018, doi: [10.1109/JIOT.2017.2750180](https://doi.org/10.1109/JIOT.2017.2750180).
- [32] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How can edge computing benefit from software-defined networking: A survey, use cases, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017, doi: [10.1109/COMST.2017.2717482](https://doi.org/10.1109/COMST.2017.2717482).
- [33] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017, doi: [10.1109/COMST.2017.2705720](https://doi.org/10.1109/COMST.2017.2705720).
- [34] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018, doi: [10.1109/COMST.2017.2771153](https://doi.org/10.1109/COMST.2017.2771153).
- [35] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017, doi: [10.1109/COMST.2017.2745201](https://doi.org/10.1109/COMST.2017.2745201).
- [36] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, Sep. 2017, doi: [10.1109/COMST.2017.2682318](https://doi.org/10.1109/COMST.2017.2682318).
- [37] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017, doi: [10.1109/ACCESS.2017.2685434](https://doi.org/10.1109/ACCESS.2017.2685434).
- [38] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–37, doi: [10.1109/ISCO.2016.7727082](https://doi.org/10.1109/ISCO.2016.7727082).
- [39] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, "Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges," *J. Netw. Comput. Appl.*, vol. 62, pp. 18–40, Feb. 2016, doi: [10.1016/j.jnca.2015.11.009](https://doi.org/10.1016/j.jnca.2015.11.009).
- [40] J. Sakhdari, S. Izadpanah, B. Zolfaghari, S. H-Mahdizadeh-Zargar, M. Rahati-Quchani, M. Shadi, S. Abrishami, and A. Rasoolzadegan, "Edge computing: A systematic mapping study," 2021, *arXiv:2102.02720*.
- [41] J. Dille, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 50–58, Sep. 2002, doi: [10.1109/MIC.2002.1036038](https://doi.org/10.1109/MIC.2002.1036038).
- [42] M. Pathan, R. Buyya, and A. Vakali, "Content delivery networks: State of the art, insights, and imperatives," in *Content Delivery Networks (Lecture Notes Electrical Engineering)*, vol. 9. 2008, pp. 3–32, doi: [10.1007/978-3-540-77887-5_1](https://doi.org/10.1007/978-3-540-77887-5_1).
- [43] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, Aug. 2001.
- [44] D. Byrne, C. Corrado, and D. E. Sichel, "The rise of cloud computing: Minding your P's, Q's and K's," Tech. Rep., Oct. 2018, doi: [10.3386/W25188](https://doi.org/10.3386/W25188).
- [45] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2011, doi: [10.1109/MPRV.2009.64](https://doi.org/10.1109/MPRV.2009.64).
- [46] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st. Ed., MCC workshop Mobile cloud Comput. (MCC)*, 2012, pp. 13–15, doi: [10.1145/2342509.2342513](https://doi.org/10.1145/2342509.2342513).
- [47] E. M. Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G.-J. Ren, J. Zhu, and J. Farre, "What is a fog node a tutorial on current concepts towards a common definition," 2016, *arXiv:1611.09193*.
- [48] K. Velasquez, D. P. Abreu, M. R. M. Assis, C. Senna, D. F. Aranha, L. F. Bittencourt, N. Laranjeiro, M. Curado, M. Vieira, E. Monteiro, and E. Madeira, "Fog orchestration for the internet of everything: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 9, no. 1, Dec. 2018, doi: [10.1186/s13174-018-0086-3](https://doi.org/10.1186/s13174-018-0086-3).
- [49] S. Khanagha, S. Ansari, S. Paroutis, and L. Oviedo, "Mutualism and the dynamics of new platform creation: A study of Cisco and fog computing," *Strategic Manage. J.*, vol. 43, pp. 1–31, Dec. 2020, doi: [10.1002/smj.3147](https://doi.org/10.1002/smj.3147).
- [50] V. Karagiannis and S. Schulte, "Comparison of alternative architectures in fog computing," in *Proc. IEEE 4th Int. Conf. Fog Edge Comput. (ICFEC)*, May 2020, pp. 19–28, doi: [10.1109/ICFEC50348.2020.00010](https://doi.org/10.1109/ICFEC50348.2020.00010).
- [51] A. S. Alahmad, H. Kahtan, Y. I. Alzoubi, O. Ali, and A. Jaradat, "Mobile cloud computing models security issues: A systematic review," *J. Netw. Comput. Appl.*, vol. 190, Sep. 2021, Art. no. 103152, doi: [10.1016/j.jnca.2021.103152](https://doi.org/10.1016/j.jnca.2021.103152).
- [52] *GROUP—Proof of Concept Framework*, Standard ETSI GS MEC-IEG 005 V1.1.1 (2015-08), ETSI, 2015, pp. 1–14.
- [53] *Edge Computing Takes a Further Leap Forward With Move to Harmonize Standards* | Nokia. Accessed: Apr. 14, 2022. [Online]. Available: <https://www.nokia.com/blog/edge-computing-takes-a-further-leap-forward-with-move-to-harmonize-standards/>
- [54] *Multi-Access Edge Computing (MEC) MEC 5G Integration*, Standard GR MEC 031—V2.1.1, ETSI Report, 2020, pp. 1–47.
- [55] R. K. Barik, A. C. Dubey, A. Tripathi, T. Pratik, S. Sasane, R. K. Lenka, H. Dubey, K. Mankodiya, and V. Kumar, "Mist data: Leveraging mist computing for secure and scalable architecture for smart and connected health," *Proc. Comput. Sci.*, vol. 125, pp. 647–653, Jan. 2018, doi: [10.1016/j.procs.2017.12.083](https://doi.org/10.1016/j.procs.2017.12.083).
- [56] Y. Wang, "Definition and categorization of dew computing," *Open J. Cloud Comput.*, vol. 3, no. 1, pp. 1–7, 2016, doi: [10.19210/1002.3.1.1](https://doi.org/10.19210/1002.3.1.1).
- [57] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: A new paradigm for edge/cloud integration," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 76–83, Nov. 2016.
- [58] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, 2020.
- [59] J. Wei, S. Cao, S. Pan, J. Han, L. Yan, and L. Zhang, "SatEdgeSim: A toolkit for modeling and simulation of performance evaluation in satellite edge computing environments," in *Proc. 12th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2020, pp. 307–313, doi: [10.1109/ICCSN49894.2020.9139057](https://doi.org/10.1109/ICCSN49894.2020.9139057).

- [60] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2020.
- [61] A. Barnawi, M. Alharbi, and M. Chen, "Intelligent search and find system for robotic platform based on smart edge computing service," *IEEE Access*, vol. 8, pp. 108821–108834, 2020, doi: [10.1109/ACCESS.2020.2993727](https://doi.org/10.1109/ACCESS.2020.2993727).
- [62] H. D. Chantre and N. L. Saldanha da Fonseca, "The location problem for the provisioning of protected slices in NFV-based MEC infrastructure," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1505–1514, Jul. 2020, doi: [10.1109/JSAC.2020.2986869](https://doi.org/10.1109/JSAC.2020.2986869).
- [63] A. Kiani. (2018). *From Geographically Dispersed Data Centers towards Hierarchical Edge Computing*. [Online]. Available: <https://digitalcommons.njit.edu/cgi/viewcontent.cgi?article=2425&context=dissertations>
- [64] H. Badri, "Stochastic optimization methods for resource management in edge computing systems," in *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, Aug. 2020, pp. 2805–2814.
- [65] J. Lim and D. Lee, "A load balancing algorithm for mobile devices in edge cloud computing environments," *Electronics*, vol. 9, no. 4, pp. 1–13, 2020, doi: [10.3390/electronics9040686](https://doi.org/10.3390/electronics9040686).
- [66] N. Chen, S. Zhang, J. Wu, Z. Qian, and S. Lu, "Learning scheduling bursty requests in mobile edge computing using DeepLoad," *Comput. Netw.*, vol. 184, Jan. 2021, Art. no. 107655, doi: [10.1016/j.comnet.2020.107655](https://doi.org/10.1016/j.comnet.2020.107655).
- [67] E. C. Iot, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Apr. 2018.
- [68] C. Wang, S. Zhang, Z. Qian, M. Xiao, J. Wu, B. Ye, and S. Lu, "Joint server assignment and resource management for edge-based MAR system," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2378–2391, Oct. 2020, doi: [10.1109/TNET.2020.3012410](https://doi.org/10.1109/TNET.2020.3012410).
- [69] A. Madej, N. Wang, N. Athanasopoulos, R. Ranjan, and B. Varghese, "Priority-based fair scheduling in edge computing," in *Proc. IEEE 4th Int. Conf. Fog Edge Comput. (ICFEC)*, May 2020, pp. 39–48, doi: [10.1109/ICFEC50348.2020.00012](https://doi.org/10.1109/ICFEC50348.2020.00012).
- [70] K. Kolomvatsos and C. Anagnostopoulos, "An intelligent edge-centric queries allocation scheme based on ensemble models," *ACM Trans. Internet Technol.*, vol. 20, no. 4, pp. 1–25, Nov. 2020, doi: [10.1145/3417297](https://doi.org/10.1145/3417297).
- [71] F. Jalali, T. Lynar, O. J. Smith, R. R. Kolluri, C. V. Hardgrove, N. Waywood, and F. Suits, "Dynamic edge fabric Environment: Seamless and automatic switching among resources at the edge of IoT network and cloud," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2019, pp. 77–86, doi: [10.1109/EDGE.2019.00028](https://doi.org/10.1109/EDGE.2019.00028).
- [72] V. P. Kafle and A. H. A. Muktadir, "Intelligent and agile control of edge resources for latency-sensitive IoT services," *IEEE Access*, vol. 8, pp. 207991–208002, 2020, doi: [10.1109/ACCESS.2020.3038439](https://doi.org/10.1109/ACCESS.2020.3038439).
- [73] C. Nguyen, C. Klein, and E. Elmroth, "Multivariate LSTM-based location-aware workload prediction for edge data centers," in *Proc. 19th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2019, pp. 341–350, doi: [10.1109/CCGRID.2019.00048](https://doi.org/10.1109/CCGRID.2019.00048).
- [74] L. Ale, N. Zhang, S. A. King, and J. Guardiola, "Spatio-temporal Bayesian learning for mobile edge computing resource planning in smart cities," 2021, *arXiv:2103.07814*.
- [75] H. Sun, H. Yu, G. Fan, and L. Chen, "QoS-aware task placement with fault-tolerance in the edge-cloud," *IEEE Access*, vol. 8, pp. 77987–78003, 2020, doi: [10.1109/ACCESS.2020.2977089](https://doi.org/10.1109/ACCESS.2020.2977089).
- [76] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between multi-tenancy and interference: A service user allocation game," *IEEE Trans. Services Comput.*, early access, Oct. 6, 2020, doi: [10.1109/tsc.2020.3028760](https://doi.org/10.1109/tsc.2020.3028760).
- [77] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct./Dec. 2015, doi: [10.1109/tcc.2015.2449834](https://doi.org/10.1109/tcc.2015.2449834).
- [78] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekkö, and M. J. Sillanpää, "Edge computing server placement with capacitated location allocation," *J. Parallel Distrib. Comput.*, vol. 153, pp. 130–149, Jul. 2021, doi: [10.1016/j.jpdc.2021.03.007](https://doi.org/10.1016/j.jpdc.2021.03.007).
- [79] Q. Vo and D. A. Tran, "Probabilistic partitioning for edge server assignment with time-varying workload," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–8, doi: [10.1109/ICCCN.2019.8846932](https://doi.org/10.1109/ICCCN.2019.8846932).
- [80] A. S. Gonzalez and C. C. Pastor, "Edge computing node placement in 5G networks: A latency and reliability constrained framework," in *Proc. 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/5th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Jun. 2019, pp. 183–189, doi: [10.1109/CSCloud/EdgeCom.2019.00024](https://doi.org/10.1109/CSCloud/EdgeCom.2019.00024).
- [81] B. Li, Q. He, G. Cui, X. Xia, F. Chen, H. Jin, and Y. Yang, "READ: Robustness-oriented edge application deployment in edge computing environment," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1746–1759, May 2022, doi: [10.1109/TSC.2020.3015316](https://doi.org/10.1109/TSC.2020.3015316).
- [82] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between user coverage and network robustness for edge server placement," *IEEE Trans. Cloud Comput.*, early access, Jul. 10, 2020, doi: [10.1109/TCC.2020.3008440](https://doi.org/10.1109/TCC.2020.3008440).
- [83] Y. Sun, X. Chen, D. Liu, and Y. Tan, "Power-aware virtual machine placement for mobile edge computing," in *Proc. Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jul. 2019, pp. 595–600.
- [84] L. Zhao, J. Liu, Y. Shi, W. Sun, and H. Guo, "Optimal placement of virtual machines in mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6, doi: [10.1109/GLOCOM.2017.8254084](https://doi.org/10.1109/GLOCOM.2017.8254084).
- [85] D. Goncalves, K. Velasquez, M. Curado, L. Bittencourt, and E. Madeira, "Proactive virtual machine migration in fog environments," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 742–745, doi: [10.1109/ISCC.2018.8538655](https://doi.org/10.1109/ISCC.2018.8538655).
- [86] V. B. Souza, M. H. Pereira, L. H. S. Leles, and X. Masip-Bruin, "Enhancing resource availability in vehicular fog computing through smart inter-domain handover," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6, doi: [10.1109/GLOBECOM42002.2020.9322238](https://doi.org/10.1109/GLOBECOM42002.2020.9322238).
- [87] F. Tang, C. Liu, K. Li, Z. Tang, and K. Li, "Task migration optimization for guaranteeing delay deadline with mobility consideration in mobile edge computing," *J. Syst. Archit.*, vol. 112, Jan. 2021, Art. no. 101849, doi: [10.1016/j.sysarc.2020.101849](https://doi.org/10.1016/j.sysarc.2020.101849).
- [88] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, Oct. 2018, doi: [10.1109/JSAC.2018.2869954](https://doi.org/10.1109/JSAC.2018.2869954).
- [89] H. M. Makrani, H. Sayadi, N. Nazari, S. M. P. Dinakarrao, A. Sasan, T. Mohsenin, S. Rafatirad, and H. Homayoun, "Adaptive performance modeling of data-intensive workloads for resource provisioning in virtualized environment," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 5, no. 4, pp. 1–24, Mar. 2021, doi: [10.1145/3442696](https://doi.org/10.1145/3442696).
- [90] Z. Zhou, S. Yu, W. Chen, and X. Chen, "CE-IoT: Cost-effective cloud-edge resource provisioning for heterogeneous IoT applications," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8600–8614, Sep. 2020, doi: [10.1109/JIOT.2020.2994308](https://doi.org/10.1109/JIOT.2020.2994308).
- [91] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "TripRes: Traffic flow prediction driven resource reservation for multimedia IoV with edge computing," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 2, pp. 1–21, May 2021.
- [92] L. Chen and J. Xu, "Budget-constrained edge service provisioning with demand estimation via bandit learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2364–2376, Oct. 2019, doi: [10.1109/JSAC.2019.2933781](https://doi.org/10.1109/JSAC.2019.2933781).
- [93] H. Liang, G. Liu, J. Gao, and M. J. Khan, "Overflow remote warning using improved fuzzy c-means clustering in IoT monitoring system based on multi-access edge computing," *Neural Comput. Appl.*, vol. 32, no. 19, pp. 15399–15410, Oct. 2020, doi: [10.1007/s00521-019-04540-y](https://doi.org/10.1007/s00521-019-04540-y).
- [94] X. Cao, G. Tang, D. Guo, Y. Li, and W. Zhang, "Edge federation: Towards an integrated service provisioning model," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1116–1129, Jun. 2020, doi: [10.1109/TNET.2020.2979361](https://doi.org/10.1109/TNET.2020.2979361).
- [95] R. Chen, L. Li, R. Hou, T. Yang, L. Wang, and M. Pan, "Data-driven optimization for resource provision in non-cooperative edge computing market," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: [10.1109/ICC40277.2020.9149382](https://doi.org/10.1109/ICC40277.2020.9149382).
- [96] S. K. Battula, S. Garg, J. Montgomery, and B. Kang, "An efficient resource monitoring service for fog computing environments," *IEEE Trans. Services Comput.*, vol. 13, no. 4, pp. 709–722, Jul. 2020, doi: [10.1109/TSC.2019.2962682](https://doi.org/10.1109/TSC.2019.2962682).

