Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey

Wajid Rafique, Lianyong Qi, *Member, IEEE*, Ibrar Yaqoob, *Senior Member, IEEE*, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou, *Senior Member, IEEE*

***Abstract*—Millions of sensors continuously produce and transmit data to control real-world infrastructures using complex networks in the Internet of Things (IoT). However, IoT devices are limited in computational power, including storage, processing, and communication resources, to effectively perform compute-intensive tasks locally. Edge computing resolves the resource limitation problems by bringing computation closer to the edge of IoT devices. Providing distributed edge nodes across the network reduces the stress of centralized computation and overcomes latency challenges in the IoT. Therefore, edge computing presents low-cost solutions for compute-intensive tasks. Software-Defined Networking (SDN) enables effective network management by presenting a global perspective of the network. While SDN was not explicitly developed for IoT challenges, it can, however, provide impetus to solve the complexity issues and help in efficient IoT service orchestration. The current IoT paradigm of massive data generation, complex infrastructures, security vulnerabilities, and requirements from the newly developed technologies make IoT realization a challenging issue. In this research, we provide an extensive survey on SDN and the edge computing ecosystem to solve the challenge of complex IoT management. We present the latest research on Software-Defined Internet of Things orchestration using Edge (SDIoT-Edge) and highlight key requirements and standardization efforts in integrating these diverse architectures. An extensive discussion on different case studies using SDIoT-Edge computing is presented to envision the underlying concept. Furthermore, we classify state-of-the-art research in the SDIoT-Edge ecosystem based on multiple performance parameters. We comprehensively present security and privacy vulnerabilities in the SDIoT-Edge computing and provide detailed taxonomies of multiple attack possibilities in this paradigm. We highlight the lessons learned based on our findings at the end of each section. Finally, we discuss critical insights toward current research issues, challenges, and further research directions to efficiently provide IoT services in the SDIoT-Edge paradigm.**

***Index Terms*—Edge computing, Internet of Things, software-defined networking, software-defined IoT, network virtualization, IoT service orchestration.**

I. Introduction

RECENT developments in the field of Ubiquitous Computing (Ubicomp) advocate Mark Weiser's prediction of indispensable human dependence on computer systems [1]. His vision set the path for developments in the field of Ubicomp, where mobile phones and smart devices have now become an integral part of our lives [2]. These devices are equipped with a multitude of sensors, audiovisual features, and intelligent applications. Smart devices, such as watches, gadgets, bracelets, and accompanying smartphones make ubiquitousness a reality [3].

Humankind has stepped into an era of Ubicomp where wearable devices regularly log data, implement various services, and pass this information to the network at regular intervals [2]. It has been estimated that more than 50 billion devices will connect to the Internet until 2025 [4]. This exponential increase in data generation resources poses a vital challenge to communication technology [5]. A wide range of smart devices has been introduced in the market, including vehicles, wearable gadgets, measurement sensors, home appliances, healthcare, and industrial products [3]. Internet of Things (IoT) has received immense attention from industry and academia due to the growing need for IoT devices in everyday life [6], [7]. IoT devices have been involved in providing enormous economic contributions during the past few years. Therefore, extensive efforts have been put forward toward their active development and deployment [8]. The success of the Internet lies in the interoperability and open

Manuscript received June 3, 2019; revised December 1, 2019 and April 20, 2020; accepted May 21, 2020. Date of publication May 26, 2020; date of current version August 21, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1001801, in part by the National Science Foundation of China under Grant 61672276 and Grant 61872219, in part by the Natural Science Foundation of Shandong Province under Grant ZR2019MF001, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2019104, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University. *(Corresponding author: Wanchun Dou.)*

Wajid Rafique and Wanchun Dou are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China, and also with the Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China (e-mail: rafiqwajid@smail.nju.edu.cn; douwc@nju.edu.cn).

Lianyong Qi is with the School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China (e-mail: lianyongqi@gmail.com).

Ibrar Yaqoob is with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: ibraryaqoob@ieee.org).

Muhammad Imran is with the College of Computer and Information Science, King Saud University, Riyadh 11451, Saudi Arabia (e-mail: cimran@ksu.edu.sa).

Raihan Ur Rasool is with the College of Engineering and Science, Victoria University, Melbourne, VIC 3000, Australia (e-mail: raihan.rasool@live.vu.edu.au).

Digital Object Identifier 10.1109/COMST.2020.2997475

access to multiple hardware and software platforms [9]–[11]. However, diverse IoT architectures provoke disparate network implementations [12]. Thus, different data formats, communication procedures, and protocols pose a complex challenge that makes IoT a vertically fragmented network system [13]. An increase in the number of connected devices introduces a massive amount of data, which poses another challenge for today's networks to handle it effectively [14], [15]. Therefore, we must develop new edge technologies that classify and filter IoT big data before transmitting to the central cloud data center. Similarly, the term associated with IoT big data underlines many anticipated challenges related to data management, privacy, and provenance [16].

Edge computing renders the ability to process compute-intensive tasks of the resource-limited IoT devices that cannot be performed locally [17], [18]. It can effectively distribute network computation and avoid peak loads in IoT networks [19]. It has gained tremendous attention over the past few years, such that researchers, technology leaders, and governments are putting their effort toward wider edge computing deployment [20]. It brings the computation resources closer to mobile devices to support resource-limited IoT infrastructure to effectively perform complex computations [21]. However, moving the computational infrastructure closer to the locality brings many technical challenges related to service discovery, mobility management, and user handover [22], [23]. The performance of the IoT applications is degraded while sending data to the edge and then waiting for the response. Edge-based devices have many applications, including transportation, homes, healthcare, cities, and buildings [24]. Edge cloudlets can be placed between IoT infrastructure and the central cloud to support the delay-sensitive IoT applications. The central cloud infrastructure possesses powerful data centers to execute higher latency service requests [25].

Communication networks weave the fabric of today's digital world, where the significance of a network is ascertained by Metcalfe's law, which states that the value of a communication network is directly proportional to the number of connected devices [42]. Therefore, Software-Defined Networking (SDN) has become an important technology for network service provisioning due to its flexible management and programmability [26], [43]. SDN provides a layered framework where each plane operates separately, including data, control, and application planes [44]. Separation of the data and the control plane facilitates network administration at runtime, traffic management, network evolution, and flexible network programmability [45], [46]. Thus, due to the widespread proliferation of IoT and its management complexities, Software-Defined Internet of Things (SDIoT) architecture has been proposed for an effective management [47]–[50]. Similarly, edge computing brings cloud services near the edge of IoT to increase scalability and interoperability [51]. Novel SDN and edge-based IoT implementations create a new communication perspective for effective service provisioning. The cloud services are integral for IoT realization; therefore, SDIoT service orchestration using edge computing has been adopted to realize Software-Defined Internet of Things and Edge (SDIoT-Edge) framework [52]–[55]. This architecture helps in efficient IoT realization; however, the disparate set of hardware infrastructures pose new challenges of communication, interoperability, management, and lack of a unified architecture. Security, reliability, and privacy of devices and data produced and transferred during IoT operation also pose further challenges in the SDIoT-Edge ecosystem [56], [57]. Communication technologies need to be evolved with the development of new infrastructures. However, current communication technologies lack in enduring effective communication, resource management, privacy, and security challenges [58], [59]. Consequently, these challenges make effective IoT deployment a complex task. Therefore, addressing these challenges to effectively reap benefits from the vital IoT infrastructure and implement ubiquitousness, in reality, is long overdue [52].

A. Motivation of This Survey

The motivation of this survey comes from the realization of the immense increase in IoT devices and their impact on human life. According to the Gartner survey, the IoT-enabled infrastructure will grow to 21 billion connected devices in 2020, depicting an estimated 82% increase as compared to the Gartner's prediction of 2018. Furthermore, there will be around 250 million smart vehicles on the road until 2020 [60]. Moreover, IoT has been involved in generating a significant return on investment where the revenue of IoT service providers, vendors, and solution developers is expected to hike up-to \$1 trillion in 2025 [61]. The IoT-enabled smart homes concept has been transformed into a reality where everything performs sophisticated measurements and produces data giving rise to data generating sources. As most IoT generates personal and sophisticated data, it is infeasible to send all the data to the remote data centers for the processing, which causes security and privacy issues. Moreover, transferring all the data to the remote data centers may overload the communication infrastructure. Edge computing provides a solution for such challenges where the computation of the resource-limited devices can be offloaded at the edge, which alleviates the challenge of traffic overload and privacy concerns. Edge computing provides optimal solutions for battery-constrained devices and latency-sensitive applications. Due to a rapid and disproportional increase in the IoT infrastructure, the need for smart network management techniques is increased. IoT devices cannot be programmed to handle complex rules and customized traffic forwarding due to memory constraints. Consequently, traditional networking technology suffers from providing feasible solutions in handling the application-specific needs of IoT. Traditional network management paradigms also experience scalability and modularity issues, whereas SDN provides centralized IoT management, resource virtualization, innovation, and programmability.

This survey provides state-of-the-art literature on the communication and service technologies for un-interrupted service orchestration from IoT. The disjoint development of the IoT infrastructure provokes non-standardized solutions that leverage security challenges, interoperability issues, QoS concerns, and management problems [62]. A diverse range of application

TABLE I
A COMPARISON OF PREVIOUS SURVEYS ON IoT SERVICE ORCHESTRATION USING SDN AND EDGE COMPUTING

References	**SDN**	**Edge Computing**	**IoT**	**Virtualization**	**SDIoT-Edge Framework**	**Standardization**	**Edge-IoT Security**	**5G Networks**
Farris et al. [26]	✓		✓	✓			✓	
Jilani et al. [27]		✓	✓					✓
Abbas et al. [28]	✓	✓		✓			✓	✓
Pawami et al. [29]	✓	✓	✓	✓				✓
Farhana et al. [30]	✓		✓	✓				✓
Mouradian et al. [31]	✓	✓	✓	✓				
Salman et al. [32]	✓		✓	✓		✓		✓
Roman et al. [33]	✓	✓		✓			✓	
Baktir et al. [34]	✓	✓	✓	✓		✓		
Mao et al. [35]	✓	✓	✓	✓		✓		✓
Yu et al. [36]		✓	✓					✓
Alessio et al. [37]	✓		✓					
Elazhary [38]		✓	✓			✓		✓
Mukherjee et al. [39]		✓		✓				
Ai et al. [40]		✓	✓					✓
Christos et al. [41]		✓	✓			✓		
Our Survey	✓	✓	✓	✓	✓	✓	✓	✓

domains also invoked multiple technology-specific standards in the IoT ecosystem; however, the underlying service provisioning process is identical [63]. Moreover, IoT suffers from higher security threats because the security solutions deployment in IoT is challenging as compared to traditional networks due to the heterogeneity [64], [65]. Consequently, the data leakage concerns at the intermediate nodes during the data transfer poses higher privacy issues. Attack vectors involving IoT also increase due to the highly vulnerable nature of IoT, which gives rise to various devastating attacks [66]. Due to the challenges mentioned above, we design this survey to provide a comprehensive approach in highlighting and addressing the significant challenges in SDIoT-Edge realization. This survey presents the current models of network virtualization and edge computing in IoT to provide a comprehensive reference to the researchers.

B. Comparison With Related Survey Articles and Contributions of This Survey

Several surveys focusing on different aspects of virtualization and cloud computing for IoT have been conducted during the past few years including edge computing for IoT [27]–[29], [36], fog computing, [37]–[39] virtualization [32], [34], security [26], [33], [67], and programmability [30]. A few research papers address the combined perspective of IoT-Edge and their application areas [52]–[55]. Most of these researches address an individual aspect of IoT-Edge, for example, standardization, virtualization, or security. However, there is a lack of survey publications on the SDIoT-Edge computing ecosystem from a comprehensive perspective, including the architecture, virtualization, requirements, standardization, and security, given that this is a novel paradigm, that lies on the intersection of SDN, IoT, and Edge computing. Table I outlines the most recent surveys on the IoT taxonomy from the perspective of SDN, edge computing, virtualization, security, communication technologies, and architecture. The available studies provide an abstract understanding of the IoT integration with fog computing and SDN; moreover, most of them omit a crucial infrastructure of either IoT, edge computing, or SDN. In [32], authors consider the virtualization of fog computing with IoT; however, they did not consider the edge computing paradigm, which has become crucial for latency-sensitive IoT applications.

This survey outlines the literature where SDN, IoT, and edge computing can collaborate and offer novel services besides complementing existing applications. We propose technological grounds on all the three paradigms in developing a comprehensive architecture where centralized management in SDN, computation offloading in edge computing, and sophisticated measurements in IoT can be integrated efficiently. A comparison of this survey with the already available surveys in this paradigm is presented in Table I, which demonstrates that the available literature lacks in providing a comprehensive study on the SDIoT-Edge ecosystem. Most of the available literature discusses only individual aspects of the SDIoT-Edge ecosystem; however, we perform a holistic study of the literature available on the SDIoT-Edge paradigm. We include a comprehensive discussion on key requirements of SDN, IoT, and edge computing that are critical in envisioning SDIoT-Edge solutions. Furthermore, the standardization issues and security challenges have been extensively discussed to benefit the readers in understanding key vulnerabilities and limitations of the current IoT ecosystem. We present the architecture, requirements, applications, standardization, and security aspects of the SDIoT-Edge framework. We include taxonomies of security vulnerabilities and the possibilities of

TABLE II
LIST OF ACRONYMS AND THEIR EXPLANATION

AmI	Ambient Intelligence	NGSON	Next Generation Overlay Networks
APS	Application Service Provider	NIDS	Network Intrusion Detection System
AR	Augmented Reality	NIST	National Institute of Standards and Technology
AWS	Amazon Web Services	NOS	Network Operating System
BLE	Bluetooth Low Energy	OCI	Open Carrier Interface
BS	Base Stations	ONF	Open Networking Foundation
C-DPI	Control-Data Plane Interface	OS	Operating System
Co	Connected Object	OSGI	Open Service Gateway Initiative
CoAP	Constrained Application Layer Protocol	PKI	Public Key Infrastructure
CoRE	Constrained RESTful Environment	QoS	Quality of Service
CSCC	Cloud Standard Customer Council	RAC	Radio Access Networks
DDoS	Distributed Denial of Service	REST	REpresentational State Transfer
DTLS	Datagram Transport Layer Security	RFC	Request for Comments
EaaS	Edge-as-a-Service	ROLL	Routing Over Low Power and Lossy Networks
EC2	Elastic Compute Cloud	RPL	Routing Protocol for Low Power and Lossy Networks
EPC	Evolved Packet Core	RSU	Roadside Units
ETSI	European Telecommunications Standards Institute	RSUC	Roadside Units Controller
HTTP	Hypertext Transfer Protocol	SDIoT	Software-Defined IoT
I2RS	Interface to Routing Systems	SDIoV	Software-Defined Internet of Vehicles
IETF	Internet Engineering Task Force	SDN	Software-Defined Networking
IIoT	Industrial Internet of Things	SDSec	Software-Defined Security
IKEv2	Internet Key Exchange	SD-WAN	Software-Defined Wide Area Network
IoA	Internet of Everything	SDWN	Software-Defined Wireless Networks
IoMT	Internet of Medical Things	SFC	Service Function Chaining
IoT	Internet of Things	SIoT	Social Internet of Things
IoV	Internet of Vehicles	SLA	Service Level Agreement
IP	Internet Protocol	SOA	Service-Oriented Architecture
IPS	Internet Service Provider	SPEC	Standard Performance Evaluation Corporation
ISO	International Standard Organization	SQL	Structured Query Language
ISOC	Internet Society	TCAM	Ternary Content-Addressable Memory
ITS	Intelligent Transport System	TCP	Transmission Control Protocol
ITU	International Telecommunication Union	TLS	Transport Layer Security
LFA	Link Flooding Attack	Ubicomp	Ubiquitous Computing
LLDP	Link Layer Discovery Protocol	UDP	User Datagram Protocol
LoRa	Long Range	VLAN	Virtual Local Area Network
LPWAN	Low-Power-Wide-Area Network	VNF	Virtual Network Function
LTE	Longterm Evolution	Vo	Virtual Object
M2M	Machine-to-Machine	VR	Virtual Reality
MAC	Media Access Control	VXLAN	Virtual Extensible LAN
MEC	Multi-access Edge Computing	WoT	Web of Things
MiTM	Man in the Middle	WPAN	Wireless Personal Area Networks
NDN	Named Data Networking	WSNs	Wireless Sensor Networks
NFV	Network Function Virtualization	WWAN	Wireless Wide Area Networks

attacks in the SDIoT-Edge ecosystem. Finally, a broad discussion on the issues and challenges have been incorporated to provide the researchers and practitioners an insight into the future research in this paradigm. We explicitly discuss key lessons learned at the end of each section to summarize the insights obtained from the discussion. To the best of our knowledge, this is the first survey that broadly covers major aspects of the SDIoT-Edge ecosystem, from requirements to deployment, standardization, and security. The key contributions of this survey are as follows.

- We present the evolution of SDIoT-Edge by reporting relevant literature on IoT, SDN, and edge computing. Moreover, the key requirements of the diverse underlying technologies are presented that are critical in envisioning the SDIoT-Edge concept.
- We categorize the available literature on SDIoT-Edge and provide a comprehensive taxonomy of the solutions using multiple performance parameters.
- We critically discuss, analyze, and evaluate current standardization efforts in SDIoT-Edge. Moreover, a detailed discussion on the key case studies using SDIoT-Edge is presented.
- We discuss security and privacy issues of SDIoT-Edge and present detailed taxonomies of the most devastating attack challenges that can exploit the vulnerabilities in the current infrastructure.
- Novel open research issues, challenges, limitations, and future research directions are presented that provide a roadmap for future research in SDIoT-Edge.

C. Organization of the Survey

The structure of this paper is given in Fig. 1, whereas Table II describes the acronyms used in this research. This survey is organized as follows. Section II provides the core definition of IoT, SDN, and edge computing, including architecture, working principles, and components. Moreover, it present the architecture of the SDIoT-Edge ecosystem, after a detailed discussion on these platforms. Section III introduces the key requirements of heterogeneous platforms in SDIoT-Edge. Furthermore, Section IV presents a detailed taxonomy of the current SDIoT-Edge literature in different categories. Section V discusses state-of-the-art case studies by employing the SDIoT-Edge architecture, whereas Section VI describes the

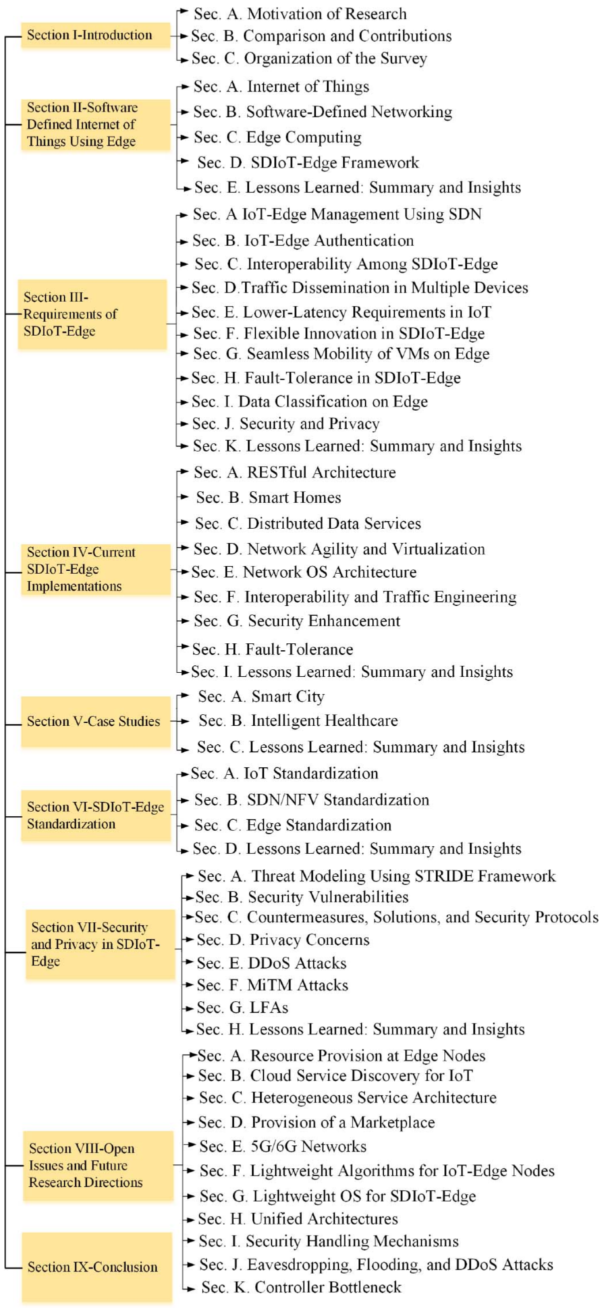
Section I-Introduction
Sec. A. Motivation of Research
Sec. B. Comparison and Contributions
Sec. C. Organization of the Survey
Section II-Software Defined Internet of Things Using Edge
Sec. A. Internet of Things
Sec. B. Software-Defined Networking
Sec. C. Edge Computing
Sec. D. SDIoT-Edge Framework
Sec. E. Lessons Learned: Summary and Insights
Section III-Requirements of SDIoT-Edge
Sec. A IoT-Edge Management Using SDN
Sec. B. IoT-Edge Authentication
Sec. C. Interoperability Among SDIoT-Edge
Sec. D.Traffic Dissemination in Multiple Devices
Sec. E. Lower-Latency Requirements in IoT
Sec. F. Flexible Innovation in SDIoT-Edge
Sec. G. Seamless Mobility of VMs on Edge
Sec. H. Fault-Tolerance in SDIoT-Edge
Sec. I. Data Classification on Edge
Sec. J. Security and Privacy
Sec. K. Lessons Learned: Summary and Insights
Section IV-Current SDIoT-Edge Implementations
Sec. A. RESTful Architecture
Sec. B. Smart Homes
Sec. C. Distributed Data Services
Sec. D. Network Agility and Virtualization
Sec. E. Network OS Architecture
Sec. F. Interoperability and Traffic Engineering
Sec. G. Security Enhancement
Sec. H. Fault-Tolerance
Sec. I. Lessons Learned: Summary and Insights
Section V-Case Studies
Sec. A. Smart City
Sec. B. Intelligent Healthcare
Sec. C. Lessons Learned: Summary and Insights
Section VI-SDIoT-Edge Standardization
Sec. A. IoT Standardization
Sec. B. SDN/NFV Standardization
Sec. C. Edge Standardization
Sec. D. Lessons Learned: Summary and Insights
Section VII-Security and Privacy in SDIoT-Edge
Sec. A. Threat Modeling Using STRIDE Framework
Sec. B. Security Vulnerabilities
Sec. C. Countermeasures, Solutions, and Security Protocols
Sec. D. Privacy Concerns
Sec. E. DDoS Attacks
Sec. F. MiTM Attacks
Sec. G. LFAs
Sec. H. Lessons Learned: Summary and Insights
Section VIII-Open Issues and Future Research Directions
Sec. A. Resource Provision at Edge Nodes
Sec. B. Cloud Service Discovery for IoT
Sec. C. Heterogeneous Service Architecture
Sec. D. Provision of a Marketplace
Sec. E. 5G/6G Networks
Sec. F. Lightweight Algorithms for IoT-Edge Nodes
Sec. G. Lightweight OS for SDIoT-Edge
Sec. H. Unified Architectures
Sec. I. Security Handling Mechanisms
Sec. J. Eavesdropping, Flooding, and DDoS Attacks
Sec. K. Controller Bottleneck
Section IX-Conclusion

Fig. 1. Structure of the paper.

standardization efforts in SDIoT-Edge. Section VII categorizes the security and privacy concerns of SDIoT-Edge including the detailed taxonomies of different attacks. Section VIII illustrates current issues, challenges, limitations, and future research directions, and finally, Section IX concludes the paper.

II. Software-Defined Internet of Things Using Edge Computing

This section describes essential technologies that can be utilized for an efficient implementation of IoT using edge computing and SDN. Furthermore, we present the core technologies that provide the basis for the research on the integration of edge computing and SDN for efficient implementation of IoT. We separately discuss the underlying architectures and provide a detailed framework of SDIoT-Edge.

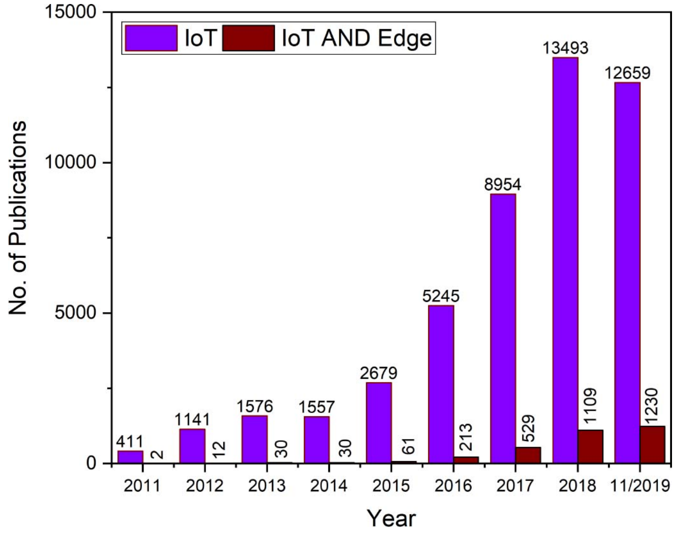

Fig. 2. The number of publications on IoT and its convergence with edge computing [68].

A. Internet of Things

IoT has become one of the most popular terms in research and academia during the past few years [69]–[71]. Fig. 2 encompasses the number of publications in the field of IoT and edge computing from 2011 to November 2019, depicting the growing importance of the research in this area. Fig. 2 shows the IoT and edge computing research trends based on the Scopus bibliographic database; it depicts an enormous increase in the number of publications in both the fields [68]. According to the International Telecommunication Union (ITU), the IoT is an architecture that weaves physical and virtual components together [72]. The IoT definition provided by the Internet Engineering Task Force (IETF) is that it is an Internet that can concurrently operate among TCP/IP and non-TCP/IP protocols, whereas the things are related to the objects that are identified by unique addresses [73]. IEEE provides a comprehensive definition by describing the IoT as a network that interlinks uniquely addressable physical and virtual devices by utilizing novel communication protocols. The things/devices in the IoT are dynamically configurable and provide interfaces that facilitate their access over the Internet [74]. IoT has many variants, as researchers named them during their evolution and invention, such that it is an umbrella term that encompasses many technologies, such as Machine-to-Machine (M2M), Internet of Anything (IoA), Industrial Internet of Things (IIoT), Internet of Medical Things (IoMT), Web of Things (WoT), Social Internet of Things (SIoT), and Internet of Everything (IoE).