- [97] C. Jiang, Y. Qiu, H. Gao, T. Fan, K. Li, and J. Wan, "An edge computing platform for intelligent operational monitoring in internet data centers," *IEEE Access*, vol. 7, pp. 133375–133387, 2019, doi: [10.1109/ACCESS.2019.2939614](https://doi.org/10.1109/ACCESS.2019.2939614).
- [98] N. Apolonia, F. Freitag, L. Navarro, S. Girdzijauskas, and V. Vlassov, "Gossip-based service monitoring platform for wireless edge cloud computing," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 789–794, doi: [10.1109/ICNSC.2017.8000191](https://doi.org/10.1109/ICNSC.2017.8000191).
- [99] R. M. Abid, T. Benbrahim, and S. Biaz, "IEEE 802.11s wireless mesh networks for last-mile internet access: An open-source real-world indoor testbed implementation," *Wireless Sensor Netw.*, vol. 2, no. 10, pp. 725–738, 2010, doi: [10.4236/wsn.2010.210088](https://doi.org/10.4236/wsn.2010.210088).
- [100] J. Liu, K. Luo, Z. Zhou, and X. Chen, "ERP: Edge resource pooling for data stream mobile computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4355–4368, Jun. 2019, doi: [10.1109/JIOT.2018.2882588](https://doi.org/10.1109/JIOT.2018.2882588).
- [101] Y. Zheng, W. Xia, L. Jiang, F. Yan, and L. Shen, "Distributed multi-agent cooperative resource sharing algorithm in fog networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6, doi: [10.1109/GLOBECOM42002.2020.9322581](https://doi.org/10.1109/GLOBECOM42002.2020.9322581).
- [102] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1227–1240, Jun. 2020, doi: [10.1109/TNET.2020.2979807](https://doi.org/10.1109/TNET.2020.2979807).
- [103] C. Tang, S. Xia, Q. Li, W. Chen, and W. Fang, "Resource pooling in vehicular fog computing," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–14, Dec. 2021, doi: [10.1186/s13677-021-00233-x](https://doi.org/10.1186/s13677-021-00233-x).
- [104] V. Karagiannis, N. Desai, S. Schulte, and S. Punnekkat, "Addressing the node discovery problem in fog computing," *OpenAccess Informat.*, vol. 80, no. 5, pp. 1–5, 2020, doi: [10.4230/OASfcs.Fog-IoT.2020.5](https://doi.org/10.4230/OASfcs.Fog-IoT.2020.5).
- [105] U. Polit, "Mobility-aware mechanisms for fog node discovery and selection," Ph.D. thesis, 2020.
- [106] I. Murturi and S. Dustdar, "A decentralized approach for resource discovery using metadata replication in edge networks," *IEEE Trans. Services Comput.*, early access, May 20, 2021, doi: [10.1109/TSC.2021.3082305](https://doi.org/10.1109/TSC.2021.3082305).
- [107] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," *IEEE Access*, vol. 7, pp. 131543–131558, 2019, doi: [10.1109/ACCESS.2019.2938660](https://doi.org/10.1109/ACCESS.2019.2938660).
- [108] Z. Liao, J. Peng, B. Xiong, and J. Huang, "Adaptive offloading in mobile-edge computing for ultra-dense cellular networks based on genetic algorithm," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–16, Dec. 2021, doi: [10.1186/s13677-021-00232-y](https://doi.org/10.1186/s13677-021-00232-y).
- [109] X. Wang, J. Ye, and J. C. S. Lui, "Joint D2D collaboration and task offloading for edge computing: A mean field graph approach," in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQOS)*, Jun. 2021, pp. 1–10, doi: [10.1109/IWQOS52092.2021.9521271](https://doi.org/10.1109/IWQOS52092.2021.9521271).
- [110] I. Kovacevic, E. Harjula, S. Glisic, B. Lorenzo, and M. Ylianttila, "Cloud and edge computation offloading for latency limited services," *IEEE Access*, vol. 9, pp. 55764–55776, 2021, doi: [10.1109/ACCESS.2021.3071848](https://doi.org/10.1109/ACCESS.2021.3071848).
- [111] J. Long, Y. Luo, X. Zhu, E. Luo, and M. Huang, "Computation offloading through mobile vehicles in IoT-edge-cloud network," *EURASIP J. Wireless Commun. Netw.*, vol. 2020, no. 1, pp. 1–21, Dec. 2020, doi: [10.1186/s13638-020-01848-5](https://doi.org/10.1186/s13638-020-01848-5).
- [112] M. D. Hossain, T. Sultana, M. A. Hossain, M. I. Hossain, L. N. T. Huynh, J. Park, and E.-N. Huh, "Fuzzy decision-based efficient task offloading management scheme in multi-tier MEC-enabled networks," *Sensors*, vol. 21, no. 4, pp. 1–26, 2021, doi: [10.3390/s21041484](https://doi.org/10.3390/s21041484).
- [113] E. Kilcioglu, H. Mirghasemi, I. Stupia, and L. Vandendorpe, "An energy-efficient fine-grained deep neural network partitioning scheme for wireless collaborative fog computing," *IEEE Access*, vol. 9, pp. 79611–79627, 2021, doi: [10.1109/ACCESS.2021.3084689](https://doi.org/10.1109/ACCESS.2021.3084689).
- [114] F. Taşyaran, B. Demireller, K. Kaya, and B. Uçar, "Streaming hypergraph partitioning algorithms on limited memory environments," 2021, *arXiv:2103.05394*.
- [115] M. Breitbach, D. Schafer, J. Edinger, and C. Becker, "Context-aware data and task placement in edge computing environments," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2019, pp. 1–10.
- [116] H. Gao, X. Wang, X. Ma, W. Wei, and S. Mumtaz, "Com-DDPG: A multiagent reinforcement learning-based offloading strategy for mobile edge computing," 2020, *arXiv:2012.05105*.
- [117] R. G. Pacheco, R. S. Couto, and O. Simeone, "Calibration-aided edge inference offloading via adaptive model partitioning of deep neural networks," 2020, *arXiv:2010.16335*.
- [118] T. A. Putra and J.-S. Leu, "Multilevel neural network for reducing expected inference time," *IEEE Access*, vol. 7, pp. 174129–174138, 2019, doi: [10.1109/ACCESS.2019.2952577](https://doi.org/10.1109/ACCESS.2019.2952577).
- [119] X. Tian, J. Zhu, T. Xu, and Y. Li, "Mobility-included DNN partition offloading from mobile devices to edge clouds," *Sensors*, vol. 21, no. 1, pp. 1–16, 2021, doi: [10.3390/s21010229](https://doi.org/10.3390/s21010229).
- [120] G. Tefera, K. She, M. Chen, and A. Ahmed, "Congestion-aware adaptive decentralised computation offloading and caching for multi-access edge computing networks," *IET Commun.*, vol. 14, no. 19, pp. 3410–3419, Dec. 2020, doi: [10.1049/iet-com.2020.0630](https://doi.org/10.1049/iet-com.2020.0630).
- [121] W. Bai, Z. Ma, Y. Han, M. Wu, Z. Zhao, M. Li, and C. Wang, "Joint optimization of computation offloading, data compression, energy harvesting, and application scenarios in fog computing," *IEEE Access*, vol. 9, pp. 45462–45473, 2021, doi: [10.1109/ACCESS.2021.3067702](https://doi.org/10.1109/ACCESS.2021.3067702).
- [122] C. S. Yang, R. Pedarsani, and A. S. Avestimehr, "Edge computing in the dark: Leveraging contextual-combinatorial bandit and coded computing," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1–13, Jun. 2021, doi: [10.1109/TNET.2021.3058685](https://doi.org/10.1109/TNET.2021.3058685).
- [123] J. Zhang, X. Zhang, and W. Zhang, "Microseismic search engine," in *Proc. Int. Expo. 83rd Annu. Meeting SEG Expand. Geophys. Frontiers*, 2013, pp. 2140–2144, doi: [10.1190/segam2013-1277.1](https://doi.org/10.1190/segam2013-1277.1).
- [124] M. Yang, H. Zhu, H. Qian, Y. Koucheryayv, K. Samouylov, and H. Wang, "Peer offloading with delayed feedback in fog networks," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 1–12, Mar. 2021, doi: [10.1109/JIOT.2021.3067919](https://doi.org/10.1109/JIOT.2021.3067919).
- [125] J. Almutairi and M. Aldossary, "Modeling and analyzing offloading strategies of IoT applications over edge computing and joint clouds," *Symmetry*, vol. 13, no. 3, pp. 1–19, Mar. 2021, doi: [10.3390/sym13030402](https://doi.org/10.3390/sym13030402).
- [126] S. L. Li, J. B. Du, D. S. Zhai, X. L. Chu, and F. R. Yu, "Task offloading, load balancing, and resource allocation in MEC networks," *IET Commun.*, vol. 14, no. 9, pp. 1451–1458, Jun. 2020, doi: [10.1049/iet-com.2018.6122](https://doi.org/10.1049/iet-com.2018.6122).
- [127] S. Bi, L. Huang, H. Wang, and Y.-J. Angela Zhang, "Lyapunov-guided deep reinforcement learning for stable online computation offloading in mobile-edge computing networks," 2020, *arXiv:2010.01370*.
- [128] D. Sun, H. He, H. Yan, S. Gao, X. Liu, and X. Zheng, "LR-stream: Using latency and resource aware scheduling to improve latency and throughput for streaming applications," *Future Gener. Comput. Syst.*, vol. 114, pp. 243–258, Jan. 2021, doi: [10.1016/j.future.2020.08.003](https://doi.org/10.1016/j.future.2020.08.003).
- [129] X. Gao, X. Huang, S. Bian, Z. Shao, and Y. Yang, "PORA: Predictive offloading and resource allocation in dynamic fog computing systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: [10.1109/ICC.2019.8762031](https://doi.org/10.1109/ICC.2019.8762031).
- [130] X. Xia, F. Chen, Q. He, G. Cui, P. Lai, M. Abdelrazek, J. Grundy, and H. Jin, "Graph-based optimal data caching in edge computing," in *Service-Oriented Computing (Lecture Notes in Computer Science)*, vol. 11895, Oct. 2019, pp. 477–493, doi: [10.1007/978-3-030-33702-5_37](https://doi.org/10.1007/978-3-030-33702-5_37).
- [131] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "A novel mobile edge network architecture with joint caching-delivering and horizontal cooperation," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 19–31, Jan. 2021, doi: [10.1109/TMC.2019.2938510](https://doi.org/10.1109/TMC.2019.2938510).
- [132] Y. Sun, Z. Chen, M. Tao, and H. Liu, "Bandwidth gain from mobile edge computing and caching in wireless multicast systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3992–4007, Jun. 2020, doi: [10.1109/TWC.2020.2979147](https://doi.org/10.1109/TWC.2020.2979147).
- [133] A. Asheralieva and D. Niyato, "Combining contract theory and Lyapunov optimization for content sharing with edge caching and device-to-device communications," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1213–1226, Jun. 2020, doi: [10.1109/TNET.2020.2978117](https://doi.org/10.1109/TNET.2020.2978117).
- [134] Y.-T. Lin, C.-C. Yen, and J.-S. Wang, "Video popularity prediction: An autoencoder approach with clustering," *IEEE Access*, vol. 8, pp. 129285–129299, 2020, doi: [10.1109/ACCESS.2020.3009253](https://doi.org/10.1109/ACCESS.2020.3009253).
- [135] Q. Fan, X. Li, J. Li, Q. He, K. Wang, and J. Wen, "PA-cache: Evolving learning-based popularity-aware content caching in edge networks," 2020, *arXiv:2002.08805*.
- [136] X. Gao, X. Huang, Y. Tang, Z. Shao, and Y. Yang, "History-aware online cache placement in fog-assisted IoT systems: An integration of learning and control," *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14683–14704, Oct. 2021, doi: [10.1109/JIOT.2021.3072115](https://doi.org/10.1109/JIOT.2021.3072115).