The M2M concept encompasses a broad range of networked devices that collect sensor data and send it to the network. It constitutes any technology that enables devices to interact and perform actions in an automated fashion [75]. The most critical component of M2M is the field-deployed wireless

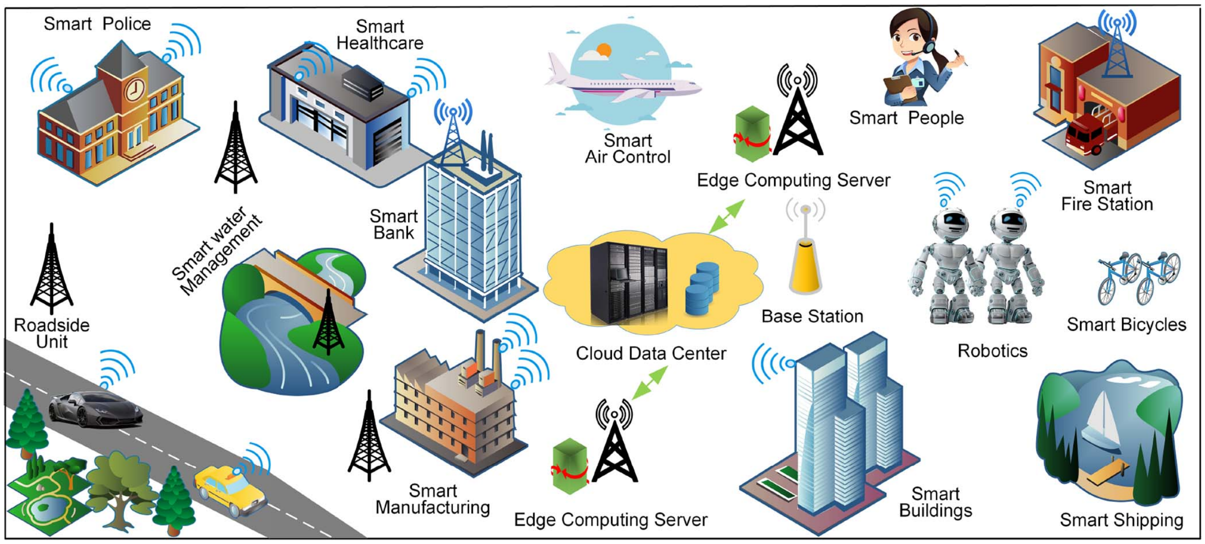

Fig. 3. IoT application areas.

devices having embedded sensors and RFID-wireless communication networks with necessary wireline, including Wi-Fi, ZigBee, WiMAX, Wireless Local Area Network (WLAN), and generic DSL. IoA goes beyond the connectivity of physical objects and includes anything that generates data [76]. Cisco defines IoE as a networked connection of people, processes, data, and things to enhance user experience and smart decision making. IoE creates a compound effect by all-round connectivity, intelligence, and cognition [77]. The major difference that separates IoE from IoA and IoT is that the IoT and IoA may not necessarily contain people; however, IoE does contain individuals. IIoT is the extension of IoT, which enables the use of IoT in the industrial applications [78]. By leveraging the M2M communication, big data, and machine learning, IIoT facilitates the industries to attain better manufacturing efficiency, decision-making capability, and efficient resource management [79]–[81]. Extensive use of IoT in healthcare during the past few years provokes the concept of IoMT, which encompasses networked medical devices and applications that enhance smart healthcare operations [82]. A few applications of IoMT include remote patient monitoring, wearable inspection devices, medication orders tracking, and smart hospital beds. WoT is a refinement of IoT where the smart things are not only connected with the Internet but also with the Web resources [83]. SIoT is defined as the IoT where the things are capable of autonomously creating a social relationship with other objects like humans [84].

IoT studies are composed up of numerous application areas, such as Ubicomp, Ambient Intelligence (AmI), smart homes, and smart cities [85], [86]. Actually, Ubicomp and pervasive computing were proposed before the IoT in the 1980s [87]. Mark Weiser described Ubicomp as a smart environment that is invisibly interconnected using actuators, sensors, and computing objects [42], [88], [89]. This idea pioneered innovation in Internet technology and delved into the development of multiplatform computing.

IoT devices have been employed in many aspects of human life, and their industrial deployment is the most critical aspect because it requires a considerable amount of special effort depending on the environment in which it is being deployed [90], [91]. The most prominent issues faced by IoT deployment are the privacy and security of data [26], [92] and lack of standardization [93]. In this regard, the IIoT consortium has been constituted with the help of many state-of-the-art technology organizations, including Cisco, AT&T, GE, and Intel [94]. The paradigm of IoT covers three areas of broader concepts, including the Internet, things, and semantics [32]. The two most widely used operational architectures of IoT are as follows.

1) *Event-Based Architecture:* In this architecture, the operational data is transferred when a specific event occurs.
2) *Time-Based Architecture:* In this architecture, the data is transferred after a specific time interval.

The characteristics of the IoT include size, space, time, intelligence, everything-as-a-service, and complex systems [32]. The size is an important characteristic, as the IoT is composed up of a massive repository of devices around the globe. Fig. 3 represents the application areas of the IoT; it can be observed that the IoT has been used in almost all fields of life, e.g., industry, health, farming, communication, aviation, transportation, and banking that aim to facilitate human lives. Diverse application areas pose a multitude of domain-specific requirements, including interoperability, communication, and security. In this regard, special attention must be given to reliable IoT communication management. SDN has been widely deployed in current data center networks due to its characteristics of centralized control, efficient resource management, and programmability [95], [96]. There is a high need to use SDN for decentralized IoT network provisioning and management.

B. Software-Defined Networking

SDN has emerged as an essential solution for flexible network deployment and offers efficient network management by providing a centralized view of the network [96]–[100]. With the increasing demands of users due to the broad

Internet access and IoT applications, network developers, service providers, and network carriers have to provide up-to-date services to the users. In the same way, communication networks are expanding exponentially, which makes it complex to manage them. In this regard, SDN has been proposed to enable independence between the controller and data planes, which equips application developers and service providers to proactively manage the network resources and offer flexible network expansion [101], [102]. This separation also provides efficient resource management, where network operators can configure, upgrade, and maintain network resources dynamically [103]. In addition, as the network is logically centralized, the controller has access to all components of the network where resources and traffic can be effectively managed [104].

For the practical envisioning of edge computing in IoT infrastructures, there is a need for a simplified architecture that hides all the complexities of the communication and provides a simplified view to the user. Therefore, SDN, due to its widespread implementation, has become a key candidate for edge service orchestration. SDN is a feasible solution for edge implementation, providing flexibility and high manageability by separating the control and data planes [105], [106]. The control mechanism of SDN can reduce the edge computing architectural and implementation complexities by providing a novel mechanism for networking and allowing efficient resource management simultaneously.

In edge computing, the generated traffic needs to be routed toward the server to complete the device service requirements [35]. As SDN is based on the flexible and intelligent control of the network, it can alleviate complex communication needs at the edge, for example, service discovery, provisioning, and orchestration. In contrast to the traditional networks that rely on the distributed management of network elements, SDN utilizes OpenFlow protocol to flexibly manage the network infrastructure [107]. In traditional networks, the traffic packets are handled by single or multiple combinations of header packets, e.g., MAC address of destination, IP prefixes, multiple combinations of IP addresses, and UDP/TCP port numbers. Alternatively, SDN architecture enables a wide range of packet features by utilizing the Control-Data Plane Interface (C-DPI), a widely adopted example of which is the OpenFlow protocol [96]–[99], [107]. A general-purpose architecture of SDN encompasses three planes, including the data, control, and application plane. The detailed description of SDN planes is discussed below.

1) Data Plane: It is the lowest plane in the SDN architecture that directly deals with the physical network infrastructure, including switches, routers, and access points. The controller manages these devices using the C-DPI. The network infrastructure and the controllers coordinate using a secure channel, implementing different security protocols such as Transport Layer Security (TLS). OpenFlow is a highly adopted protocol deployed for C-DPI and used for communication among data plane devices and the controller.

2) Control Plane: It is called the brain of SDN, which manages the whole decision-making process of the network. It consists of a software implementation of one or more than one controller for effective control over the network. It comprises all the arrangements to enable an intercontroller, data plane to the controller, and application plane to the controller communication. Control and functional components are available in the control logic of the controller, where the functional components include a virtualizer and coordinator. The main SDN control logic maps requirements of the network applications to the commands for the data plane elements [108].

3) Application Plane: It consists of multiple user applications that talk to the controller to achieve abstraction for a logically centralized controller to make coordinated decisions. The Application Plane to Control Plane communication Interface (A-CPI) uses the REpresentational State Transfer (REST) Application Programming Interface (API) [109].

The traffic is governed by the central controller, which has global view of the network and uses the following flow rule installation modes.

- *Proactive Mode:* In this mode, flow rules are configured in the data plane switches before the arrival of data packets. In this situation, when a packet arrives at the switch, it already contains the information on how to process this flow, which limits the involvement of the controller and results in a faster communication.
- *Reactive Mode:* In this mode, when a new flow arrives at the data plane switch, it performs flow rule lookup in their corresponding flow tables. If no match is found, the switch forwards this flow to the controller in a PACKET_IN message. Subsequently, the controller allocates a flow rule based on the network policies and sends it to the switch using a PACKET_OUT message. The incoming rules in the future will be handled according to the flow rule matching process by the switch.
- *Hybrid Mode:* In this mode, the controller has the advantage of utilizing both proactive and reactive modes. The phenomenon behind this is that the administrator sometimes installs proactive rules in devices that the controller modifies reactively to enhance optimal flow, in addition to installing new flows based on network traffic.

In SDN, OpenFlow switches consist of three main components, including flow table, OpenFlow protocol, and secure channel. The OpenFlow-enabled switches keep a variety of flow tables to store forwarding rules to manage network traffic. Each flow rule has three components, including a "rule" attribute, an "action" field, and a "status." The "rule" attribute is used to describe the flow information based on specific header features, for example, source to destination delivery. The field of "action" constitutes the forwarding information on receiving a specific rule, whereas the "status" entry shows the current status of the flow. The secure channel provides an interface used by the controller to communicate with the data plane devices to govern the network and transfer packets [110]. The controller accepts PACKET_IN messages from the switches for the new flows, assigns them a rule according to the values in the header, and sends a PACKET_OUT message back to the switch.

C. Edge Computing

The hierarchy of the edge infrastructure in the IoT paradigm is shown in Fig. 4. This figure shows that the edge cloudlets

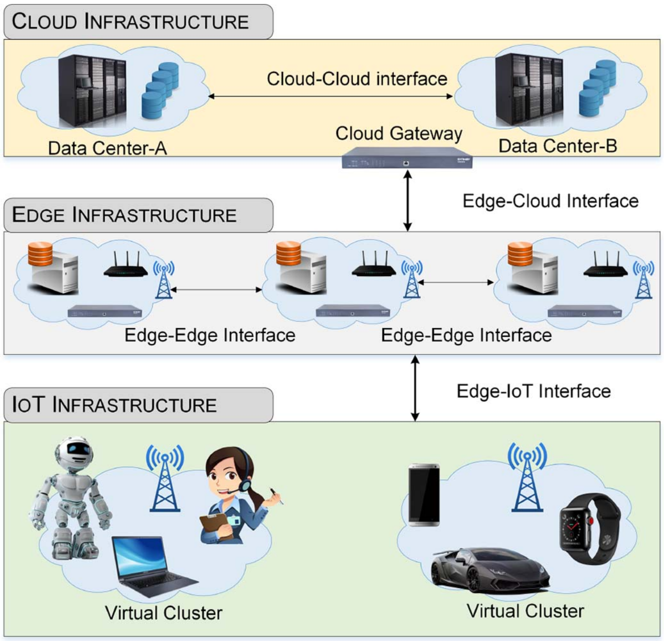

Fig. 4. Hierarchy of edge and cloud infrastructure.

are placed between the cloud data center and IoT infrastructure, which provides intermediate offloading capabilities to the IoT devices [27], [111], [112]. The computation tasks that need higher computational resources can be transferred to the cloud infrastructure through edge devices [113]. The cloud infrastructure contains sufficient computational and storage resources to perform the required tasks.

Edge-enabled devices like smartwatches, phones, and health bracelets have been widely introduced in the market during the past several years. These novel devices continuously create data logs, implement numerous services, and produce and transmit data to the network. However, most IoT devices are yet limited in computation capabilities and continuously need intermediate computation capabilities outside IoT. The traditional cloud suffers in this situation due to the latency requirements of most of the IoT applications. Moreover, these devices have real-time requirements and Quality of Service (QoS) constraints that must be addressed by the computation platform. In the presence of the traditional cloud, a solution has been put forward that leverages the cloud services and brings them closer to the devices, known as edge computing [52], [53]. In the massive proliferation of IoT devices, some computation can be performed at the edge rather than transferring the whole task to the remote central cloud, which increases latency. However, edge implementation in a distributed environment involves considerable complexities of mobility management, device authentication, fault-tolerance, and data management, which can be managed by resource virtualization. In this regard, we discuss edge enabling technologies, including Network Function Virtualization (NFV), in the following.

1) Convergence of NFV and Edge Computing: Network function virtualization deals with the transformation of hardware-oriented functions, such as firewalls or DNS to the software applications. It is an essential enabling technology for dynamic service orchestration in IoT networks. In traditional networks, the network functions are provided as proprietary services [114]. These network functions are attached to a sequenced chain known as Service Function Chaining (SFC) [115], where the data traffic needs to follow the specific SFC order to orchestrate a service [116]. For example, a service provided by SFC is decomposed into the firewall, Deep Packet Inspection (DPI), and load balancer. The network packets are forced to traverse these functions to effectively accomplish the service. However, in current networks, the packets cannot follow the strict requirements of SFC, where novel service provisioning techniques may require the deployment of certain middle-boxes. Due to the heterogeneous nature of current networks and similar underlying nature of SDN and NFV, they can be integrated to separate the network management function from hardware to the software. Due to the virtualization capabilities, NFV enables efficient network services deployment at heterogeneous locations without using high-cost hardware to fulfill SFC requirements. NFV can operate as a service orchestrator in the programmable SDN paradigm where SDN automates the service chaining by installing customized flow rules at the forwarding stations [117], [118]. Seamless integration of SDN and NFV enhances network services where the network functions are implemented as software deployed over servers. Implementing SDN and NFV over the IoT-Edge will enhance the performance of SFC in latency-critical applications and reduce the overhead of long-haul transmission delays [119].

A minimal cloud application platform is necessary for the availability of computing, i.e., processing power, networking, and storage to the edge nodes to support IoT applications [120]. In this situation, SDN and NFV technologies provide network services at the edge [32]. NFV brings computing infrastructure near the edge of the devices for data-intensive and low-latency applications. NFV replaces traditional high-cost, vendor-specific, and specialized hardware capable enough to provide services on low-cost devices for the data-intensive and low-latency applications [121]. NFVs are flexible, which can be launched and terminated on demand. In the same way, SDN is an optimal match for NFV from the edge to the network [5]. SDN is capable of simultaneously reducing the cost and enhances the programmability and flexibility of NFV due to the separation of control and data planes. SDN and NFV are complementary resources for the effective implementation of edge computing. The control-data plane separation enables ease of compatibility of NFV with the existing solutions. NFV provides the necessary infrastructure for SDN, over which it can operate. The convergence of SDN and NFV for edge computing brings novel research directions for innovation toward cost-effective and fast services and application deployment [122]. In the future, edge implementation will formulate a general perspective where the stakeholders might incorporate Application Service Providers (ASPs), Internet Service Providers (ISPs), software, and device vendors. The convergence of NFV and SDN implies the upcoming fifth-generation (5G) networking paradigm and a trend toward flexible software implementation. The 5G networking has been

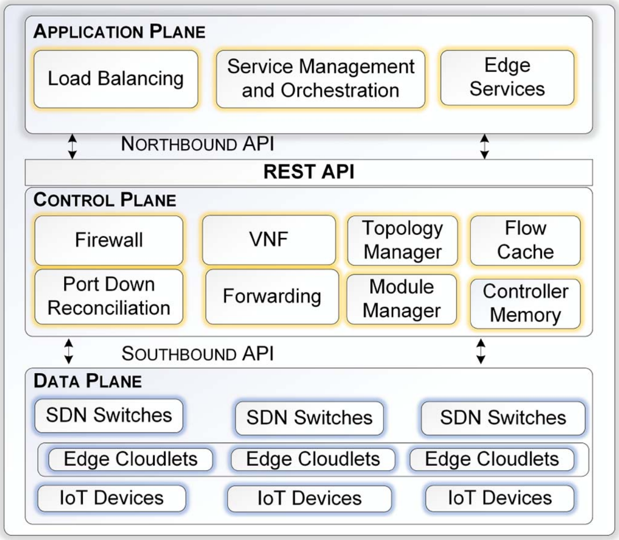

Fig. 5. High-level architecture of SDIoT-Edge.

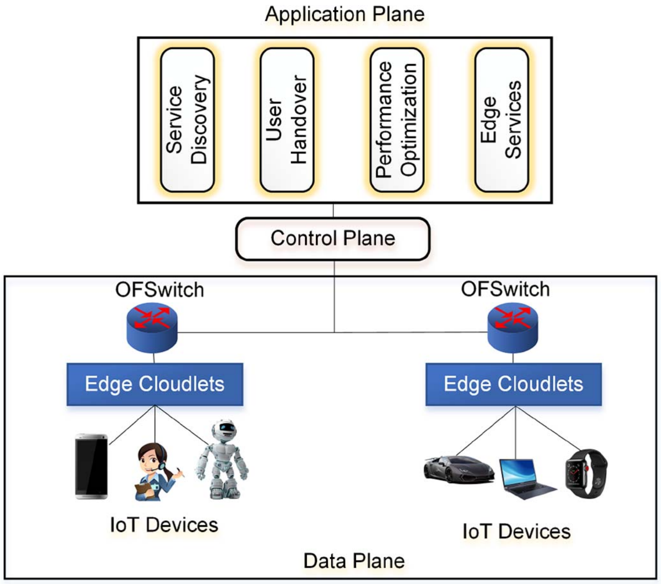

Fig. 6. The components of SDIoT-Edge, redrawn from [34].

launched with fast video traffic, big data service capabilities, IoT processing, and providing wide adoption of Virtual Reality (VR) applications [123]. The concept of network softwarization expresses the fact that all components of the network are managed by the software. Therefore, enabling network slicing into different logical units is imperative, where each slice performs a different set of functionalities.

2) Dynamic Orchestration: Edge computing and its realization in IoT is currently in its evolutionary phase and suffers from many challenges. One of the critical issues is the effective cloud orchestration to monitor, select, control, and deploy the requirements of hardware and software resources for application delivery [124]. However, the challenge is to provide orchestration facilities for IoT-Edge open-source cloud solutions [125], and commercially available providers such as Elastic Compute Cloud (EC2) do not provide the functionality for IoT-Edge applications [18]. They still rely on the simple analysis methods to assign the requests, which are prone to errors for a sophisticated set of cloud services. Additionally, most of the methods for service orchestration are customized for specialized applications and are not ready for edge cloud because, in IoT-Edge, multiple IoT applications are deployed in a shared edge cloudlet. There is still a need for the tools and abstractions for distributed device management to optimize the allocation of resources and fulfill IoT application demands.

3) Dynamic Offloading: In edge computing, the orchestrator needs to continuously cooperate with IoT to handle the offloading tasks and flexibly commit required resources [126]. The technology also lacks in providing a proper framework for configuration, in addition to integration techniques to optimally offload IoT applications and manage them on edge. In this context, a virtual machine (VM)-oriented offloading framework can be deployed and easily managed [28]. However, there is also a need to manage these VMs effectively for specific IoT applications committed to the cloud. For such an implementation, the solution must integrate NFV and SDN-supported edge platforms for resource management.

4) Home Cloud: This is one of the typical examples of SDN and NFV-enabled automated orchestration for dynamic offloading [3], [127]. The home cloud proposes an open framework for automated IoT applications for future edge networks. It configures an efficient NFV and SDN architecture for edge cloudlets for effective service orchestration at the edge and supports dynamic offloading for IoT. Innovative cloud services use proprietary protocols that are closed, private, time-consuming, and are available in customized designs [58]. Similarly, these techniques are not portable on different platforms. In such a home cloud orchestration framework, a specific northbound API has been provided to the application developers [27]. They use Service Level Agreements (SLAs) and transform them into deliverable objects that can be parsed to machine recognizable techniques for the allocation of resources, configuration, management, and control. Finally, the home cloud delivers core services that act as intermediaries between edge service providers and application developers to manage edge-based IoT applications for efficient service delivery. This mechanism will provide characteristics of dynamic allocation, portability, and high scalability that are not currently available in any cloud environment.

D. A Framework of Software-Defined Internet of Things Using Edge Computing

An SDIoT-Edge architecture has been proposed in the literature, which deals with the IoT service orchestration issues using edge computing [52]–[55]. Fig. 5 shows the architecture of SDIoT-Edge encompassing three planes including SDN data, control, and an application plane [34]. In the data plane, IoT devices seek services for offloading the compute-intensive tasks. In the traditional SDN architecture, these planes reside at two different levels; the novelty of this architecture comes from the northbound application plane, where customized northbound applications reside. These virtualized applications decide the behavior of the control mechanism, including end-to-end service orchestration [128]. Fig. 6 shows that the edge cloudlets are used to connect IoT devices with the data plane infrastructure. The SDN controller manages all the components of the network, whereas the application plane incorporates the edge services to efficiently manage the SDIoT-Edge infrastructure. In SDN, the controller hosts different northbound applications to fulfill the required functions and present a singular model to the requesting applications. Every application belonging to edge service orchestration uses northbound API and triggers events. In response to these events, service requests are provided by the controller in terms of commands [129]. Subsequently, these commands are compiled and transformed to low-level OpenFlow messages by the controller, which are then passed to the switches to service the requests.

The motivation of SDIoT-Edge has been instigated by the resource-limited nature of IoT devices. As the number of IoT devices is increasing exponentially with time, SDN becomes a crucial management technology for such a huge network. SDN uses centralized network management to guide network traffic from the source to the destination. Hence, IoT devices can be efficiently managed using SDN, whereas the edge computing can provide offloading services near the resource-limited IoT devices, which require a constant interaction among the IoT and edge infrastructure [52]–[55]. The SDIoT-Edge architecture needs the following integral services to optimally orchestrate the operational requirements of these versatile architectures.

1) Service Discovery: It is possible that different IoT devices have specialized functionalities and require various services from the network. Moreover, they may have no knowledge about the available edge services [19]. For example, there is a need for an environment where IoT devices can make a request by identifying the required computation power and storage [130]. This necessity is implemented by SDN, where the service discovery module acquires information of available services at the edge.

2) Service Provisioning and Migration: This module provides services onto the edge cloudlets based on different performance parameters. If a service has low utilization, this module decides on migrating the VM that hosts this service to another cloudlet.

3) Performance Tuning: The implementation of edge computing has been triggered by real-time application needs [131], [132]. This module utilizes SDN and OpenFlow for managing service utilization to balance or administer the load on various servers.

4) User Handover: In a diverse IoT network, devices may leave and enter edge cloudlets at runtime, causing service disruptions due to the unavailability of handover mechanisms [133]. This module provides forecasting information for the next coverage areas and enables mechanisms for seamless services provisioning.

The flow rule installation mechanism on the switches has been shown in Fig. 7. The figure shows that IoT initially requests the edge gateway, which then communicates with the data plane switch to fulfill the service request by using the proactive, reactive, and hybrid flow rule installation methods. The communication schemes in the SDIoT-Edge have been indicated with different colors.

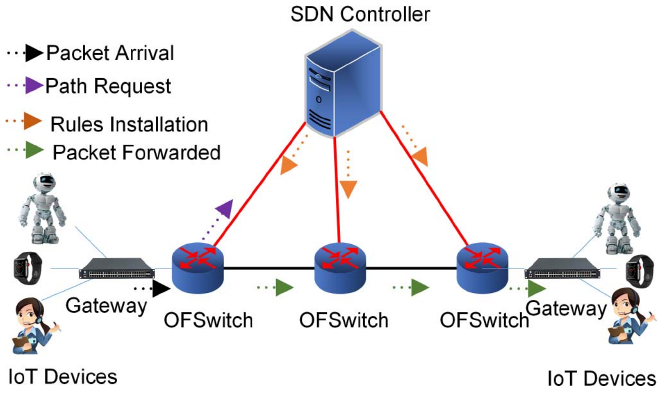

Fig. 7. Flow rule installation in an SDIoT-Edge network.

A detailed SDIoT-Edge architecture is presented in Fig. 8, which shows data plane, control plane, southbound communication interface, northbound RESTful API, and application plane. The data plane contains the resource-limited IoT devices, which utilize the edge cloudlets to offload the compute-intensive tasks. It communicates with the control plane using the southbound interface, which employs OpenFlow protocol. Moreover, the control plane uses the northbound interface, which employs REST APIs to interact with the application plane. Network programmers can develop customized applications to desirably control the network traffic. The Base Station (BS) provides a communication interface to the edge cloudlets and IoT devices. Moreover, edge cloudlets support compute-intensive tasks on the resource-limited IoT devices. The edge cloudlets comprise of low-capacity servers, which fulfill latency-sensitive service requests from the IoT devices. Alternatively, the cloud data center contains sufficient resources to support compute-intensive tasks at the cost of higher latency. Here, SDN provides scalability for the efficient management and deployment of services compared to the traditional approaches. For example, the group table specification facility is available in the OpenFlow versions 1.1 and further [110]. This implies multiple flows to be addressed by the same group identifier, which enables the group table entry to be inserted for multiple flows. This facilitates the process of updating a set of flows by only updating the referred group entry, compared to updating every single flow rule. Another critical factor is that it provides the support for incorporating multiple controllers associated with version 1.2 and above [134]. It enables the switch to communicate with the controllers as a single entity, although multiple controllers are used that may act as a master, slave, or have equal roles.