- [137] O. K. Kazi, S. A. Memon, E. Saba, Z. Ali, and F. Naz, "Infras-structure sharing and remedies in next generation cellular networks," *Int. J. Comput. Sci. Netw. Secur.*, vol. 20, no. 12, p. 185, 2020, doi: [10.22937/IJCSNS.2020.20.12.20](https://doi.org/10.22937/IJCSNS.2020.20.12.20).
- [138] Z. Xu, L. Zhou, S. Chi-Kin Chau, W. Liang, Q. Xia, and P. Zhou, "Collaborate or separate? Distributed service caching in mobile edge clouds," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 2066–2075, doi: [10.1109/INFOCOM41043.2020.9155365](https://doi.org/10.1109/INFOCOM41043.2020.9155365).
- [139] N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for Internet of Things," *Proc. IEEE 9th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2014, pp. 21–24.
- [140] W. Liu, K. Nakauchi, and Y. Shoji, "A neighbor-based probabilistic broadcast protocol for data dissemination in mobile IoT networks," *IEEE Access*, vol. 6, pp. 12260–12268, 2018, doi: [10.1109/ACCESS.2018.2808356](https://doi.org/10.1109/ACCESS.2018.2808356).
- [141] A. Ullah, S. Yaqoob, M. Imran, and H. Ning, "Emergency message dissemination schemes based on congestion avoidance in VANET and vehicular FoG computing," *IEEE Access*, vol. 7, pp. 1570–1585, 2019, doi: [10.1109/ACCESS.2018.2887075](https://doi.org/10.1109/ACCESS.2018.2887075).
- [142] S. Luo, T. Ma, W. Shan, P. Fan, H. Xing, and H. Yu, "Efficient multi-source data delivery in edge cloud with rateless parallel push," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10495–10510, Oct. 2020, doi: [10.1109/JIOT.2020.2996800](https://doi.org/10.1109/JIOT.2020.2996800).
- [143] C.-M. Huang, S.-Y. Lin, and Z.-Y. Wu, "The K-hop-limited V2V2I VANET data offloading using the mobile edge computing (MEC) mechanism," *Veh. Commun.*, vol. 26, Dec. 2020, Art. no. 100268, doi: [10.1016/j.vehcom.2020.100268](https://doi.org/10.1016/j.vehcom.2020.100268).
- [144] Comcast: Pandemic Drove Peak Internet Traffic Up 32% in 2020 | VentureBeat. Accessed: Mar. 31, 2022. [Online]. Available: <https://venturebeat.com/2021/03/02/comcast-peak-internet-traffic-rose-32-in-pandemic-in-2020/>
- [145] X. Wang, Z. Zhou, P. Han, T. Meng, G. Sun, and J. Zhai, "Edge-stream: A stream processing approach for distributed applications on a hierarchical edge-computing system," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Nov. 2020, pp. 14–27, doi: [10.1109/SEC50012.2020.00009](https://doi.org/10.1109/SEC50012.2020.00009).
- [146] C. Qiao, J. Wang, and Y. Liu, "Beyond QoE: Diversity adaption in video streaming at the edge," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 317–326, doi: [10.1109/ICDCS.2019.00039](https://doi.org/10.1109/ICDCS.2019.00039).
- [147] F. Fu, Y. Kang, Z. Zhang, F. R. Yu, and T. Wu, "Soft actor–critic DRL for live transcoding and streaming in vehicular fog-computing-enabled IoV," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1308–1321, Feb. 2021, doi: [10.1109/JIOT.2020.3003398](https://doi.org/10.1109/JIOT.2020.3003398).
- [148] S. Dube, W. Y. Wan, and H. Nugroho, "A novel approach of IoT stream sampling and model update on the IoT edge device for class incremental learning in an edge-cloud system," *IEEE Access*, vol. 9, pp. 29180–29199, 2021, doi: [10.1109/ACCESS.2021.3059251](https://doi.org/10.1109/ACCESS.2021.3059251).
- [149] J. F. Lopes, E. J. Santana, V. G. T. da Costa, B. B. Zarpelao, and S. Barbon, "Evaluating the four-way performance trade-off for data stream classification in edge computing," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 1013–1025, Jun. 2020, doi: [10.1109/TNSM.2020.2983921](https://doi.org/10.1109/TNSM.2020.2983921).
- [150] O. Jodelka, C. Anagnostopoulos, and K. Kolomvatsos, "Adaptive novelty detection over contextual data streams at the edge using one-class classification," in *Proc. 12th Int. Conf. Inf. Commun. Syst. (ICICS)*, May 2021, pp. 213–219.
- [151] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, and M. Pan, "Energy-efficient proactive caching for adaptive video streaming via data-driven optimization," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5549–5561, Jun. 2020, doi: [10.1109/JIOT.2020.2981250](https://doi.org/10.1109/JIOT.2020.2981250).
- [152] A. Erfanian, H. Amirpour, F. Tashtarian, C. Timmerer, and H. Hellwagner, "LwTE: Light-weight transcoding at the edge," *IEEE Access*, vol. 9, pp. 112276–112289, 2021, doi: [10.1109/ACCESS.2021.3102633](https://doi.org/10.1109/ACCESS.2021.3102633).
- [153] Y. Zhu, Q. He, J. Liu, B. Li, and Y. Hu, "When crowd meets big video data: Cloud-edge collaborative transcoding for personal livecast," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 42–53, Jan. 2020, doi: [10.1109/TNSE.2018.2873311](https://doi.org/10.1109/TNSE.2018.2873311).
- [154] Apache Hadoop. Accessed: Mar. 31, 2022. [Online]. Available: <https://hadoop.apache.org/>
- [155] Apache Spark™—Unified Analytics Engine for Big Data. Accessed: Jul. 1, 2021. [Online]. Available: <https://spark.apache.org/>
- [156] A. M. Ghosh and K. Grolinger, "Edge-cloud computing for Internet of Things data analytics: Embedding intelligence in the edge with deep learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2191–2200, Mar. 2021, doi: [10.1109/TII.2020.3008711](https://doi.org/10.1109/TII.2020.3008711).
- [157] L. Valerio, A. Passarella, and M. Conti, "Optimising cost vs accuracy of decentralised analytics in fog computing environments," *IEEE Trans. Netw. Sci. Eng.*, early access, Aug. 4, 2021, doi: [10.1109/TNSE.2021.3101986](https://doi.org/10.1109/TNSE.2021.3101986).
- [158] Y. Wang, W. Wang, D. Liu, X. Jin, J. Jiang, and K. Chen, "Enabling edge-cloud video analytics for robotics applications," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2021, pp. 1–10, doi: [10.1109/INFOCOM42981.2021.9488801](https://doi.org/10.1109/INFOCOM42981.2021.9488801).
- [159] C. Wang, S. Zhang, Y. Chen, Z. Qian, J. Wu, and M. Xiao, "Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 257–266, doi: [10.1109/INFOCOM41043.2020.9155524](https://doi.org/10.1109/INFOCOM41043.2020.9155524).
- [160] T. Buddhika, M. Malensek, S. Pallickara, and S. L. Pallickara, "Living on the edge: Data transmission, storage, and analytics in continuous sensing environments," *ACM Trans. Internet Things*, vol. 2, no. 3, pp. 1–31, Aug. 2021.
- [161] H. Jin, L. Jia, and Z. Zhou, "Boosting edge intelligence with collaborative cross-edge analytics," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2444–2458, Feb. 2021, doi: [10.1109/JIOT.2020.3034891](https://doi.org/10.1109/JIOT.2020.3034891).
- [162] X. Xia, F. Chen, Q. He, J. C. Grundy, M. Abdelrazek, and H. Jin, "Cost-effective app data distribution in edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 1, pp. 31–44, Jan. 2021, doi: [10.1109/TPDS.2020.3010521](https://doi.org/10.1109/TPDS.2020.3010521).
- [163] M. Linaje, J. Berrocal, and A. Galan-Benitez, "Mist and edge storage: Fair storage distribution in sensor networks," *IEEE Access*, vol. 7, pp. 123860–123876, 2019, doi: [10.1109/ACCESS.2019.2938443](https://doi.org/10.1109/ACCESS.2019.2938443).
- [164] D. M. A. D. Silva, G. Asaamoning, H. Orrillo, R. C. Sofia, and P. M. Mendes, "An analysis of fog computing data placement algorithms," in *Proc. 16th EAI Int. Conf. Mobile Ubiquitous Syst., Comput., Netw. Services*, Nov. 2019, pp. 527–534, doi: [10.1145/3360774.3368201](https://doi.org/10.1145/3360774.3368201).
- [165] J. Xie, C. Qian, D. Guo, X. Li, S. Shi, and H. Chen, "Efficient data placement and retrieval services in edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1029–1039, doi: [10.1109/ICDCS.2019.00106](https://doi.org/10.1109/ICDCS.2019.00106).
- [166] D. Guo, J. Xie, X. Shi, H. Cai, C. Qian, and H. Chen, "HDS: A fast hybrid data location service for hierarchical mobile edge computing," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 1–14, Jun. 2021, doi: [10.1109/TNET.2021.3058401](https://doi.org/10.1109/TNET.2021.3058401).
- [167] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008, doi: [10.1145/1355734.1355746](https://doi.org/10.1145/1355734.1355746).
- [168] J. L. Herrera, J. Galan-Jimenez, J. Berrocal, and J. M. Murillo, "Optimizing the response time in SDN-fog environments for time-strict IoT applications," *IEEE Internet Things J.*, vol. 8, no. 23, pp. 17172–17185, Dec. 2021, doi: [10.1109/JIOT.2021.3077992](https://doi.org/10.1109/JIOT.2021.3077992).
- [169] D. E. Sarmiento, A. Lebre, L. Nussbaum, and A. Chari, "Decentralized SDN control plane for a distributed cloud-edge infrastructure: A survey," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 256–281, Jan. 2021, doi: [10.1109/COMST.2021.3050297](https://doi.org/10.1109/COMST.2021.3050297).
- [170] J. Liu, G. Shou, Y. Liu, Y. Hu, and Z. Guo, "Performance evaluation of integrated multi-access edge computing and fiber-wireless access networks," *IEEE Access*, vol. 6, pp. 30269–30279, 2018, doi: [10.1109/ACCESS.2018.2833619](https://doi.org/10.1109/ACCESS.2018.2833619).
- [171] S. D. A. Shah, M. A. Gregory, S. Li, and R. D. R. Fontes, "SDN enhanced multi-access edge computing (MEC) for E2E mobility and QoS management," *IEEE Access*, vol. 8, pp. 77459–77469, 2020, doi: [10.1109/ACCESS.2020.2990292](https://doi.org/10.1109/ACCESS.2020.2990292).
- [172] P. Zhao, W. Yu, X. Yang, D. Meng, L. Wang, S. Yang, and J. Lin, "Context-aware multi-criteria handover at the software defined network edge for service differentiation in next generation wireless networks," *IEEE Trans. Services Comput.*, early access, Oct. 14, 2020, doi: [10.1109/tsc.2020.3031181](https://doi.org/10.1109/tsc.2020.3031181).
- [173] S. S. C. G., V. Chamola, C.-K. Tham, G. S., and N. Ansari, "An optimal delay aware task assignment scheme for wireless SDN networked edge cloudlets," *Future Gener. Comput. Syst.*, vol. 102, pp. 862–875, Jan. 2020, doi: [10.1016/j.future.2019.09.003](https://doi.org/10.1016/j.future.2019.09.003).
- [174] P. Thorat and N. K. Dubey, "SDN-based machine learning powered alarm manager for mitigating the traffic spikes at the IoT gateways," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2020, pp. 1–6, doi: [10.1109/CONECCT50063.2020.9198356](https://doi.org/10.1109/CONECCT50063.2020.9198356).