E. Lessons Learned: Summary and Insights

This section provided an architecture of SDIoT-Edge that supports the resource-limitation problem in IoT and offers centralized management of the underlying heterogeneous architecture. Although the diverse SDIoT-Edge integration seems

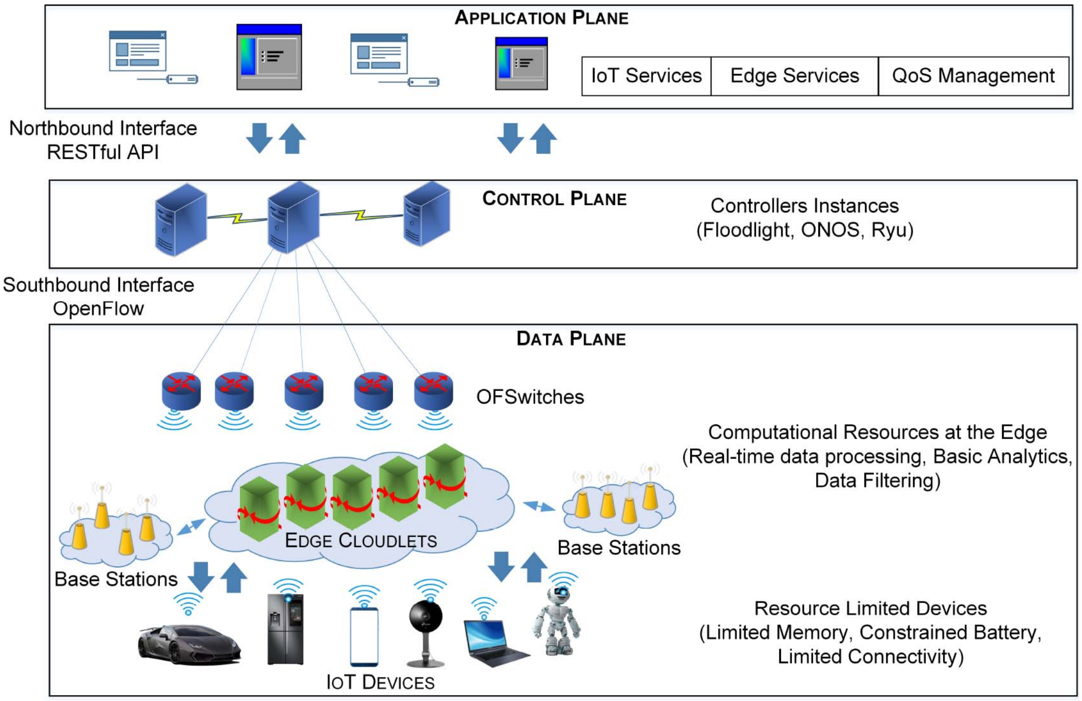

Fig. 8. Detailed SDIoT-Edge architecture composed of three planes.

promising for IoT service orchestration; it does raise considerable concerns of security, privacy, scalability, fault-tolerance, standardization, and interoperability that must be addressed prior to its realization. A large number of flows and massive amounts of data in IoT can congest the network and increase the cost of transporting data from IoT devices to the cloud data center. Edge analytics helps in data collection and analysis at the sensor node, IoT infrastructure, network switch, or any other device avoiding unnecessary data transfer to the central cloud [79]. Edge analytics can filter the data at the devices' edge and transfer the necessary computation to the central cloud. Moreover, the challenges of limited storage, energy, computational resources, mobility, handover, and QoS management induce complexities in the realization of SDIoT-Edge. User hand over can be handled by cloud access optimization algorithms, originally developed for the central cloud, which determines the optimal location to migrate the services [135]. Scalability issues may arise due to the increase in the network size where the network services can be virtualized and deployed as separate applications at the application plane of SDN.

Any solution for SDIoT-Edge should consider the hybrid flow rule installation strategy to address the real-time needs of heterogeneous IoT infrastructure. Moreover, a promising solution would be the deployment of an in-band controller strategy where the control and data traffic share the same link. This technique reduces cost as it is expensive to provide separate links (out of band controller) for data and control for a large number of switches in SDIoT-Edge [102], [136]. The traditional client-server communication style needs to be replaced with a three-tier architecture having a specific set of coordination and orchestration features. In this paradigm, an intermediate network layer can provide a communication interface among edge resources. The application plane in SDIoT-Edge can be utilized to deploy novel service requirements instigated by the edge infrastructure, including service orchestration, security, and resource management. Additionally, IoT devices need continuous offloading services, where the sequence of committing the task to the edge and re-establishment of the connection, must be handled seamlessly to address the service level requirements. Moreover, the control channel in SDN can become a bottleneck due to centralized control. The distributed control paradigm can be deployed to address this challenge; however, it raises multiple other challenges, including communication delay, controller-state synchronization, and security. Hence, the effective realization of SDIoT-Edge depends on the fulfillment of heterogeneous requirements of multiple architectures.

III. REQUIREMENTS OF SOFTWARE-DEFINED INTERNET OF THINGS AND EDGE COMPUTING ECOSYSTEM

In this section, we discuss the requirements of SDN, IoT, and edge infrastructures that are critical in an effective realization of the SDIoT-Edge paradigm. SDIoT-Edge employs the virtualization of network resources to provide services to heterogeneous devices. The transformation of hardware-based solutions toward software gives rise to low-cost IoT applications. In this regard, the requirements in the SDIoT-Edge paradigm must be effectively envisioned before any actual implementation. The challenge of resource limitations in IoT devices can be effectively managed by bringing resources closer to the edge devices using the centralized control mechanisms of SDN. Each component of SDIoT-Edge encompasses

TABLE III
A Detailed Taxonomy of the Requirements of SDIoT-Edge

Requirements	Challenges	Solutions
IoT-Edge Management using SDN [34], [137], [138]	• Traffic forwarding issues. • Dealing with network delays. • Heterogeneous network traversal by IoT traffic. • Load balancing requirements.	Standardization of A-CPI and C-DPI
IoT-Edge Authentication [55], [64], [139]–[141], [143]	• Authentication of multiple devices. • Devices' reliability issues. • Multiple local trust domains. • Credential distribution issues.	Global authentication policies
Interoperability among Heterogeneous SDIoT-Edge Infrastructure [121], [144]	• Heterogeneous product manufacturing. • Vendor-dependent products. • Multiple communication technologies. • Lack of interoperability protocols. • Multi-infrastructure interaction of data.	Virtualization standards for IoT-Edge
Traffic Dissemination in Multiple Devices [51], [145], [146]	• Heterogeneous traffic routing. • Diverse orchestration requirements. • Over utilization of network resources. • Traffic congestion issues.	Customized application development
Lower-Latency Requirements in IoT [147], [148]	• Real-time applications need lower-latency solutions. • Authentication mechanisms are time consuming. • Lack of effective offloading solutions.	Efficient offloading solutions
Flexible Innovation in SDIoT-Edge [26], [34], [149], [150]	• Hardware-oriented traditional solutions. • Lack of virtualization. • Lack of A-CPI standardization.	Standardization of A-CPI
Seamless Mobility of VMs on Edge Infrastructure [135]	• Mobility requirements. • Lack of mobility-aware VM migration. • Real-time VM handling.	Mobility-aware VM migration using NFV
Fault-tolerance in SDIoT-Edge [151]–[153]	• Changes in the network connections. • Wireless channel changes. • Mobility of devices. • Lack of backup channels.	Backup channels
Data Classification on Edge [154], [14], [155]	• Lack of data classification methods at edge. • Aggregate decision making on data. • Distributed nature of data.	Data classification techniques
Security and Privacy [16]	• Lack of global view of data. • Location-based privacy issues. • Platform-specific security policies.	Lightweight authentication solutions.

different requirements; therefore, a comprehensive discussion on the requirements from the perspective of SDN, IoT, and edge computing is necessary. A taxonomy of the requirements in the SDIoT-Edge is presented in Table III, which shows the critical requirements, challenges, and relevant solutions. We discuss these requirements in the following subsections.

A. IoT-Edge Management Using SDN

Network traffic management is one of the core factors to effectively operate the diverse SDIoT-Edge infrastructure [34], [137]. Therefore, high-tech enabling devices are necessary to control, manage, and forward the network traffic flows and to alleviate the impact of network delays. The information in SDIoT-Edge will have to traverse multiple heterogeneous networks, including radio access, backhaul network, and the Internet, where traffic control, routing, load balancing, and other management activities provoke increased traffic delays. Network scalability, manageability, and efficiency requirements can be adequately addressed by SDN-oriented solutions because of the centralized management capability [138]. In this regard, SDN-backed solutions must be optimized to deliver efficient administration, e.g., balancing network load, efficient traffic management, and concise bandwidth exploitation. Moreover, the standardization of A-CPI and C-DPI will provide a solution for the IoT-Edge management issues including traffic forwarding, combating network delays, load balancing, and heterogeneity.

B. IoT-Edge Authentication

The SDIoT-Edge computing concept is based on the interoperability among different platforms using communication protocols, heterogeneous message exchange, and virtualization. These novel features invoke immense authentication issues at the source network. First of all, the traditional trust and authentication mechanisms might become incapable due to heterogeneity in communication infrastructure [64]. Second, the diversity in the communication technologies and the softwarization of the network management will provoke security issues, reliability challenges, and attack vulnerabilities. A unified trust and authentication mechanism is required to ensure the reliability of edge servers and the connected IoT devices [139]. However, traditional cloud authentication techniques are challenging to implement in resource-limited edge servers. Hence, minimizing the security overhead posed by the authentication of the network elements is a critical concern. Diverse communication technologies in SDIoT-Edge encompass different security protocols, which inevitably create their own local trust domains [140]. In this situation, the challenge of the credentials distribution at different locations arises to enable an efficient global trust paradigm. A solution to these challenges can be to devise global authentication policies for heterogeneous networks and infrastructures. The current solutions employ a certification authority that distributes the session keys to authenticate the devices in their own trust domain. However, ensuring the privacy and security between the devices residing at different trust domains is still a challenge [55], [141], [142].

The versatile nature of network elements in the SDIoT-Edge increases the vulnerability of launching an attack on the whole network using a single compromised device. One of the prime concerns at the core network is to enable seamless security at the host network. There are many drawbacks in the existing solutions because any Network Intrusion Detection System (NIDS) will need additional infrastructure to handle a higher level of aggregate data rates [143]. In the same way, network operators are unable to provide a global view of the network, which limits the application context of the security solutions. Moreover, the approaches based on host networks depend on Operating Systems (OS) and can sometimes lead to local optima solutions. Therefore, there is a need for adequate security measures at the network level to avoid malicious activities from adversaries where network-level security solutions will help in ensuring global authentication.

C. Interoperability Among Heterogeneous SDIoT-Edge Infrastructure

Large-scale IoT manufacturing raises interoperability concerns where market-vendors are introducing non-standardized IoT products to generate more revenue. Although it lower downs the infrastructure cost; however, most of the developed products are vendor-dependent and suffer from interoperability issues. A vendor-independent environment is required to overcome the complexities caused by heterogeneous manufacturing. SDN has the potential to overcome vendor-dependency due to the continuous standardization efforts by the Open Networking Foundation (ONF). Therefore, distinct Wireless Sensor Networks (WSNs) and body area networks having different underlying hardware can operate without complications [121]. The heterogeneous infrastructure and underlying communication technologies invoke interoperability issues where network traffic traverses multiple infrastructures. For instance, a lower-latency resource request can be handled by pre-processing it at the edge cloudlets and then transferring the remaining task to the central cloud. In this situation, the data will be traversing two different communication networks. Therefore, a unified management paradigm is required to overcome the communication heterogeneity [144]. There is a lack of interoperability protocols that provide seamless interaction; however, the southbound OpenFlow protocol in SDN is capable of operating among diverse network elements at the data plane. Moreover, SDN and NFV can be deployed to provide mobility-aware VM management, which is capable of alleviating the interoperability concerns. Additionally, standard communication protocols development and virtualization standards for IoT-Edge will further enhance the interoperability in SDIoT-Edge.

D. Traffic Dissemination in Multiple Devices

Traditional networks fulfill the service requests by moving the data to the cloud and then bringing back the results to the device. This workflow generates a massive amount of network traffic [145]. Traffic overhead can be mitigated by keeping the data at the edge. Effective traffic dissemination reduces unnecessary bandwidth utilization, traffic rerouting overhead, and overcoming network congestion caused by billions of devices [146]. IoT generated data can be preprocessed at the edge, which lower downs the excessive computation load at the central cloud. SDIoT-Edge has the ability to solve the congestion problem within the core network and datacenters by traffic distribution at different edge servers [51]. However, orchestration requirements of application-specific request handling to route traffic according to the user's demands require novel traffic dissemination techniques. Customized traffic forwarding applications utilizing the centralized management of SDN can alleviate the traffic dissemination challenges in SDIoT-Edge. The requests about applications should be forwarded by comparing them with the requests received at the intermediate nodes, which will lower the related cost, network load, and traffic delays.