- [175] R. Farahani, F. Tashtarian, H. Amirpour, C. Timmerer, M. Ghanbari, and H. Hellwagner, "CSDN: CDN-aware QoE optimization in SDN-assisted HTTP adaptive video streaming," in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 525–532, doi: [10.1109/lcn52139.2021.9524970](https://doi.org/10.1109/lcn52139.2021.9524970).
- [176] C. Zhang, M. Dong, and K. Ota, "Deploying SDN control in internet of UAVs: Q-learning-based edge scheduling," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 526–537, Mar. 2021, doi: [10.1109/TNSM.2021.3059159](https://doi.org/10.1109/TNSM.2021.3059159).
- [177] R. Shinkuma, Y. Yamada, T. Sato, and E. Oki, "Flow control in SDN-edge-cloud cooperation system with machine learning," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 1304–1309, doi: [10.1109/ICDCS47774.2020.00169](https://doi.org/10.1109/ICDCS47774.2020.00169).
- [178] P. Wang, Z. Wang, F. Ye, and X. Chen, "ByteSGAN: A semi-supervised generative adversarial network for encrypted traffic classification of SDN edge gateway in green communication network," 2021, *arXiv:2103.05250*.
- [179] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-assisted congestion control mechanism for 5G network using software-defined networking," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–5, doi: [10.1109/ISWCS.2018.8491233](https://doi.org/10.1109/ISWCS.2018.8491233).
- [180] A. Akbar, M. Ibrar, M. A. Jan, A. K. Bashir, and L. Wang, "SDN-enabled adaptive and reliable communication in IoT-fog environment using machine learning and multiobjective optimization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3057–3065, Nov. 2021, doi: [10.1109/JIOT.2020.3038768](https://doi.org/10.1109/JIOT.2020.3038768).
- [181] A. Huang, N. Nikaein, T. Stenbock, A. Ksentini, and C. Bonnet, "Low latency MEC framework for SDN-based LTE/LTE-A networks," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6, doi: [10.1109/ICC.2017.7996359](https://doi.org/10.1109/ICC.2017.7996359).
- [182] D. Zhang, F. R. Yu, R. Yang, and L. Zhu, "Software-defined vehicular networks with trust management: A deep reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1–15, Feb. 2020, doi: [10.1109/tits.2020.3025684](https://doi.org/10.1109/tits.2020.3025684).
- [183] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 2016, doi: [10.1109/COMST.2015.2477041](https://doi.org/10.1109/COMST.2015.2477041).
- [184] W.-S. Kim, S.-H. Chung, and C.-W. Ahn, "Joint resource allocation based on traffic flow virtualization for edge computing," *IEEE Access*, vol. 9, pp. 57989–58008, 2021, doi: [10.1109/ACCESS.2021.3072164](https://doi.org/10.1109/ACCESS.2021.3072164).
- [185] T.-M. Pham and T.-T.-L. Nguyen, "Optimization of resource management for NFV-enabled IoT systems in edge cloud computing," *IEEE Access*, vol. 8, pp. 178217–178229, 2020, doi: [10.1109/ACCESS.2020.3026711](https://doi.org/10.1109/ACCESS.2020.3026711).
- [186] Z. Xu, W. Gong, Q. Xia, W. Liang, O. F. Rana, and G. Wu, "NFV-enabled IoT service provisioning in mobile edge clouds," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1892–1906, May 2021, doi: [10.1109/TMC.2020.2972530](https://doi.org/10.1109/TMC.2020.2972530).
- [187] T. H. L. Dinh, M. Kaneko, E. H. Fukuda, and L. Boukhatem, "Energy efficient resource allocation optimization in fog radio access networks with outdated channel knowledge," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 146–159, Mar. 2021, doi: [10.1109/TGCN.2020.3034638](https://doi.org/10.1109/TGCN.2020.3034638).
- [188] M. Ke, Z. Gao, and Y. Wu, "Compressive massive access for Internet of Things: Cloud computing or fog computing?" in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6, doi: [10.1109/ICC40277.2020.9148994](https://doi.org/10.1109/ICC40277.2020.9148994).
- [189] C. W. Zaw, N. H. Tran, Z. Han, and C. S. Hong, "Radio and computing resource allocation in co-located edge computing: A generalized Nash equilibrium model," *IEEE Trans. Mobile Comput.*, early access, Oct. 15, 2021, doi: [10.1109/TMC.2021.3120520](https://doi.org/10.1109/TMC.2021.3120520).
- [190] Z. Shao, M. A. Islam, and S. Ren, "Heat behind the meter: A hidden threat of thermal attacks in edge colocation data centers," in *Proc. Int. Symp. High-Performance Comput. Archit.*, Feb. 2021, pp. 318–331, doi: [10.1109/HPCA51647.2021.00035](https://doi.org/10.1109/HPCA51647.2021.00035).
- [191] S. Forti, G.-L. Ferrari, and A. Brogi, "Secure cloud-edge deployments, with trust," *Future Gener. Comput. Syst.*, vol. 102, pp. 775–788, Jan. 2020, doi: [10.1016/j.future.2019.08.020](https://doi.org/10.1016/j.future.2019.08.020).
- [192] A. Qasem, P. Shirani, M. Debbabi, L. Wang, B. Lebel, and B. L. Agba, "Automatic vulnerability detection in embedded devices and firmware: Survey and layered taxonomies," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–42, Mar. 2022, doi: [10.1145/3432893](https://doi.org/10.1145/3432893).
- [193] Q. He, C. Wang, G. Cui, B. Li, R. Zhou, Q. Zhou, Y. Xiang, H. Jin, and Y. Yang, "A game-theoretical approach for mitigating edge DDoS attack," *IEEE Trans. Dependable Secur. Comput.*, early access, Jan. 29, 2021, doi: [10.1109/TDSC.2021.3055559](https://doi.org/10.1109/TDSC.2021.3055559).
- [194] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3930–3944, Jul. 2021, doi: [10.1109/JIOT.2021.3100755](https://doi.org/10.1109/JIOT.2021.3100755).
- [195] S. K. Nukavarapu and T. Nadeem, "Securing edge-based IoT networks with semi-supervised GANs," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops Affiliated Events (PerCom Workshops)*, Mar. 2021, pp. 579–584, doi: [10.1109/PerComWorkshops51409.2021.9431112](https://doi.org/10.1109/PerComWorkshops51409.2021.9431112).
- [196] V. Christopher, T. Aathman, K. Mahendrakumar, R. Nawaratne, D. De Silva, V. Nanayakkara, and D. Alahakoon, "Minority resampling boosted unsupervised learning with hyperdimensional computing for threat detection at the edge of Internet of Things," *IEEE Access*, vol. 9, pp. 126646–126657, 2021, doi: [10.1109/access.2021.3111053](https://doi.org/10.1109/access.2021.3111053).
- [197] R. Smith, D. Palin, P. P. Ioulianou, V. G. Vassilakis, and S. F. Shahandashti, "Battery draining attacks against edge computing nodes in IoT networks," *Cyber-Phys. Syst.*, vol. 6, no. 2, pp. 96–116, Apr. 2020, doi: [10.1080/23335777.2020.1716268](https://doi.org/10.1080/23335777.2020.1716268).
- [198] I. G. Lee, K. Go, and J. H. Lee, "Battery draining attack and defense against power saving wireless LAN devices," *Sensors*, vol. 20, no. 7, pp. 1–13, 2020, doi: [10.3390/s20072043](https://doi.org/10.3390/s20072043).
- [199] D. Benhaddou and A. Al-Fuqaha, *Wireless Sensor and Mobile Ad-Hoc Networks Vehicular and Space Applications*. New York, NY, USA: Springer, 2015.
- [200] N. N. Tran, H. R. Pota, Q. N. Tran, and J. Hu, "Designing constraint-based false data-injection attacks against the unbalanced distribution smart grids," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9422–9435, Jun. 2021, doi: [10.1109/JIOT.2021.3056649](https://doi.org/10.1109/JIOT.2021.3056649).
- [201] F. Pan, H. Wen, X. Gao, H. Pu, and Z. Pang, "Clone detection based on BPNN and physical layer reputation for industrial wireless CPS," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3693–3702, May 2021, doi: [10.1109/TII.2020.3028120](https://doi.org/10.1109/TII.2020.3028120).
- [202] Y. Keshkarjahromi, R. Bitar, V. Dasari, S. El Rouayheb, and H. Seferoglu, "Secure coded cooperative computation at the heterogeneous edge against byzantine attacks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6, doi: [10.1109/GLOBECOM38437.2019.9013340](https://doi.org/10.1109/GLOBECOM38437.2019.9013340).
- [203] K. Matsui and H. Nishi, "Error correction method considering fog and edge computing environment," in *Proc. IEEE Int. Conf. Ind. Cyber Phys. Syst. (ICPS)*, May 2019, pp. 517–521, doi: [10.1109/ICPHYS.2019.8780317](https://doi.org/10.1109/ICPHYS.2019.8780317).
- [204] W. Tong, B. Jiang, F. Xu, Q. Li, and S. Zhong, "Privacy-preserving data integrity verification in mobile edge computing," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1007–1018, doi: [10.1109/ICDCS.2019.00104](https://doi.org/10.1109/ICDCS.2019.00104).
- [205] B. Li, Q. He, F. Chen, H. Jin, Y. Xiang, and Y. Yang, "Inspecting edge data integrity with aggregated signature in distributed edge computing environment," *IEEE Trans. Cloud Comput.*, early access, Feb. 16, 2021, doi: [10.1109/TCC.2021.3059448](https://doi.org/10.1109/TCC.2021.3059448).
- [206] A. Alazeb and B. Panda, "Maintaining data integrity in fog computing based critical infrastructure systems," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2019, pp. 40–47, doi: [10.1109/CSCI49370.2019.00014](https://doi.org/10.1109/CSCI49370.2019.00014).
- [207] A. Al-Mamun and D. Zhao, "Trustworthy edge computing through blockchains," 2020, *arXiv:2005.07741*.
- [208] H. Cui, X. Yi, and S. Nepal, "Achieving scalable access control over encrypted data for edge computing networks," *IEEE Access*, vol. 6, pp. 30049–30059, 2018, doi: [10.1109/ACCESS.2018.2844373](https://doi.org/10.1109/ACCESS.2018.2844373).
- [209] M. H. Ibrahim, "Octopus: An edge-fog mutual authentication scheme," *Int. J. Netw. Secur.*, vol. 18, no. 6, pp. 1089–1101, Nov. 2016.
- [210] M. Wazid, A. K. Das, S. Shetty, J. J. P. C. Rodrigues, and Y. H. Park, "LDKAM-ElIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment," *Sensors*, vol. 19, no. 24, pp. 1–21, 2019, doi: [10.3390/s19245539](https://doi.org/10.3390/s19245539).
- [211] S. Chen, Z. Pang, H. Wen, K. Yu, T. Zhang, and Y. Lu, "Automated labeling and learning for physical layer authentication against clone node and sybil attacks in industrial wireless edge networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 3, pp. 2041–2051, Mar. 2021, doi: [10.1109/TII.2020.2963962](https://doi.org/10.1109/TII.2020.2963962).

- [212] L. Ma, Q. Pei, L. Zhou, H. Zhu, L. Wang, and Y. Ji, "Federated data cleaning: Collaborative and privacy-preserving data cleaning for edge intelligence," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6757–6770, Apr. 2021, doi: [10.1109/JIOT.2020.3027980](https://doi.org/10.1109/JIOT.2020.3027980).
- [213] Y. Wang and T. Nakachi, "Secure face recognition in edge and cloud networks: From the ensemble learning perspective," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2393–2397, doi: [10.1109/ICASSP40776.2020.9052992](https://doi.org/10.1109/ICASSP40776.2020.9052992).
- [214] J. Cui, L. Wei, H. Zhong, J. Zhang, Y. Xu, and L. Liu, "Edge computing in VANETs—An efficient and privacy-preserving cooperative downloading scheme," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1191–1204, Jun. 2020, doi: [10.1109/JSAC.2020.2986617](https://doi.org/10.1109/JSAC.2020.2986617).
- [215] H. Zhong, Y. Zhou, Q. Zhang, Y. Xu, and J. Cui, "An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare," *Future Gener. Comput. Syst.*, vol. 115, pp. 486–496, Feb. 2021, doi: [10.1016/j.future.2020.09.021](https://doi.org/10.1016/j.future.2020.09.021).
- [216] L. Jiang, R. Tan, X. Lou, and G. Lin, "On lightweight privacy-preserving collaborative learning for internet-of-things objects," in *Proc. Int. Conf. Internet Things Design Implementation*, vol. 2, no. 2, 2019, pp. 70–81, doi: [10.1145/3302505.3310070](https://doi.org/10.1145/3302505.3310070).
- [217] A. Gottipati, A. Stewart, J. Song, and Q. Chen, "FedRAN: Federated mobile edge computing with differential privacy," in *Proc. 4th FlexNets Workshop FlexNets Netw., Artif. Intell. Support. Netw. Flex. Agil.*, 2021, pp. 14–19, 2021, doi: [10.1145/3472735.3473392](https://doi.org/10.1145/3472735.3473392).
- [218] Y. S. Can and C. Ersoy, "Privacy-preserving federated deep learning for wearable IoT-based biomedical monitoring," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–17, Feb. 2021, doi: [10.1145/3428152](https://doi.org/10.1145/3428152).
- [219] Z. Du, C. Wu, T. Yoshinaga, K. L. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Comput. Graph. Appl.*, vol. 1, pp. 1–16, 2020, doi: [10.1109/OJCS.2020.2992630](https://doi.org/10.1109/OJCS.2020.2992630).
- [220] Q. Xu, Z. Su, K. Zhang, and P. Li, "Intelligent cache pollution attacks detection for edge computing enabled mobile social networks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 241–252, Jun. 2020.
- [221] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Secur. Priv.*, May 2019, pp. 739–753, doi: [10.1109/SP.2019.00065](https://doi.org/10.1109/SP.2019.00065).
- [222] Q. Zhang, K. Wang, W. Zhang, and J. Hu, "Attacking black-box image classifiers with particle swarm optimization," *IEEE Access*, vol. 7, pp. 158051–158063, 2019, doi: [10.1109/ACCESS.2019.2948146](https://doi.org/10.1109/ACCESS.2019.2948146).
- [223] L. Zhang and J. Xu, "Fooling edge computation offloading via stealthy interference attack," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Nov. 2020, pp. 415–419, doi: [10.1109/SEC50012.2020.00062](https://doi.org/10.1109/SEC50012.2020.00062).
- [224] H. A. Alatwi and A. Aldweesh, "Adversarial black-box attacks against network intrusion detection systems: A survey," in *Proc. IEEE World AI IoT Congr. (AIoT)*, May 2021, pp. 34–40, doi: [10.1109/AIIoT52608.2021.9454214](https://doi.org/10.1109/AIIoT52608.2021.9454214).
- [225] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 11214–11224.
- [226] L. Ma and L. Liang, "Increasing-margin adversarial (IMA) training to improve adversarial robustness of neural networks," 2020, *arXiv:2005.09147*.
- [227] K. Duncan, E. Komendantskaya, R. Stewart, and M. Lones, "Relative robustness of quantized neural networks against adversarial attacks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: [10.1109/IJCNN48605.2020.9207596](https://doi.org/10.1109/IJCNN48605.2020.9207596).
- [228] A. Mirzaeian, J. Kosecka, H. Homayoun, T. Mohsenin, and A. Sasan, "Diverse knowledge distillation (DKD): A solution for improving the robustness of ensemble models against adversarial attacks," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2021, pp. 319–324, doi: [10.1109/ISQED51717.2021.9424353](https://doi.org/10.1109/ISQED51717.2021.9424353).
- [229] Z. Wang, L. Gao, T. Wang, and J. Luo, "Monetizing edge service in mobile internet ecosystem," *IEEE Trans. Mobile Comput.*, vol. 21, no. 5, pp. 1751–1765, May 2022, doi: [10.1109/tmc.2020.3025286](https://doi.org/10.1109/tmc.2020.3025286).
- [230] M. Siew, K. Guo, D. Cai, L. Li, and T. Q. S. Quek, "Let's share VMs: Optimal placement and pricing across base stations in MEC systems," 2021, *arXiv:2101.06129*.
- [231] S. Chen, L. Li, Z. Chen, and S. Li, "Dynamic pricing for smart mobile edge computing: A reinforcement learning approach," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 700–704, Apr. 2021, doi: [10.1109/LWC.2020.3039863](https://doi.org/10.1109/LWC.2020.3039863).
- [232] F. Zhang, Z. Tang, M. Chen, X. Zhou, and W. Jia, "A dynamic resource overbooking mechanism in fog computing," in *Proc. IEEE 15th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2018, pp. 89–97, doi: [10.1109/MASS.2018.00023](https://doi.org/10.1109/MASS.2018.00023).
- [233] S. Li, X. Hu, and Y. Du, "Deep reinforcement learning and game theory for computation offloading in dynamic edge computing markets," *IEEE Access*, vol. 9, pp. 121456–121466, 2021, doi: [10.1109/access.2021.3109132](https://doi.org/10.1109/access.2021.3109132).
- [234] R. Beraldi, A. Mtbbaa, and A. N. Mian, "CICO: A credit-based incentive mechanism for cooperative fog computing paradigms," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7, doi: [10.1109/GLOCOM.2018.8647959](https://doi.org/10.1109/GLOCOM.2018.8647959).
- [235] Z. Li, Z. Yang, S. Xie, W. Chen, and K. Liu, "Credit-based payments for fast computing resource trading in edge-assisted Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6606–6617, Aug. 2019, doi: [10.1109/JIOT.2019.2908861](https://doi.org/10.1109/JIOT.2019.2908861).
- [236] D. Han, W. Chen, and Y. Fang, "A dynamic pricing strategy for vehicle assisted mobile edge computing systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 420–423, Apr. 2019, doi: [10.1109/LWC.2018.2874635](https://doi.org/10.1109/LWC.2018.2874635).
- [237] C. Tang, S. Xia, Q. Li, W. Chen, and W. Fang, "Resource pooling in vehicular fog computing," *J. Cloud Comput.*, vol. 10, no. 1, pp. 1–14, Dec. 2021, doi: [10.1186/s13677-021-00233-x](https://doi.org/10.1186/s13677-021-00233-x).
- [238] D. Zhang, L. Tan, J. Ren, M. K. Awad, S. Zhang, Y. Zhang, and P.-J. Wan, "Near-optimal and truthful online auction for computation offloading in green edge-computing systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 880–893, Apr. 2020, doi: [10.1109/TMC.2019.2901474](https://doi.org/10.1109/TMC.2019.2901474).
- [239] T. Bahreini, H. Badri, and D. Grosu, "An envy-free auction mechanism for resource allocation in edge computing systems," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 313–322, doi: [10.1109/SEC.2018.00030](https://doi.org/10.1109/SEC.2018.00030).
- [240] S. Yang, "A task offloading solution for Internet of Vehicles using combination auction matching model based on mobile edge computing," *IEEE Access*, vol. 8, pp. 53261–53273, 2020, doi: [10.1109/ACCESS.2020.2980567](https://doi.org/10.1109/ACCESS.2020.2980567).
- [241] H. Sun, H. Yu, and G. Fan, "Contract-based resource sharing for time effective task scheduling in fog-cloud environment," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 1040–1053, Jun. 2020, doi: [10.1109/TNSM.2020.2977843](https://doi.org/10.1109/TNSM.2020.2977843).
- [242] T. Wang, Y. Lu, J. Wang, H. N. Dai, X. Zheng, and W. Jia, "EIHD: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for IoT systems," *IEEE Trans. Comput.*, vol. 14, no. 8, pp. 1–14, Feb. 2021, doi: [10.1109/TC.2021.3060484](https://doi.org/10.1109/TC.2021.3060484).
- [243] D. Kim, H. Lee, H. Song, N. Choi, and Y. Yi, "Economics of fog computing: Interplay among infrastructure and service providers, users, and edge resource owners," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2609–2622, Nov. 2020, doi: [10.1109/TMC.2019.2925797](https://doi.org/10.1109/TMC.2019.2925797).
- [244] J. Nakazato, M. Nakamura, T. Yu, Z. Li, K. Maruta, G. K. Tran, and K. Sakaguchi, "Market analysis of MEC-assisted beyond 5G ecosystem," *IEEE Access*, vol. 9, pp. 53996–54008, 2021, doi: [10.1109/ACCESS.2021.3068839](https://doi.org/10.1109/ACCESS.2021.3068839).
- [245] M. Siew, D. Cai, L. Li, and T. Q. S. Quek, "A sharing-economy inspired pricing mechanism for multi-access edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 5–10, doi: [10.1109/GLOBECOM42002.2020.9322554](https://doi.org/10.1109/GLOBECOM42002.2020.9322554).
- [246] Z.-L. Chang and H.-Y. Wei, "Flat-rate pricing for green edge computing with latency guarantee: A Stackelberg game approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6, doi: [10.1109/GLOBECOM38437.2019.9014203](https://doi.org/10.1109/GLOBECOM38437.2019.9014203).
- [247] Y. Hui, Z. Su, T. H. Luan, and C. Li, "Reservation service: Trusted relay selection for edge computing services in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2734–2746, Dec. 2020, doi: [10.1109/JSAC.2020.3005468](https://doi.org/10.1109/JSAC.2020.3005468).
- [248] Z. Xu, Y. Qin, P. Zhou, J. C. S. Lui, W. Liang, Q. Xia, W. Xu, and G. Wu, "To cache or not to cache: Stable service caching in mobile edge-clouds of a service market," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 421–431, doi: [10.1109/ICDCS47774.2020.00051](https://doi.org/10.1109/ICDCS47774.2020.00051).