E. Lower-Latency Requirements in IoT

Real-time applications like online gaming, VR, and ultra-high-definition video streaming need extremely high data access rates at lower latency. Therefore, available edge solutions become highly vulnerable in such cases due to the enormous amount of data produced by the loT. Moreover, the use of traditional authentication mechanisms among IoT and edge increases the latency in serving the requests. Traditional offloading solutions also add latency in the overall service orchestration process where efficient offloading solutions need to be developed to fulfill latency requirements in the IoT. For example, Foursquare [156] and Google Now [157] need to provide a real-time response to the users. Lower-latency requirements need reliable solutions where effective backup links must be provided to ensure fault-tolerance. Latency-sensitive applications are in high demand due to extensive progress in IoT where the information is generated and consumed locally, avoiding extra overhead on the network[147]. Applications, such as autonomous cars, industrial robots, and control applications, need a quick response time, which is as low as 10-50ms [148], [158]. Moreover, smart vehicular technology is still in its evolutionary phase, where the communication delay between the request generation and the service provision has not yet been optimally minimized.

F. Flexible Innovation in SDIoT-Edge

Due to the lack of virtualization in traditional networks, bringing innovation becomes a challenging task [149]. These networks are not equipped with A-CPI standardization, which provokes traffic management issues. SDN provides network programmability by decoupling the control and the data plane in the IoT-Edge paradigm and enables dynamic services to a disparate set of devices [34]. The integration of edge servers and IoT devices needs dynamic management. SDIoT-Edge can be managed at variable levels of orchestration by deploying a centralized controller equipped with north-bound applications. In the traditional networking paradigm, the deployment of novel protocols provokes the need for new hardware or redesigning the switch-chips, which induces more cost. For instance, implementing Virtual Extensible LAN (VXLAN) [150], which is a novel cloud and data center protocol, will require upgradation of the whole infrastructure inducing more cost and effort. Instead, software switches like Open vSwitch can be programmed for customized traffic forwarding. Moreover, OpenFlow supports VXLAN, which ensures the implementation of the controller accordingly. Thus, SDN can treat the network as a flexible software [26]. However, flexible innovation requires the standardization of the A-CPI, which will enhance the development and deployment of novel applications in SDIoT-Edge.

G. Seamless Mobility of VMs on Edge Infrastructure

Mobility is one of the key characteristics of SDIoT-Edge applications. When the user moves, the distance among the corresponding servers increases, which deteriorates the user experience [135]. In these applications, the trajectory of the users provides their spatial preferences to the edge servers, which can be leveraged to improve the service orchestration efficiency. Secci *et al.* proposed a method, which links the user mobility with the VM mobility [135]. This method then determines the best location for migrating the service to enhance the user experience. However, it is designed for the central cloud, which needs additional enhancements for the SDIoT-Edge. SDN provides the control and management capabilities to migrate VMs over the edge infrastructure. It is challenging to migrate a VM from one data center to another while the service is running. In the same way, ensuring seamless service provision to IoT devices and addressing the requests simultaneously is one of the key challenges. Therefore, the seamless transportation of VMs without disrupting the services is a core requirement to improve the QoS in data centers. NFV characteristics can be leveraged to perform network slicing and allow mobility-aware VM migration at the edge of the devices.

H. Fault-Tolerance in SDIoT-Edge

The mobility of the devices poses significant challenges in providing reliable services to the users [151]. Computation offloading may suffer because of the frequent changes in the network connections and wireless channels. These changes provoke a catastrophic impact on the latency-critical and compute-intensive applications [152]. For example, augmented reality-based applications aim at providing seamless and exciting virtual experience to the users. Failure of the video streams because of the intermittent network connections may upset the users. Another example is the implementation of SDIoT-Edge in the military, where high-speed and ultra-reliable communication is always required in a high-mobility environment, faults in this scenario can be fatal and bring disastrous consequences. Hence, the design of a fault-tolerant SDIoT-Edge paradigm is required, which can cope with the three major challenges comprising of fault-prevention, fault-detection, and fault-recovery [151]. The fault prevention in SDIoT-Edge can be ensured by employing backup offloading links. Moreover, fault-tolerance can be achieved by micro BSs or central clouds that have larger coverage to provide seamless edge services. However, the challenge is to provide efficient QoS and energy consumption for the backup links and to handle protection clouds for the single-user and multiuser edge computing. Finally, fault-detection deals with information collected by intelligent inspection techniques after defined intervals or using feedback on the provided services. Moreover, time-efficient channel and mobility characterization methods would ensure fault identification. The fault recovery approaches should work in a way that the already running services are not disturbed. Moreover, the affected services can be migrated to the fault-tolerant wireless links equipped with adaptive power distribution mechanisms. The alternative techniques include migrating the processes to neighboring servers by using direct or ad-hoc relay nodes [153], [159].

I. Data Classification on Edge

A massive amount of data generated by smart devices poses a challenge to its effective handling. 5G/6G networks can efficiently address data classification issues by providing real-time data analytics capabilities. Moreover, the distributed nature of data presents challenges on data classification and aggregate decision making. Data need to be classified at edge nodes to save upstream bandwidth where only the mandatory data is transferred to the central cloud. The data is classified based on latency requirements; if the computation is latency-tolerant then it is transferred to the central cloud, whereas the latency-sensitive applications are executed on the edge. In this situation, the classes may correspond to "on-edge" or "on-cloud" computation. The examples of data classification include AR applications where edge servers need to accomplish multiple compute-intensive tasks in a shorter period, such as recognizing users based on their actions using pattern recognition and predicting user requests through machine learning. Data classification may be completed by multiple edge servers where aggregate decision making capability is needed based on data collected at all the servers. Therefore, data reduction and classification techniques are of paramount importance to enable cost-effective solutions [14], [154]. A technique for global data classification is proposed in [155] where the training task is performed at all the local fog cloudlets, and then these local models are aggregated to a global model. However, this solution suffers from the challenges of lower-accuracy and higher computational cost.

Providing a global view of data will facilitate collaborative decision making, reducing upstream bandwidth, and overhead of transferring computation to and from the central cloud. There is a need to devise techniques on edge nodes that can ensure the privacy of the data where a privacy-preserved copy of the data is submitted back to the IoT devices on request.

J. Security and Privacy

Securing network infrastructure is one of the prime concerns in a heterogeneous environment where multiple users, devices, and vendors are participating on a single platform [16]. In the same way, IoT service orchestration involves data transfer to the multiple concerned parties. The network infrastructure is owned by different vendors; so, the control of such devices should be assigned to relevant hosts providing services. The data is generated by IoT infrastructure and transferred to the local edge servers, which limit the global view of data to devise security mechanisms. Data interaction at different platforms raises location-based privacy issues where the location can reveal the identity of the data originator. Similarly, service providers must abide by the access of big data that originated from encrypted resources [160]. Service orchestration in IoT systems is based on diverse infrastructures integration. Therefore, different device authorities will have their own policies to access the data from a different perspective, which raises privacy concerns. In this regard, lightweight authentication solutions are needed to deal with the privacy issues at distributed IoT-Edge environments.

K. Lessons Learned: Summary and Insights

In this section, we provided the critical requirements of using SDN in efficiently managing and maintaining IoT infrastructure using edge services. Edge computing platforms are the core components for future IoT development and integration. Some examples are home robotics, smart cities, smart homes, VR, autonomous cars, crowdsourcing, and M2M communications. The implementation of SDIoT-Edge needs immense technical considerations of the diverse underlying platforms. Compared with the servers in data centers, IoT devices possess limited resources that pose limitations on executing complex computations. In this regard, compute-intensive tasks can be offloaded to the edge nodes that will save energy on IoT devices and help in performing the tasks on a remote resource. However, communication protocols need to be developed to enable interoperability among different technologies.

Resource limitations in IoT need network-level security solutions, where additional hardware/inspection techniques are required at the data plane of the SDIoT-Edge to perform the network surveillance. Additionally, security applications can be developed and deployed at the application plane of SDN. Many open-source controllers provide the facility for the application development such as Floodlight, which is an open-source controller that provides the facility to gather network statistics using JSON-based REST API. Data centers have been continuously equipped with immense resources that can fulfill the needs of IoT. However, as the number of IoT devices is increasing enormously, it is infeasible to process all the data at the data center. Therefore, data classification strategies will play a significant role in future IoT-Edge networks. The companies employing SDIoT-Edge can set parameters on deciding, which information needs to be transferred to the cloud or on-premise data store. Therefore, future research can be directed in developing novel low-complexity data reduction algorithms to address the IoT big data challenges. Hardware development for powerful edge nodes can also reduce the load from the central data centers; moreover, this can also facilitate in addressing the latency-sensitive IoT requirements. Therefore, an effective realization of SDIoT-Edge needs to consider the requirements of SDN, IoT, and edge computing. The efficient realization of these requirements can immensely improve the performance of the IoT infrastructure. The available SDIoT-Edge implementations provide a baseline to develop efficient service orchestration frameworks. We discuss the available solutions in the SDIoT-Edge paradigm in the next section.

IV. Current Software-Defined Internet of Things and Edge Computing Implementations

The decentralized management in traditional networks makes them inadaptable to the requirements of novel IoT infrastructure. The novel network technologies such as SDN can manage the network complexities efficiently and allow the development of applications and services that meet the demands of IoT [96]. SDN enables the flexible configuration of the data plane devices and flow management, whereas NFV provides virtualization of the resources. The potential requirements of the IoT-Edge include QoS guarantee, service layer provisioning, and big data management [161], [162]. We categorize the prevailing literature on SDIoT-Edge into multiple classes based on different features. A comparison of the current literature on the SDIoT-Edge environment is provided in Table IV. We categorize the literature based on the underlying architectural characteristics, including OS, communication between the planes, distributed data handling, traffic management, and fault-tolerance. We discuss all the categories below.

A. RESTful SDIoT Architecture

REST API provides a communication mechanism in the application and control plane in SDN. The REST communication concept has been extended to provide communication among different layers of SDIoT-Edge [163]–[165]. A RESTful SDIoT architecture that accompanies multiple

TABLE IV
A TAXONOMY OF THE PREVIOUS LITERATURE ON SDIOT-EDGE

Research Category	Approach Used	Scope	Architecture
RESTful SDIoT Architecture [163], [164], [165].	• RESTful Architecture. • North and southbound REST API. • Centralized controller equipped with processor and storage.	• SDN-oriented architecture with processor and a storage. • Cloud support for providing virtualization services. • REST APIs for north and southbound communication.	SDIoT
Smart Homes [166]–[170].	• IoT architecture for smart home management. • Virtualization services using centralized controller. • A gateway Open vSwitch for communication.	• Home management using trusted ISP services. • IoT devices service orchestration using computation offloading. • A multilayer SDN controller for granular management. • Data plane include an Open vSwitch to connect all the home devise.	SDIoT-Edge
Distributed Data Services [171]–[174].	• Data-centric approach to provide services. • SDN for virtualization. • Preprocessing of data before transferring.	• Different architectural paradigms for different services. • Including publish/subscribe, layered, and centralized management. • Data classification and filtering before transferring.	SDIoT
Network Agility and Virtualization [175]–[179].	• OpenStack and FlowVisor-based network partitioning. • Preemptive flow rule installation. • Open vSwitches and eNodeB for cloud support. • Edge services for compute-intensive tasks.	• Efficient IoT network management by allocating network resources. • Network slices are assigned to different controllers. • Ease of management and programmability using SDN. • Efficient traffic management by proactive flow rule installation.	SDIoT-Edge
NOS Architecture [180]–[183].	• NOS-based SDN controller support. • SDN-WISE protocol support. • Customized OS for IoT. • Support security through add-ons. • Connectivity using bluetooth, Wi-Fi, and IEEE 802.15.4.	• NOS extension for heterogeneous IoT management. • NOS having low computation power, memory, and processing. • Cloud support to offload compute-intensive tasks. • Provides kernel support other features by add-ons. • Provides interoperability, scalability, and connectivity.	SDIoT-Edge
Interoperability and Traffic Engineering [184]–[187].	• The feasibility of incorporating interoperability. • Modularity among devices of different vendors. • SDN and NFV provide efficient IIoT management. • Ubiquitous wireless broadband support by alternative routes. • Reduce packet loss using traffic management.	• IoT agents learn a broader view of the network. • Install flow rules proactively. • Openv Switch connect diverse set of devices. • WSNs and diverse IoT devices are managed using SDIoT. • Better traffic engineering using management planes. • Topology, admission control, and optimization planes support.	SDIoT-Edge
Security Enhancement [26], [139], [188], [189].	• Use of data service, control, and physical planes. • Address manageability, privacy, and scalability. • Separation of concerns provide granular security. • Traffic surveillance can be performed. • Virtualization can provide effective security management.	• High privacy using different controllers. • Controller for security, storage, and control. • Security controller authenticates the network traffic. • Security management using the Diffie-Hellman algorithm [192]. • Security management on top of underlying web-based architecture.	SDIoT
Fault-tolerance [151], [152], [191]–[193].	• Avoid faults in service orchestration. • Application isolation in data transport. • Fault detection using differential resource allocation. • Reliable distributed data storage. • Authentication of IoT devices.	• Fault-tolerance using backup links. • Slight increase in overhead. • Reliable service selection in VANETs. • Trust-based service provisioning. • Reliable data transportation to and from edge.	SDIoT

modules, such as API for the northbound plane, database, processor, and southbound APIs was proposed in [163]. Usually, the southbound interface handles protocols, such as Hypertext Transfer Protocol (HTTP) and Constrained Application Layer Protocol (CoAP) in SDIoT. Most of the RESTful architectures own a processor and a storage database that stores node-level information. The southbound API provides communication interfaces to the control plane and data plane, whereas the northbound REST API connects the controller and application plane. Wen *et al.* proposed a REST framework for efficient IoT implementation using SDN, which contains northbound RESTful API services to communicate with the application plane [163]. A database provides IoT data storage to retain the status information of nodes, topology, and task management. A processor at the control plane ensures node layout, path transmission, and optimization. A southbound REST API is utilized for communication with the data plane devices. It performs data format transformation, parsing, and transmission to and from the control plane.