- [249] H. Lv, Z. Zheng, F. Wu, and G. Chen, "Strategy-proof online mechanisms for weighted AoI minimization in edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1277–1292, May 2021, doi: [10.1109/JSAC.2021.3065078](https://doi.org/10.1109/JSAC.2021.3065078).
- [250] C. J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia, and T. Leyvand, "Machine learning at facebook: Understanding inference at the edge," in *Proc. 25th IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, 2019, pp. 331–344, 2019, doi: [10.1109/HPCA.2019.00048](https://doi.org/10.1109/HPCA.2019.00048).
- [251] H. Wang and B. Raj, "On the origin of deep learning," Feb. 2017, *arXiv:1702.07800*.
- [252] X. Liu, W. Xia, and Z. Fan, "A deep neural network pruning method based on gradient L1-norm," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 2070–2074, doi: [10.1109/ICCC51575.2020.9345039](https://doi.org/10.1109/ICCC51575.2020.9345039).
- [253] S.-K. Yeom, K.-H. Shim, and J.-H. Hwang, "Toward compact deep neural networks via energy-aware pruning," 2021, *arXiv:2103.10858*.
- [254] F. Tung, S. Muralidharan, and G. Mori, "Fine-pruning: Joint fine-tuning and compression of a convolutional network with Bayesian optimization," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 1–12, doi: [10.5244/c.31.115](https://doi.org/10.5244/c.31.115).
- [255] S. Roy, P. Panda, G. Srinivasan, and A. Raghunathan, "Pruning filters while training for efficiently optimizing deep learning networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7, doi: [10.1109/IJCNN48605.2020.9207588](https://doi.org/10.1109/IJCNN48605.2020.9207588).
- [256] F. Yu, L. Cui, P. Wang, C. Han, R. Huang, and X. Huang, "EasiEdge: A novel global deep neural networks pruning method for efficient edge computing," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1259–1271, Feb. 2021, doi: [10.1109/JIOT.2020.3034925](https://doi.org/10.1109/JIOT.2020.3034925).
- [257] X. Zhou, W. Zhang, H. Xu, and T. Zhang, "Effective sparsification of neural networks with global sparsity constraint," 2021, *arXiv:2105.01571*.
- [258] Y. He, J. Lin, Z. Liu, H. Wang, L. J. Li, and S. Han, "AMC: AutoML for model compression and acceleration on mobile devices," *Computer Vision (Lecture Notes in Computer Science)*, vol. 11211, 2018, pp. 815–832, doi: [10.1007/978-3-030-01234-2_48](https://doi.org/10.1007/978-3-030-01234-2_48).
- [259] A. Bstract, "Adaptive weight sparsity for training deep neural networks," in *Proc. ICLR*, 2018, pp. 1–10. [Online]. Available: <https://openreview.net/pdf?id=Bkj1sW-Ab>
- [260] P. Udupa, G. Mahale, K. K. Chandrasekharan, and S. Lee, "IKW: Inter-kernel weights for power efficient edge computing," *IEEE Access*, vol. 8, pp. 90450–90464, 2020, doi: [10.1109/ACCESS.2020.2993506](https://doi.org/10.1109/ACCESS.2020.2993506).
- [261] H. Yang, M. Tang, W. Wen, F. Yan, D. Hu, A. Li, H. Li, and Y. Chen, "Learning low-rank deep neural networks via singular vector orthogonality regularization and singular value sparsification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 2899–2908, doi: [10.1109/CVPRW50498.2020.00347](https://doi.org/10.1109/CVPRW50498.2020.00347).
- [262] M. Nagel, R. A. Amjad, M. van Baalen, C. Louizos, and T. Blankevoort, "Up or down? Adaptive rounding for post-training quantization," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 7154–7163.
- [263] J. Hu, W. L. Goh, and Y. Gao, "Dynamically-biased fixed-point LSTM for time series processing in AIoT edge device," in *Proc. IEEE 3rd Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Washington, DC, USA, Jun. 2021, pp. 1012–1026.
- [264] S.-E. Chang, Y. Li, M. Sun, R. Shi, H. K.-H. So, X. Qian, Y. Wang, and X. Lin, "Mix and match: A novel FPGA-centric deep neural network quantization framework," in *Proc. IEEE Int. Symp. High-Performance Comput. Archit. (HPCA)*, Feb. 2021, pp. 208–220, doi: [10.1109/HPCA51647.2021.00027](https://doi.org/10.1109/HPCA51647.2021.00027).
- [265] C. Baskin, N. Liss, E. Schwartz, E. Zheltonozhskii, R. Giryes, A. M. Bronstein, and A. Mendelson, "UNIQ: Uniform noise injection for non-uniform quantization of neural networks," *ACM Trans. Comput. Syst.*, vol. 37, nos. 1–4, pp. 1–15, Nov. 2019, doi: [10.1145/3444943](https://doi.org/10.1145/3444943).
- [266] Y. Fu, H. You, Y. Zhao, Y. Wang, C. Li, K. Gopalakrishnan, Z. Wang, and Y. Lin, "FracTrain: Fractionally squeezing bit savings both temporally and spatially for efficient DNN training," 2020, *arXiv:2012.13113*.
- [267] T. Huang, T. Luo, and J. T. Zhou, "Adaptive precision training for resource constrained devices," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 1403–1408, doi: [10.1109/ICDCS47774.2020.00185](https://doi.org/10.1109/ICDCS47774.2020.00185).
- [268] Y. Li, J. Cao, Z. Li, S. Oh, and N. Komuro, "Lightweight single image super-resolution with dense connection distillation network," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 17, no. 1s, pp. 1–17, Mar. 2021, doi: [10.1145/3414838](https://doi.org/10.1145/3414838).
- [269] I. Ruiz, B. Raducanu, R. Mehta, and J. Amores, "Optimizing speed/accuracy trade-off for person re-identification via knowledge distillation," *Eng. Appl. Artif. Intell.*, vol. 87, Jan. 2020, Art. no. 103309, doi: [10.1016/j.engappai.2019.103309](https://doi.org/10.1016/j.engappai.2019.103309).
- [270] D. Guo, H. Wang, and M. Wang, "Context-aware graph inference with knowledge distillation for visual dialog," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 1, 2021, doi: [10.1109/TPAMI.2021.3085755](https://doi.org/10.1109/TPAMI.2021.3085755).
- [271] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50X fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*.
- [272] H.-S. Choi, S. Park, J. H. Lee, H. Heo, D. Jeon, and K. Lee, "Real-time denoising and dereverberation with tiny recurrent U-Net," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5789–5793, doi: [10.1109/icassp39728.2021.9414852](https://doi.org/10.1109/icassp39728.2021.9414852).
- [273] D. AbdulQader, S. Krishnan, and C. N. Coelho, Jr., "Enabling incremental training with forward pass for edge devices," 2021, *arXiv:2103.14007*.
- [274] H. Dutta, N. Nataraj, and S. A. Mahindre, "Consensus based multi-layer perceptrons for edge computing," 2021, *arXiv:2102.05021*.
- [275] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *Computer Vision (Lecture Notes in Computer Science)*, vol. 9905, 2016, pp. 21–37, doi: [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2).
- [276] B. Borhanuddin, N. Jamil, S. D. Chen, M. Z. Baharuddin, K. S. Z. Tan, and T. W. M. Ooi, "Small-scale deep network for DCT-based images classification," in *Proc. 4th Int. Conf. Workshops Recent Adv. Innov. Eng. (ICRAIE)*, Nov. 2019, pp. 1–6, doi: [10.1109/ICRAIE47735.2019.9037777](https://doi.org/10.1109/ICRAIE47735.2019.9037777).
- [277] E. Baccarelli, M. Scarpiniti, A. Momenzadeh, and S. S. Ahrabi, "Learning-in-the-Fog (LiFo): Deep learning meets fog computing for the minimum-energy distributed early-exit of inference in delay-critical IoT realms," *IEEE Access*, vol. 9, pp. 25716–25757, 2021, doi: [10.1109/ACCESS.2021.3058021](https://doi.org/10.1109/ACCESS.2021.3058021).
- [278] J. Mao, X. Chen, K. W. Nixon, C. Krieger, and Y. Chen, "MoDNN: Local distributed mobile computing system for deep neural network," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2017, pp. 1396–1401, doi: [10.23919/DATE.2017.7927211](https://doi.org/10.23919/DATE.2017.7927211).
- [279] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: Distributed adaptive deep learning inference on resource-constrained IoT edge clusters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2348–2359, Nov. 2018, doi: [10.1109/TCAD.2018.2858384](https://doi.org/10.1109/TCAD.2018.2858384).
- [280] T. Mohammed, C. Joe-Wong, R. Babbar, and M. D. Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 854–863, doi: [10.1109/INFOCOM41043.2020.9155237](https://doi.org/10.1109/INFOCOM41043.2020.9155237).
- [281] F. Xue, W. Fang, W. Xu, Q. Wang, X. Ma, and Y. Ding, "EdgeLD: Locally distributed deep learning inference on edge device clusters," in *Proc. IEEE 22nd Int. Conf. High Perform. Comput. Communications; IEEE 18th Int. Conf. Smart City; IEEE 6th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2020, pp. 613–619, doi: [10.1109/HPCC-SmartCity-DSS50907.2020.00078](https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00078).
- [282] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Y. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54, 2017, pp. 1273–1282.
- [283] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 300–310, doi: [10.1109/ICDCS47774.2020.00026](https://doi.org/10.1109/ICDCS47774.2020.00026).
- [284] H. H. Yang, A. Arafa, T. Q. S. Quek, and H. V. Poor, "Age-based scheduling policy for federated learning in mobile edge networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8743–8747, doi: [10.1109/ICASSP40776.2020.9053740](https://doi.org/10.1109/ICASSP40776.2020.9053740).
- [285] W. J. Gross, B. H. Meyer, and A. Ardakani, "Hardware-aware design for edge intelligence," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 113–127, 2021, doi: [10.1109/ojcas.2020.3047418](https://doi.org/10.1109/ojcas.2020.3047418).
- [286] C. Zhu, K. Huang, S. Yang, Z. Zhu, H. Zhang, and H. Shen, "An efficient hardware accelerator for structured sparse convolutional neural networks on FPGAs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 28, no. 9, pp. 1953–1965, Sep. 2020, doi: [10.1109/TVLSI.2020.3002779](https://doi.org/10.1109/TVLSI.2020.3002779).
- [287] K. Zhang, C. Hawkins, X. Zhang, C. Hao, and Z. Zhang, "On-FPGA training with ultra memory reduction: A low-precision tensor method," 2021, *arXiv:2104.03420*.

- [288] I. Colbert, J. Daly, K. Kreutz-Delgado, and S. Das, "A competitive edge: Can FPGAs beat GPUs at DCNN inference acceleration in resource-limited edge computing applications?" 2021, *arXiv:2102.00294*.
- [289] L. Yang, Z. Yan, M. Li, H. Kwon, L. Lai, T. Krishna, V. Chandra, W. Jiang, and Y. Shi, "Co-exploration of neural architectures and heterogeneous ASIC accelerator designs targeting multiple tasks," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6, doi: [10.1109/DAC18072.2020.9218676](https://doi.org/10.1109/DAC18072.2020.9218676).
- [290] *Artificial Intelligence & Autopilot | Tesla*. Accessed: Mar. 31, 2022. [Online]. Available: <https://www.tesla.com/AI>
- [291] A. Yazdanbakhsh, K. Seshadri, B. Akin, J. Laudon, and R. Narayanaswami, "An evaluation of edge TPU accelerators for convolutional neural networks," 2021, *arXiv:2102.10423*.
- [292] Y. E. Wang, G.-Y. Wei, and D. Brooks, "Benchmarking TPU, GPU, and CPU platforms for deep learning," 2019, *arXiv:1907.10701*.
- [293] J. Feldmann, N. Youngblood, M. Karpov, H. Gehring, X. Li, M. Stappers, M. Le Gallo, X. Fu, A. Lukashchuk, A. Raja, J. Liu, D. Wright, A. Sebastian, T. Kippenberg, W. Pernice, and H. Bhaskaran, "Parallel convolution processing using an integrated photonic tensor core," 2020, *arXiv:2002.00281*.
- [294] W. Zhang, B. Gao, J. Tang, P. Yao, S. Yu, M.-F. Chang, H.-J. Yoo, H. Qian, and H. Wu, "Neuro-inspired computing chips," *Nature Electron.*, vol. 3, no. 7, pp. 371–382, Jul. 2020, doi: [10.1038/s41928-020-0435-7](https://doi.org/10.1038/s41928-020-0435-7).
- [295] I. H. Im, S. J. Kim, and H. W. Jang, "Memristive devices for new computing paradigms," *Adv. Intell. Syst.*, vol. 2, no. 11, Nov. 2020, Art. no. 2000105, doi: [10.1002/aisy.202000105](https://doi.org/10.1002/aisy.202000105).
- [296] C.-X. Xue, Y.-C. Chiu, T.-W. Liu, T.-Y. Huang, J.-S. Liu, T.-W. Chang, H.-Y. Kao, J.-H. Wang, S.-Y. Wei, C.-Y. Lee, and S.-P. Huang, "A CMOS-integrated compute-in-memory macro based on resistive random-access memory for AI edge devices," *Nature Electron.*, vol. 4, no. 1, pp. 81–90, Jan. 2021, doi: [10.1038/s41928-020-00505-5](https://doi.org/10.1038/s41928-020-00505-5).
- [297] Z. Wang, J. Hu, G. Min, Z. Zhao, and J. Wang, "Data-augmentation-based cellular traffic prediction in edge-computing-enabled smart city," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 1–9, Jul. 2020.
- [298] E. T. Michailidis, N. I. Miridakis, A. Michalakis, E. Skondras, D. J. Vergados, and D. D. Vergados, "Energy optimization in massive MIMO UAV-aided MEC-enabled vehicular networks," *IEEE Access*, vol. 9, pp. 117388–117403, 2021, doi: [10.1109/ACCESS.2021.3106495](https://doi.org/10.1109/ACCESS.2021.3106495).
- [299] T. Bai, C. Pan, C. Han, and L. Hanzo, "Reconfigurable intelligent surface aided mobile edge computing," 2021, *arXiv:2102.02569*.
- [300] J. Wu, B. Yang, L. Wang, and J. Park, "Adaptive DRX method for MTC device energy saving by using a machine learning algorithm in an MEC framework," *IEEE Access*, vol. 9, pp. 10548–10560, 2021, doi: [10.1109/ACCESS.2021.3049532](https://doi.org/10.1109/ACCESS.2021.3049532).
- [301] M. Merluzzi, N. di Pietro, P. Di Lorenzo, E. C. Strinati, and S. Barbarossa, "Discontinuous computation offloading for energy-efficient mobile edge computing," 2020, *arXiv:2008.03508*.
- [302] B. Nour and S. Cherkaoui, "A network-based compute reuse architecture for IoT applications," 2021, *arXiv:2104.03818*.
- [303] L. Li, G. Xu, P. Liu, Y. Li, and J. Ge, "Jointly optimize the residual energy of multiple mobile devices in the MEC-WPT system," *Futur. Internet*, vol. 12, no. 12, pp. 1–18, 2020, doi: [10.3390/fi12120233](https://doi.org/10.3390/fi12120233).
- [304] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021, doi: [10.1109/ACCESS.2021.3050155](https://doi.org/10.1109/ACCESS.2021.3050155).
- [305] Q. Liu, T. Han, and E. Moges, "EdgeSlice: Slicing wireless edge computing network with decentralized deep reinforcement learning," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 234–244, doi: [10.1109/ICDCS47774.2020.00028](https://doi.org/10.1109/ICDCS47774.2020.00028).
- [306] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint planning of network slicing and mobile edge computing: Models and algorithms," *IEEE Trans. Cloud Comput.*, early access, Aug. 24, 2021, doi: [10.1109/TCC.2021.3107022](https://doi.org/10.1109/TCC.2021.3107022).
- [307] L. N. T. Huynh, Q.-V. Pham, T. D. T. Nguyen, M. D. Hossain, Y.-R. Shin, and E.-N. Huh, "Joint computational offloading and data-content caching in NOMA-MEC networks," *IEEE Access*, vol. 9, pp. 12943–12954, 2021, doi: [10.1109/ACCESS.2021.3051278](https://doi.org/10.1109/ACCESS.2021.3051278).
- [308] X. Tang, C. Cao, Y. Wang, S. Zhang, Y. Liu, M. Li, and T. He, "Computing power network: The architecture of convergence of computing and networking towards 6G requirement," *China Commun.*, vol. 18, no. 2, pp. 175–185, Feb. 2021, doi: [10.23919/JCC.2021.02.011](https://doi.org/10.23919/JCC.2021.02.011).
- [309] Q. Zhang, L. Liu, C. Pu, Q. Dou, L. Wu, and W. Zhou, "A comparative study of containers and virtual machines in big data environment," in *Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2018, pp. 178–185, doi: [10.1109/CLOUD.2018.00030](https://doi.org/10.1109/CLOUD.2018.00030).
- [310] N. G. Bachiega, P. S. L. Souza, S. M. Bruschi, and S. D. R. S. de Souza, "Container-based performance evaluation: A survey and challenges," in *Proc. IEEE Int. Conf. Cloud Eng. (ICE)*, Apr. 2018, pp. 398–403, doi: [10.1109/IC2E.2018.00075](https://doi.org/10.1109/IC2E.2018.00075).
- [311] I. M. A. Jawarneh, P. Bellavista, F. Bosi, L. Foschini, G. Martuscelli, R. Montanari, and A. Palopoli, "Container orchestration engines: A thorough functional and performance comparison," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: [10.1109/ICC.2019.8762053](https://doi.org/10.1109/ICC.2019.8762053).
- [312] M. Gawel and K. Zielinski, "Analysis and evaluation of kubernetes based NFV management and orchestration," in *Proc. IEEE 12th Int. Conf. Cloud Comput. (CLOUD)*, Jul. 2019, pp. 511–513, doi: [10.1109/CLOUD.2019.00094](https://doi.org/10.1109/CLOUD.2019.00094).
- [313] L. Civolani, G. Pierre, and P. Bellavista, "FogDocker: Start container now, fetch image later," in *Proc. 12th IEEE/ACM Int. Conf. Utility Cloud Comput.*, Dec. 2019, pp. 51–59, doi: [10.1145/3344341.3368811](https://doi.org/10.1145/3344341.3368811).
- [314] L. Ma, S. Yi, N. Carter, and Q. Li, "Efficient live migration of edge services leveraging container layered storage," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2020–2033, Sep. 2019, doi: [10.1109/TMC.2018.2871842](https://doi.org/10.1109/TMC.2018.2871842).
- [315] S. R. Chaudhry, A. Palade, A. Kazmi, and S. Clarke, "Improved QoS at the edge using serverless computing to deploy virtual network functions," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10673–10683, Oct. 2020, doi: [10.1109/JIOT.2020.3011057](https://doi.org/10.1109/JIOT.2020.3011057).
- [316] R. Xiao, Y. Zhang, X. H. Cui, F. Zhang, and H. H. Wang, "A hybrid task crash recovery solution for edge computing in IoT-based manufacturing," *IEEE Access*, vol. 9, pp. 106220–106231, 2021, doi: [10.1109/ACCESS.2021.3068471](https://doi.org/10.1109/ACCESS.2021.3068471).
- [317] R. Mahmud and A. N. Toosi, "Con-Pi: A distributed container-based edge and fog computing framework," 2021, *arXiv:2101.03533*.
- [318] M. V. Ngo, T. Luo, H. T. Hoang, and T. Q. S. Ouek, "Coordinated container migration and base station handover in mobile edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6, doi: [10.1109/GLOBECOM42002.2020.9322368](https://doi.org/10.1109/GLOBECOM42002.2020.9322368).
- [319] V. Kjørveziroski, C. B. Canto, P. J. Roig, K. Gilly, A. Mishev, V. Trajkovic, and S. Filiposka, "IoT serverless computing at the edge: Open issues and research direction," *Trans. Netw. Commun.*, vol. 9, no. 4, pp. 1–33, Dec. 2021, doi: [10.14738/tnc.94.11231](https://doi.org/10.14738/tnc.94.11231).
- [320] M. S. Aslanpour, A. N. Toosi, C. Cicconetti, B. Javadi, P. Sbarski, D. Taibi, M. Assuncao, S. S. Gill, R. Gaire, and S. Dustdar, "Serverless edge computing: Vision and challenges," in *Proc. Australas. Comput. Sci. Week Multiconference*, Feb. 2021, pp. 1–10, doi: [10.1145/3437378.3444367](https://doi.org/10.1145/3437378.3444367).
- [321] W. Jin, R. Xu, S. Lim, D. H. Park, C. Park, and D. Kim, "Dynamic inference approach based on rules engine in intelligent edge computing for building environment control," *Sensors*, vol. 21, no. 2, pp. 1–21, 2021, doi: [10.3390/s21020630](https://doi.org/10.3390/s21020630).
- [322] K. C. Serdaroglu and S. Baydere, "Edge computing based smart meeting room controller," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Jun. 2019, pp. 1–4, doi: [10.1109/ISNCC.2019.8909191](https://doi.org/10.1109/ISNCC.2019.8909191).
- [323] J. Chen, K. Li, Q. Deng, K. Li, and P. S. Yu, "Distributed deep learning model for intelligent video surveillance systems with edge computing," *IEEE Trans. Ind. Informat.*, early access, Apr. 4, 2020, doi: [10.1109/tii.2019.2909473](https://doi.org/10.1109/tii.2019.2909473).
- [324] D. S. Johnson, W. Lorenz, M. Taenser, S. Mimalakis, S. Grollmisch, J. Abeßer, and H. Lukashevich, "DESED-FL and URBAN-FL: Federated learning datasets for sound event detection," 2021, *arXiv:2102.08833*.
- [325] N. Najji, M. R. Abid, D. Benhaddou, and N. Krami, "Context-aware wireless sensor networks for smart building energy management system," *Information*, vol. 11, no. 11, pp. 1–21, 2020, doi: [10.3390/info11110530](https://doi.org/10.3390/info11110530).
- [326] A. Mukherjee, S. Misra, A. Sukrutha, and N. S. Raghuvanshi, "Distributed aerial processing for IoT-based edge UAV swarms in smart farming," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 107038, doi: [10.1016/j.comnet.2019.107038](https://doi.org/10.1016/j.comnet.2019.107038).
- [327] O. Debauche, S. Mahmoudi, S. A. Mahmoudi, P. Manneback, J. Bindelle, and F. Lebeau, "Edge computing for cattle behavior analysis," in *Proc. 2nd Int. Conf. Embedded Distrib. Syst. (EDiS)*, Nov. 2020, pp. 52–57, doi: [10.1109/EDiS49545.2020.9296471](https://doi.org/10.1109/EDiS49545.2020.9296471).
- [328] K. Lee, B. N. Silva, and K. Han, "Deep learning entrusted to fog nodes (DLEFN) based smart agriculture," *Appl. Sci.*, vol. 10, no. 4, p. 1544, Feb. 2020, doi: [10.3390/app10041544](https://doi.org/10.3390/app10041544).