However, the discussed techniques lack offloading capabilities using the southbound OpenFlow protocol. The edge infrastructure can be placed at the data plane of the RESTful SDIoT architecture to support latency-sensitive IoT applications. These techniques employ a central cloud for request-servicing, which may become ineffective for the IoT applications. Although RESTful SDIoT architectures facilitate virtualization and interoperability at southbound and northbound interfaces, the compute-intensive and resource-limited applications can be efficiently managed at the edge.

B. Smart Homes

The smart home is one of the most important use cases of SDIoT-Edge, where many examples of smart homes have been discussed in the literature [166]–[170]. An architecture based on automating a smart home using IoT has been proposed in [166], where a smart home connects different appliances with the Internet using Majord'Home management framework. They considered the Connected Object (Co) and a Virtual Object (Vo) that are managed by an avatar. The ISP acts as the Majord'Home that provides user object management by virtualization. This research [166] has been extended by a generic framework for any smart IoT perspective [167]. The Co in previous research was extended as an entity that can produce, get, and disseminate data flow in the network. This framework employs one vertical as well as three horizontal planes. Data plane is composed of all the Cos that are able to generate and receive data without the mediation of routing/forwarding procedures. The control plane consists of two sublevels, where the first level is composed of a controller, the second level consists of a CoVo controller, and an application plane is on top of these two planes. The vertical plane is called the management plane, which owns multiple management modules, such as the network manager, application manager, and a Vo manager that are represented in the operation support layer. For the proof of their concept, they tested their platform with two Bob and Lice Majord'Homes, where each of them has one Open vSwitch, which connects all of the home appliances. The CoVo-based ISP controller acts as the gateway of Majord'Home [167]. However, the use of multiple controllers induces synchronization issues and increases latency in transferring state information among all the controllers. Although multiple smart home solutions have been presented in this section, the hierarchical controllers are a soft target for the adversaries. The adversaries can attack the state information between the controllers, which may cause inconsistencies in the synchronization of both the controllers. This strategy may provoke the smart home appliances to malfunction, which may cause life-threatening issues to the smart home users.

C. Distributed Data Services

IoT has been connecting billions of distributed devices all over the world; therefore, a resilient distributed architecture is required for a seamless operation [171]–[174]. An SDN-based IoT architecture employing distributed data services has been proposed by [171]. SDN provides mobility handling, flexibility, and data agility; at the same time, the dynamic digital system manages big data aspects. The paradigm of publish/subscribe is used to provide services to the WSNs. This data-centric approach provides the benefits of using data as an addressable entity because IoT services are based on the dissemination of the collected data. Their architecture consists of three-layered domains, namely, an M2M domain that uses a gateway to connect heterogeneous devices, a network domain that comprises multiple access networks, and an application domain that encompasses applications of IoT. However, these techniques suffer from security and privacy issues where the distributed paradigm makes the device authentication a complicated task. The devices in one domain can be authenticated using the authorization keys. However, authentication issues arise when multiple edge-oriented networks interact. Therefore, a global authentication mechanism is necessary, which can authenticate the keys generated in multiple networks.

D. Network Agility and Virtualization

Network agility corresponds to the programmability of the modern networks, which leverages the separation of data and control planes. It enables customized application development at the application plane to flexibly control the network traffic. When addressing the SDN for IoT, an obvious challenge is to manage the communication among controllers and switches [175], [176], [178], [179]. To efficiently deal with this challenge, a preemptive rule installation mechanism has been proposed by [177]. This work has been further fine-tuned by providing the concept of a software-defined solution for diverse IoT networks [184]. In this research, an IoT controller interacts with the devices using the installed IoT agents. The interaction requests are recorded by the proposed IoT controller to learn a broader view of the underlying network and to compute the forwarding rules that can be installed on the data plane switches. An overlay network is developed above these networks that allow a seamless collaboration among them. Soft Internet is a novel initiative for the future software-defined Internet [176]. It provides connectivity and management in a software-defined way to deal with the complexity and heterogeneity of the future Internet. The authors in [179] propose two virtualization levels, namely, an end-user and network-level virtualization. In the later, devices at the same physical location are considered, and the virtualization process slices the resources into multiple logical functions. Physical resources at the same level are transferred using logical functions, whereas the user level virtualization is performed by having physical resources at different places.

The software-defined infrastructure manager has been proposed in [175], which utilizes OpenStack, a cloud-based controller, and FlowVisor, which is a network controller. The FlowVisor carries out computation resource management, whereas the controller performs a versatile set of operations, such as network resource management, topology information collection, and managing Open vSwitches' update process. The FlowVisor layer is incorporated to allow partitioning of the network and allocating slices to a particular controller. Tadinada [178] described the benefits of using SDN for network agility, dynamicity, and flexibility to overcome the constraints of the traditional networks. They propose the VortiQa (an application development kit) open network director and the VortiQa open network switch by utilizing two SDN implementations for effective IoT deployment. In the first use case, a cloud-based controller manages the Open vSwitch, which acts as an IoT gateway managed by a cloud. In the second use case, an Open vSwitch is placed on eNodeB (LTE radio access for indoor purposes) to offload data from the Evolved Packet Core (EPC) network, to provide better user experience and decrease the operational and capital expenditure. The IoT gateway performs the functions of transmitting data among data plane devices, securing the devices, QoS provisioning, and authorizing the devices to transport data safely among gateways and providing efficient access control. Consequently, the eNodeB Open vSwitch provides separation of planes and manipulates the packets in a way that they are not able to traverse the EPC network. The discussed virtualization techniques rely heavily on the hardware/software network infrastructure, which invokes higher capital costs. These techniques may suffer from interoperability issues due to the heterogeneity and vendor-specific technologies. Moreover, security issues are inevitable due to the data transfer among diverse platforms.

E. Network Operating System Architecture

The heterogeneity is one of the main aspects of IoT. The OS in IoT hides the complexities of heterogeneous components and provides a generalized view to the developers by using network-level protocols, such as IPv6 [194] and 6LowPAN [195]. The main characteristics of IoT-OS should be to provide the ability to connect a massive number of heterogeneous devices [180]–[183]. A Network Operating System (NOS) manages the heterogeneity in the network paradigm and enables the use of different applications for a different set of network devices. In this regard, the authors in [180] proposed an OS for IoT to extend the NOS-based SDN controller, which employs the support of the SDN-WISE protocol that enhances the characteristics of SDN for WSNs. A programmable architecture for SDIoT has been proposed in [181], using three layers of the controller to avoid a single point of failure. TinySDN has been proposed and includes multiple controllers over a WSN [196]. However, the authors do not explain the intercontroller communication and the selection process of the controllers. An SDN-WISE-based network OS has been proposed in [183]. This OS treats the IoT network in a unified way by providing a generic abstraction for the disparate IoT networks. To connect diverse devices and networks, there is a need to develop abstractions for network elements and communication protocols. Therefore, a unified OS can connect versatile networks in SDIoT-Edge. However, the update patching, solution maintenance, and rollback are prime issues in the current OS for low-powered devices. It is challenging to update IoT solutions at remote locations having non-reliable Internet connections and computational resource constraints. The Internet and power failure spoil the update installation where effective rollback mechanisms should be developed. Thus the development of a lightweight OS is necessary for constrained devices where rollback, maintenance, and update patching issues are particularly addressed.

F. Interoperability and Traffic Engineering

As IoT networks are composed of heterogeneous devices, efficient traffic engineering can provide optimum routes and reduce network overhead [184]–[187]. The current literature regarding this paradigm discusses the feasibility of incorporating interoperability among devices of different vendors. The prevailing solutions are equipped with auto-configuration and recognition mechanisms, where switches and gateways dynamically perform management and configuration. The applications, such as virtual sensors, software-defined wireless networks, and virtual cell management are utilized as use cases to implement virtualization, where SDN and NFV provide flexible IoT management [185]. The authors in [186] proposed an IIoT in ubiquitous wireless broadband for better traffic engineering in SDIoT, which includes three management phases in a controller, including topology, admission control, and optimization of location. The centralized management reduces packet loss by incorporating alternative route mechanisms. Although the traffic engineering techniques enforce optimal path selection to control the network traffic, it becomes a single point of failure in the in-band SDN control strategies where a single link is used for the data and control path [102], [136]. Moreover, the available techniques suffer from interoperability issues because of the diversity of communication protocols, each having domain-specific requirements. Therefore, standardized communication protocols are necessary to deal with interoperability issues.

G. Security Enhancement

IoT security is one of the prime concerns due to the unavailability of specialized security mechanisms in IoT devices. The research in IoT security has received immense attention during the past several years [26], [139], [188], [189]. To address the issues of manageability, privacy, and scalability in IoT, an architecture consisting of three planes, namely, data service, control, and a data plane, has been proposed in [188], [189]. The data plane is composed of infrastructure devices, whereas the control plane has been divided into blocks that include Software-Defined Security (SDSec), software-defined controller, IoT controller, and software-defined storage. SDSec performs data authentication and forwards it to the data collector for processing. The data collector then forwards it to the IoT controller, which computes the path to the destination and update rules. The authors in [189] propose a novel SDN-oriented WoT architecture that deals with the limitations of security, data, and things management. They employ Diffie-Hellman method [190] on top of underlying Web-based architecture, which effectively hides the complexities of management and provides security. This architecture consists of three planes, including access, control, and application plane, to offer data access, control, and application services to the underlying network. The current security enhancement strategies are either reactive, or they require higher computational resources to ensure security, which makes them hard to implement in the SDIoT-Edge. However, developing low-cost proactive solutions is challenging in the resource-limited SDIoT-Edge paradigm. Therefore, network security solutions developed as flexibly deployable software at the application plane of SDIoT-Edge can be extremely beneficial [198], [199].

H. Fault-Tolerance

SDIoT-Edge amplifies the service orchestration capabilities where the negligible bugs at a smaller scale or in the testing paradigm (e.g., straggler [200]) might invoke debilitating impact on system reliability. Fault-tolerance is an essential characteristic in SDIoT-Edge due to the intermittent network connectivity, resource limitation, heterogeneity, and harsh deployment environments [151]. It is extremely necessary to ensure end-to-end delivery during data processing and transmission to and from the edge nodes. CEFIoT architecture [151] uses application isolation, data transport, and multi-cluster management to ensure fault-tolerance in IoT applications. The layered architecture of CEFIoT offers compute-placement on edge or cloud without code modifications. Similarly, energy-efficiency and data reliability should be modeled in an integrated manner for the latency-sensitive IoT applications [201]. An energy-efficient distributed data storage method ensuring explicit data-reliability has been proposed in [152]. This technique adaptively reconfigures the system parameters in an energy-efficient way while ensuring continuous reliability. Zhang *et al.* [191] propose a fault-tolerant framework for trust-based service provisioning in vehicular networks using integrated adversarial behavior detection. Fog computing is capable of analyzing and storing related data in vehicular networks while retaining the dynamic trust weights based on attribute parameters of every vehicle. The weights are then incorporated into the proposed service delivery framework, which contains trustworthy vehicle

TABLE V
A DETAILED TAXONOMY OF SDIoT-EDGE SOLUTIONS WITH REFERENCE TO DIFFERENT PERFORMANCE PARAMETERS

Research	Category	Scalability	Security	Offloading	Application Domain	Cloud Domain	Solution/ Architecture	Feature
RESTful Architecture [163]	RESTful SDIoT Architectures	Low	Low	✓	Smart manufacturing	-	Framework	REST services
Adaptive Transmission Optimization [164]		Medium	Low	✓	Smart manufacturing	Fog	Solution	Optimized communication
Autonomic Computation Offloading [165]				✓	General purpose IoT	Edge/Fog	Solution	Computation offloading
Majord'Home [166]	Smart Homes	Low	Low	✓	Home networks	Edge	Framework	Smart homes
HomeCloud Auto configuration [170]		Low	Low	✓	Smart homes	Cloud	Solution	Home appliances
SDLAN for Smart Environments [167]		Low	Medium	✓	Smart environments	Edge	Solution	Centralized control
IoT Big Data Analytics [168]		Medium	Low	✓	Smart homes	Fog	Solution	Data-intensive tasks
Smart Environments in IoT [169]		Low	Low	✓	Smart environments	Fog	Solution	Social IoT
Publish/Subscribe-enabled SDN [171]	Distributed Data Services	High	Low	✓	General purpose IoT	Cloud	Architecture	Distributed control
Cross-layer Access Control [172]		Medium	High	✓	General purpose IoT	-	Solution	Security among layers
PICO Middleware [173]		Medium	Medium	✓	Smart grids	Central cloud	Architecture	Heterogeneous environment
Middlebridge for IoT [174]		Low	Low		WoT	Central cloud	Solution	Application layer middleware
Network virtualization [197]	Network Agility and Virtualization	Low	Low		WSNs	-	Architecture	Network virtualization
Software-Defined Infrastructure [175]		Low	Low		Heterogeneous networks	Central cloud	Solution	Flexible resource provision
Smart Internet Provisioning [176]		Low	Low	✓	Heterogeneous networks	Central cloud	Architecture	Service-aware network control
Scalability for IoT [178]		Medium	Medium	✓	Smart environments	Edge	Architecture	Mobility management
Pre-emptive Flow Installation [177]		Low	Low		Smart environments	-	Solution	Efficient flows
NOS for IoT [180]	NOS architecture	Medium	Low	-	SDN-WISE	-	Solution	NOS for IoT
Programmable Architecture for IoT [181]		High	Medium	✓	SDN	Fog	Solution	Multiple controllers
Unified Control of Sensors [183]		High	Medium	✓	WSNs	Cloud	Solution	Unified control
Empowering IoT using SDN [184]	Interoperability and Traffic Engineering	Low	Low		SDIoT	-	-	Traffic engineering
SDIoT with NFV [185]		Low	Low		SDIoT	-	Architecture	Virtualization in IoT
SDN for IIoT [186]		High	High	✓	Smart manufacturing	Cloud	Architecture	Reliability and security
Containerized IoT Services [187]		Medium	Medium	✓	Multi-access edge	Edge	Solution	Virtualization in IoT
Secure SDIoT [188]	Security Enhancement	Medium	High	✓	SDIoT cloud	Cloud	Architecture	Storage and security
Virtualized Computation in IoT [139]		High	High	✓	5G-enabled IoT	Fog	Solution	Virtualization in IoT
WoT-SDN [189]		High	High	✓	WoT	Cloud	Solution	Scalable and secure WoT
CEFIoT [152]	Fault-Tolerance	Low	High	✓	WSNs	Edge/Central cloud	Architecture	Failure avoidance
Fault-Tolerant Data Storage [151]		High	High	✓	Large-scale networks	Edge	Solution	Fault-tolerant IoT-Edge
Reliable Multi-service Delivery [191]		High	High	✓	Vehicular networks	Fog	Solution	Misbehavior detection
SIoTFog [192]		High	High	✓	Large-scale networks	Fog	Solution	Byzantine resilient IoT-Fog
Fog Computing for IoT [193]		Medium	Medium	✓	Large-scale networks	Fog	Solution	Bloom filtering to authenticate