- [329] L. Hou, K. Zheng, Z. Liu, X. Xu, and T. Wu, "Design and prototype implementation of a blockchain-enabled Lora system with edge computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2419–2430, Feb. 2021, doi: [10.1109/IJOT.2020.3027713](https://doi.org/10.1109/IJOT.2020.3027713).
- [330] A. Willner and V. Gowtham, "Toward a reference architecture model for industrial edge computing," *IEEE Commun. Standards Mag.*, vol. 4, no. 4, pp. 42–48, Dec. 2020, doi: [10.1109/MCOMSTD.001.2000007](https://doi.org/10.1109/MCOMSTD.001.2000007).
- [331] F. Liu, C. Liang, and Q. He, "Remote malfunctioning smart meter detection in edge computing environment," *IEEE Access*, vol. 8, pp. 67436–67443, 2020, doi: [10.1109/ACCESS.2020.2985725](https://doi.org/10.1109/ACCESS.2020.2985725).
- [332] T. Hafeez, L. Xu, and G. Mcardle, "Edge intelligence for data handling and predictive maintenance in IIOT," *IEEE Access*, vol. 9, pp. 49355–49371, 2021, doi: [10.1109/ACCESS.2021.3069137](https://doi.org/10.1109/ACCESS.2021.3069137).
- [333] A. K. Tanwani, N. Mor, J. Kubiawicz, J. E. Gonzalez, and K. Goldberg, "A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4559–4566, doi: [10.1109/ICRA.2019.8793690](https://doi.org/10.1109/ICRA.2019.8793690).
- [334] J. Mehami, M. Nawwi, and R. Y. Zhong, "Smart automated guided vehicles for manufacturing in the context of industry 4.0," *Proc. Manuf.*, vol. 26, pp. 1077–1086, Jan. 2018, doi: [10.1016/j.promfg.2018.07.144](https://doi.org/10.1016/j.promfg.2018.07.144).
- [335] X. Zhu, X. Zeng, and W. Ma, "Lightweight cross-fusion network on human pose estimation for edge device," *IEEE Access*, vol. 1, p. 1, 2021, doi: [10.1109/ACCESS.2021.3065574](https://doi.org/10.1109/ACCESS.2021.3065574).
- [336] H. Gao, W. Huang, and Y. Duan, "The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–23, Feb. 2021, doi: [10.1145/3391198](https://doi.org/10.1145/3391198).
- [337] A. Mandal, A. Sinaeepourfard, and S. K. Naskar, "VDA: Deep learning based visual data analysis in integrated edge to cloud computing environment," in *Proc. Adjunct Proc. Int. Conf. Distrib. Comput. Netw.*, Jan. 2021, pp. 7–12, doi: [10.1145/3427477.3429781](https://doi.org/10.1145/3427477.3429781).
- [338] E.-T. Bouali, M. R. Abid, E.-M. Boufounas, T. A. Hamed, and D. Benhaddou, "Renewable energy integration into cloud & IoT-based smart agriculture," *IEEE Access*, vol. 10, pp. 1175–1191, 2022, doi: [10.1109/ACCESS.2021.3138160](https://doi.org/10.1109/ACCESS.2021.3138160).
- [339] S. Shao, Q. Zhang, S. Guo, and F. Qi, "Task allocation mechanism for cable real-time online monitoring business based on edge computing," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1344–1355, Mar. 2021, doi: [10.1109/JSYST.2020.2988266](https://doi.org/10.1109/JSYST.2020.2988266).
- [340] C. Lee, S.-H. Kim, and C.-H. Youn, "Cooperating edge cloud-based hybrid online learning for accelerated energy data stream processing in load forecasting," *IEEE Access*, vol. 8, pp. 199120–199132, 2020, doi: [10.1109/ACCESS.2020.3035421](https://doi.org/10.1109/ACCESS.2020.3035421).
- [341] T. Li, J. Yang, and D. Cui, "Artificial-intelligence-based algorithms in multi-access edge computing for the performance optimization control of a benchmark microgrid," *Phys. Commun.*, vol. 44, Feb. 2021, Art. no. 101240, doi: [10.1016/j.phycom.2020.101240](https://doi.org/10.1016/j.phycom.2020.101240).
- [342] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 217–228, Jan. 2020, doi: [10.1109/SAC.2019.2951966](https://doi.org/10.1109/SAC.2019.2951966).
- [343] P. Arthurs, L. Gillam, P. Krause, N. Wang, K. Halder, and A. Mouzakitis, "A taxonomy and survey of edge cloud computing for intelligent transportation systems and connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 7, 2021, doi: [10.1109/TITS.2021.3084396](https://doi.org/10.1109/TITS.2021.3084396).
- [344] Q. Yuan, J. Li, H. Zhou, G. Luo, T. Lin, F. Yang, and X. S. Shen, "Cross-domain resource orchestration for the edge-computing-enabled smart road," *IEEE Netw.*, vol. 34, no. 5, pp. 60–67, Sep. 2020, doi: [10.1109/MNET.011.2000007](https://doi.org/10.1109/MNET.011.2000007).
- [345] X. Xu, K. Liu, K. Xiao, L. Feng, Z. Wu, and S. Guo, "Vehicular fog computing enabled real-time collision warning via trajectory calibration," *Mobile Netw. Appl.*, vol. 25, no. 6, pp. 2482–2494, Dec. 2020, doi: [10.1007/s11036-020-01591-7](https://doi.org/10.1007/s11036-020-01591-7).
- [346] X. Huang, P. He, A. Rangarajan, and S. Ranka, "Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 2, pp. 1–28, 2020, doi: [10.1145/3373647](https://doi.org/10.1145/3373647).
- [347] N. Ameer, P. Kumar, V. Kumar, P. Dinesh, and M. M. D. Babu, "Automated sensing system for monitoring road surface condition using fog computing," *Int. J. Adv. Eng., Manage. Sci.*, vol. 4, no. 3, pp. 215–218, 2018, doi: [10.22161/ijaems.4.3.12](https://doi.org/10.22161/ijaems.4.3.12).
- [348] S. Jiang, X. Li, and J. Wu, "Hierarchical edge-cloud computing for mobile blockchain mining game," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 1327–1336, doi: [10.1109/ICDCS.2019.00133](https://doi.org/10.1109/ICDCS.2019.00133).
- [349] Y. Fan, L. Wang, W. Wu, and D. Du, "Cloud/edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 2, pp. 451–463, Apr. 2021, doi: [10.1109/TCSS.2021.3049152](https://doi.org/10.1109/TCSS.2021.3049152).
- [350] A. F. Subahi, "Edge-based IoT medical record system: Requirements, recommendations and conceptual design," *IEEE Access*, vol. 7, pp. 94150–94159, 2019, doi: [10.1109/ACCESS.2019.2927958](https://doi.org/10.1109/ACCESS.2019.2927958).
- [351] I. García-Magariño, J. Varela-Aldas, G. Palacios-Navarro, and J. Lloret, "Fog computing for assisting and tracking elder patients with neurodegenerative diseases," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1225–1235, Sep. 2019, doi: [10.1007/s12083-019-00732-4](https://doi.org/10.1007/s12083-019-00732-4).
- [352] S. Sedaghat and A. H. Jahangir, "RT-TelSurg: Real time telesurgery using SDN, fog, and cloud as infrastructures," *IEEE Access*, vol. 9, pp. 52238–52251, 2021, doi: [10.1109/ACCESS.2021.3069744](https://doi.org/10.1109/ACCESS.2021.3069744).
- [353] V. Hayyolalam, M. Aloqaily, O. Ozkasap, and M. Guizani, "Edge intelligence for empowering IoT-based healthcare systems," 2021, *arXiv:2103.12144*.
- [354] S. Sakib, M. M. Fouda, Z. M. Fadlullah, N. Nasser, and W. Alasmarty, "A proof-of-concept of ultra-edge smart IoT sensor: A continuous and lightweight arrhythmia monitoring approach," *IEEE Access*, vol. 9, pp. 26093–26106, 2021, doi: [10.1109/ACCESS.2021.3056509](https://doi.org/10.1109/ACCESS.2021.3056509).
- [355] M. A. Asghar, M. J. Khan, H. Shahid, M. Shoruffuzaman, N. N. Xiong, and R. M. Mehmood, "Semi-skipping layered gated unit and efficient network: Hybrid deep feature selection method for edge computing in EEG-based emotion classification," *IEEE Access*, vol. 9, pp. 13378–13389, 2021, doi: [10.1109/ACCESS.2021.3051808](https://doi.org/10.1109/ACCESS.2021.3051808).
- [356] I. L. Olokodana, S. P. Mohanty, and E. Kougianos, "Ordinary-kriging based real-time seizure detection in an edge computing paradigm," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2020, pp. 1–6, doi: [10.1109/ICCE46568.2020.9043004](https://doi.org/10.1109/ICCE46568.2020.9043004).
- [357] K. Lee and C.-H. Youn, "REINDEAR: REINforcement learning agent for dynamic system control in edge-assisted augmented reality service," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 949–954, doi: [10.1109/ICTC49870.2020.9289225](https://doi.org/10.1109/ICTC49870.2020.9289225).
- [358] C. Sarathchandra, K. Haensge, S. Robitzsch, M. Ghassemian, and U. Olvera-Hernandez, "Enabling bi-directional haptic control in next generation communication systems: Research, standards, and vision," 2021, *arXiv:2104.04297*.
- [359] B. Coffen and M. S. Mahmud, "TinyDL: Edge computing and deep learning based real-time hand gesture recognition using wearable sensor," in *Proc. IEEE Int. Conf. E-Health Netw., Appl. Services (HEALTHCOM)*, Mar. 2021, pp. 1–6, doi: [10.1109/HEALTHCOM49281.2021.9399005](https://doi.org/10.1109/HEALTHCOM49281.2021.9399005).
- [360] A. Akter, A. Islam, and S. Y. Shin, "Mobile edge computing based mixed reality application for the assistance of blind and visually impaired people," in *Proc. 7th Int. Conf. Inf. Commun. Technol. (ICoICT)*, Jul. 2019, pp. 1–5.
- [361] M. S. Hossain, F. B. Islam, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Decentralized latency-aware edge node grouping with fault tolerance for internet of battlefield things," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 420–423, doi: [10.1109/ICTC49870.2020.9289442](https://doi.org/10.1109/ICTC49870.2020.9289442).
- [362] X. Hou, Z. Ren, J. Wang, S. Zheng, W. Cheng, and H. Zhang, "Distributed fog computing for latency and reliability guaranteed swarm of drones," *IEEE Access*, vol. 8, pp. 7117–7130, 2020, doi: [10.1109/ACCESS.2020.2964073](https://doi.org/10.1109/ACCESS.2020.2964073).
- [363] H. Zhang, X. Zhang, and J. Huang, "A demonstration system for the generation and interaction of battlefield situation based on hololens," in *Proc. 2nd Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, Sep. 2019, pp. 293–296, doi: [10.1109/ICISCAE48440.2019.221638](https://doi.org/10.1109/ICISCAE48440.2019.221638).
- [364] M. M. Omwenga, D. Wu, Y. Liang, L. Yang, D. Huston, and T. Xia, "Autonomous cognitive GPR based on edge computing and reinforcement learning," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Nov. 2019, pp. 348–354, doi: [10.1109/ICII.2019.00066](https://doi.org/10.1109/ICII.2019.00066).
- [365] K. M. Bellazi, R. Marino, J. M. Lanza-Gutierrez, and T. Riesgo, "Towards an machine learning-based edge computing oriented monitoring system for the desert border surveillance use case," *IEEE Access*, vol. 8, pp. 218304–218322, 2020, doi: [10.1109/ACCESS.2020.3042699](https://doi.org/10.1109/ACCESS.2020.3042699).

- [366] J. P. Queralt, J. Raitoharju, T. N. Gia, N. Passalis, and T. Westerlund, "AutoSOS: Towards multi-UAV systems supporting maritime search and rescue with lightweight AI and edge computing," 2020, *arXiv:2005.03409*.
- [367] S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "EC-SAGINs: Edge computing-enhanced space-air-ground integrated networks for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 1–13, Apr. 2021, doi: [10.1109/JIOT.2021.3052542](https://doi.org/10.1109/JIOT.2021.3052542).
- [368] Q. Li, S. Wang, X. Ma, Q. Sun, H. Wang, S. Cao, and F. Yang, "Service coverage for satellite edge computing," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 695–705, Jan. 2022, doi: [10.1109/JIOT.2021.3085129](https://doi.org/10.1109/JIOT.2021.3085129).
- [369] V. Leon, G. Lentar, E. Petrongonas, D. Soudris, G. Furano, A. Tavoularis, and D. Moloney, "Improving performance-power-programmability in space avionics with edge devices: VBN on Myriad2 SoC," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 3, pp. 1–23, May 2021, doi: [10.1145/3440885](https://doi.org/10.1145/3440885).
- [370] B. Denby and B. Lucia, "Orbital edge computing: Machine inference in space," *IEEE Comput. Archit. Lett.*, vol. 18, no. 1, pp. 59–62, Jan. 2019, doi: [10.1109/LCA.2019.2907539](https://doi.org/10.1109/LCA.2019.2907539).
- [371] M. S. Munir, N. H. Tran, W. Saad, and C. S. Hong, "Multi-agent meta-reinforcement learning for self-powered and sustainable edge computing systems," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 3, pp. 3353–3374, Sep. 2021, doi: [10.1109/TNSM.2021.3057960](https://doi.org/10.1109/TNSM.2021.3057960).
- [372] B. Guler and A. Yener, "Energy-harvesting distributed machine learning," 2021, *arXiv:2102.05639*.
- [373] A. Rennoch and A. Willner, "Edge Computing standardisation and initiatives," *INFORMATIK*, pp. 333–337, 2021, doi: [10.18420/inf2020_31](https://doi.org/10.18420/inf2020_31).



SALMANE DOUCH (Graduate Student Member, IEEE) received the degree in electronics engineering from the Department of Electrical Engineering, Hassania School of Public Works, Casablanca, in 2020. He is currently pursuing the Ph.D. degree with the Smart System Laboratory, Mohammed V. University of Rabat. He is also a Research Assistant in a research project funded by the USAID with AI Akhawayn University, Ifran. His main research interests include cloud computing, edge computing, edge intelligence, AI safety, and security at edge.



MOHAMED RIDUAN ABID (Member, IEEE) received the Ph.D. degree in computer science from Auburn University, USA, in 2010, following a Fulbright Student Scholarship. He is currently an Associate Professor with Columbus State University. In 2007, he received a Fulbright Research Scholarship to research at the University of Houston, USA. He is a USA National Academy of Sciences (NAS) Arab-American Fellow. In 2018, he received the NAS Scholarship to join Purdue University, USA. He is fond of coaching and programming. He has over 50 publications in renowned conferences and journals. His main research interests include cloud computing, the Internet of Things (IoT), and high-performance computing (HPC). He is a member of ACM. He won, for six consecutive years (2014–2019), the first prize as a Coach at the National ACM Moroccan Collegiate Programming Contest (MCPC) organized by the Moroccan—Association for Computer Machinery (ACM); and qualified twice to the ACM International Collegiate Programming Contest (ICPC) World Finals in Marrakesh, in 2015; and in Beijing, in 2018.



KHALID ZINE-DINE received the Ph.D. degree from the Mohammed V. University of Rabat, Morocco, in 2000. He spent four years in bank information systems as the Networks and Systems Security Project Manager. Currently, he is a Full Professor with the Faculty of Sciences, Mohammed V. University of Rabat. His research interests include area of wireless ad hoc and sensor networks, mobility, cloud computing, and systems and networks architectures and protocols.

Recently, he has been interested in the field of microgrid and energy efficiency. He was a co-organizer and the co-chair of conferences and is involved in more than six Ph.D. thesis and funded projects.



DRISS BOUZIDI is an Associate Professor in computer sciences with ENSIAS, Mohammed V. University of Rabat, Morocco. His research interests include distributed systems and security services. He has made many contributions to several chapters in some international books related to e-learning. He was the Vice-Chair of the international conference NGNS'09; the TCP Chair of NGNS10, NGNS12, and ICEER13; and the Chair of JDSIRT'2017. He is a Founding Member of two

research associations APRIMT and e-NGN.



DRISS BENHADDOU (Senior Member, IEEE) received the Ph.D. degree in optoelectronics from Montpellier 2 University, France, and the Ph.D. degree in computer networks and telecommunications from the University of Missouri-Kansas City. He is currently a Professor and a Fulbright Scholar with the University of Houston, where he is actively involved in a broad range of research activities in smart systems development, wireless and optical networking space, optical instrumentation, and the IoT application.

In addition, he is spearheading the development of a new state-of-the-art wireless and optical networking research laboratory at the University of Houston. He has four years of industry experience working with Sprint and Lambda Optical Systems Inc., where he played a key role in systems integration activities. He has published more than 130 peer-reviewed publications.

...