selection for misbehavior detection. A Byzantine fault-tolerant network to enhance transmission and processing efficiency using resource strategies for IoT-Fog computing has been proposed in [192]. It is a three-tier heterogeneous IoT-Fog model consisting of routers as fog nodes. The breadth-first search and two Byzantine fault-tolerant resource allocation strategies are used to distribute fog node's workload capacities to the requesting IoT users. Authors in [193] use a certification authority to authenticate the IoT devices to ensure fault-tolerance in IoT-Fog. The IoT devices use digital certificates issued by a specific certification authority where a central fog node is responsible for certificate revocation using bloom filtering.

Although the proposed approaches ensure fault-tolerance, they induce more traffic delays during traffic inspection. Every checkpoint adds a latency value to the data transfer; however, avoiding faults is imperative in the current information-centric networks.

Along with the solutions mentioned above, many state-of-the-art edge architectures have also been proposed in the literature. The ENORM framework provides edge computing facilities by bringing the computation resources close to the edge of devices. It is composed of a manager that manages edge nodes, a monitor, and a hardware layer that includes the host OS and applications to communicate with the edge of the devices [202]. The Open Carrier Interface (OCI) is an open-source edge computing framework that provides an abstraction layer for the edge services [203]. This framework offers an interface for the network providers to enable edge computing. It is composed of global and local OCI, a resource manager, and an OCI library. The EdgeX Foundry project was started by the Linux Foundation to develop an edge computing framework [204]. The primary purpose of this project is to provide edge services for IoT ecosystems. This project uses embedded devices such as gateways to support IIoT. An edge computing framework for IoT was proposed by Eurotech [205]. It can provide services to the devices developed using the Open Service Gateway Initiative (OSGi) and modular IoT framework. The Edge-as-a-service framework has been presented for the distributed cloud in [206]. It consists of a discovery platform that identifies edge devices and makes them available for service provisioning, where a service provider facilitates the offloading requests.

I. Lessons Learned: Summary and Insights

Table V classifies the literature on SDIoT-Edge using eight different parameters. The category parameter corresponds to the broad classification of the proposed techniques discussed in this section. The scope and architecture of these techniques are also presented in Table IV. The scalability shows how the solution performs when exposed to higher workloads or an increased number of devices. The IoT infrastructure has been growing tremendously; therefore, scalable solutions are required for the future needs of IoT. The offloading parameter corresponds to the computation offloading capability in the solution. The application domain illustrates the application area, for which the architecture has been proposed. The cloud domain demonstrates the provisioning of cloud infrastructure, including edge, fog, and the central cloud paradigm. In the last two columns, we show that the current research provides a solution or only proposes an architecture, whereas the feature column demonstrates a key aspect of the solution. The best solution for SDIoT-Edge should be highly scalable, provide high security, and should support computation offloading capabilities using the edge infrastructure.

As mentioned in the above discussion, there are numerous proposals for the effective adoption of IoT-Edge infrastructure that have been proposed. However, the improved features and diversity of the devices increase complications and hinder their adoption. Though the provided abstractions are comprehensive for small IoT architectures, there is still a need to develop scalable and secure solutions for the SDIoT-Edge ecosystem. SDN provides the basis to re-visit the deployment of network functions. It promotes the idea of the softwarization of the infrastructure, which supports heterogeneity and dynamicity. The gateway switches managed by SDN can operate as ingress hardware to support the connected IoT infrastructure. SDN enhances IoT network management functions to cater to the challenge of high scalability. Although it provides efficient management, centralized control poses issues of throughput, latency, availability, and single point of failure of the network.

Even though numerous ideas and solutions toward wider implementation of IoT are conceived, the novel complexities of the disparate infrastructure hinder their adoption. It has become relatively easy to develop novel solutions for IoT. However, novel solutions need to consider standardization and existing solutions to enhance interoperability among the prevalent devices. The existing solutions lack in addressing all the critical aspects of IoT to enhance seamless service provisioning and prompt support. A distributed service provisioning solution can be considered to overcome the resource-limitations in the edge cloudlets. However, this may cause extra overhead of latency constraints and the abstraction of a singular cloud. Moreover, the location-awareness must be addressed explicitly because of the mobile nature of IoT devices. Standardized test-beds for IoT experimentation are highly needed, which can provide actual results. Finally, we have discussed multiple proposals to solve the service orchestration problem in IoT; however, there is still a need to develop real solutions that explicitly address these problems.

V. Case Studies

SDIoT-Edge enables efficient resource management using programmability, computation offloading, and dynamic control. In this section, we discuss the use cases of SDIoT-Edge, including smart cities and intelligent healthcare. We select these two case studies because of their higher influence in facilitating human lives where the smart cities enable sustainable living standards, whereas intelligent healthcare ensures a healthy lifestyle using smart healthcare infrastructure.

A. Smart City

The miniaturization of the sensory technologies provoked immense development in the smart cities [158], [207]. Smart cities facilitate the citizens to enjoy a secure, autonomous, and reliable lifestyle using the smart infrastructure. Compute-intensive applications in smart cities produce an immense amount of data that needs latency-aware computation techniques. SDIoT-Edge efficiently manages smart city infrastructure by addressing the requirements of fault-tolerance, latency, security, and reliability. Smart cities can get extensive benefits from SDIoT-Edge including flexible innovation, traffic dissemination on multiple devices, infrastructure management, and interoperability among heterogeneous devices. In this section, we provide two detailed use cases of the smart cities, including Intelligent Transportation Systems (ITS), and smart homes in the following.

1) Intelligent Transportation Systems: IoT enables a novel paradigm of the Internet of Vehicles (IoV) infrastructure, which utilizes edge computing to offer novel services for transportation systems. In an ITS, IoV connects different vehicles with the Roadside Units (RSU) and the other vehicles using sensors and geofencing technologies. IoV leverages the edge cloudlets for service provisioning and orchestration. In this paradigm, SDN enables virtualization of the resources, which transform hardware-oriented services into software-based solutions. Tremendous research has been currently performed on smart vehicles in academia and industry [208]. ITS has been currently used with battery-powered smart vehicles to provide Eco-friendly transportation services. A use case of pay-per-charge has been adopted in Germany, which uses crowdsourcing to charge the battery-powered vehicles [209]. This ITS ecosystem requires the integration of smart charging stations, vehicles, and payment facilities with the cloud. Smart vehicles can assess the nearest charging station and book an appointment on a mutually acceptable charging price. The charging price can be paid using the smart online transaction without a significant interference of the user. In this situation, SDIoT-Edge comes into play to support the whole ecosystem by handling runtime transactions and charging stations' appointments at the edge. These vehicles use the installed sensors to perform most of the tasks autonomously. Edge computing provides the capability to perform surveillance with a 360-degree view of the vehicles from the other vehicles and geofencing boundaries. The sensing capabilities provided by the brake sensors, measurement sensors, and positions sensors enable appropriate decision-making abilities for the controllers to take countermeasures.

Fig. 9 shows an ITS paradigm in a smart city. We can observe the RSUs, which provides the communication services to the driving vehicles. The Roadside Unit Controllers (RSUC) manage the RSUs and are controlled by distributed controllers presenting a logically single view of the SDN controller. The distributed SDN controllers in the ITS provides services of traffic management, edge resource discovery, and mobility management. The BSs shown in the figure facilitate the accessibility to the edge services. Edge computing in this architecture provides efficient low-latency services to the users. The ITS has been equipped with the OpenFlow protocol, which is governed by the distributed controllers at the control plane. The vehicles are equipped with long-distance 5G/6G/WiMax connections for the communication. The ITS ecosystem enables communication with the fueling stations, payment endpoints, and the emergency infrastructure connected by the Internet backbone. A hybrid communication

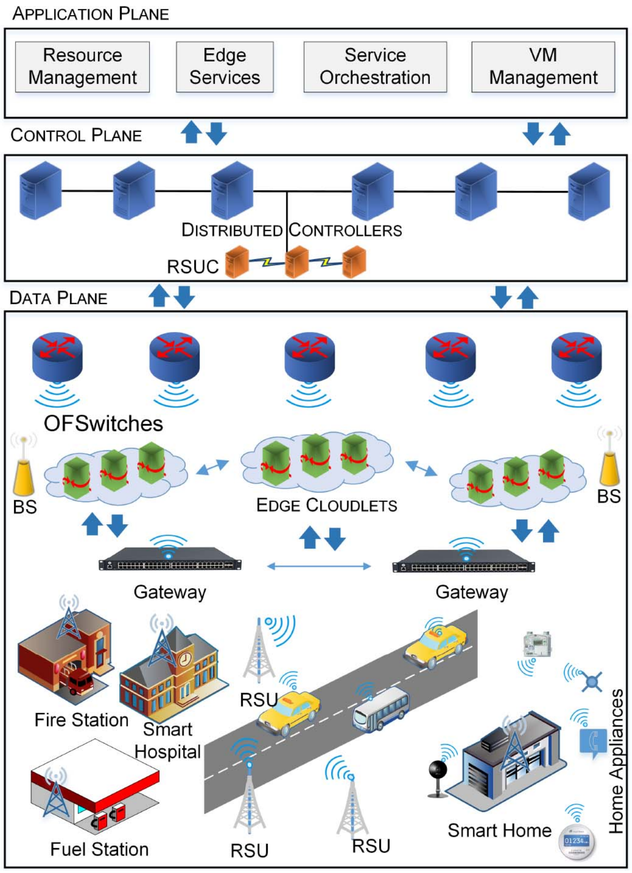

Fig. 9. A smart city paradigm using SDIoT-Edge ecosystem.

procedure has been utilized where the distributed controllers send the abstract rules to the RSUC, whereas the precise control has always been decided by the control plane. The vehicles continuously communicate with each other using status messages as well as with the geofencing and surveillance infrastructure. The controllers maintains a global connectivity graph using the information propagation from the RSUC and the BS, which facilitates a centralized decision-making capability.

Communication in the ITS includes control and data traffic. Edge computing provide low-latency services to the requesting ITS applications. SDN in ITS provides virtualization, whereas BSs and RSUC can handle the hypervisor for VM management, which enables portability and efficient resource utilization. The application plane in the SDN enables mobility-aware VM management. Vehicles in the ITS are continuously moving and need to distribute the offloaded tasks considering the service mobility and topology changes. ITS employs the Service-Oriented Architecture (SOA) for the low-level message exchange. The controller, BS, and RSUCs provide services according to the installed software. For instance, consider a task that needs to be performed in an ITS that may need operations at different endpoints in the system. Here, the SDN controllers manage the services at different service-providing platforms and stores the results. Finally, the control plan incorporates the results of all the subtasks and provides the cumulative results back to the request originator.

2) Smart Homes: The available smart homes' use cases provide the framework for future living in a smart city paradigm. Many companies have been working on prototype solutions for smart homes. The interaction among smart devices in smart homes produces a massive amount of data, and the real-time analytics of the data can provide immense opportunities in realizing smart city concepts [48]. For example, the fire stations of a smart city can monitor the real-time status of the homes and provide rapid response in case of an emergency. Similarly, smart hospitals can monitor the patient's daily routine in smart homes and propose remedial prescriptions to patients. These opportunities enable a significant cost reduction and improved living standards.

Smart homes are usually composed of sensors equipped in the appliances. The categories of components in smart homes can be physical systems, communication, and context-aware components. The physical systems consist of the home appliances equipped with sensors that are capable of capturing and forwarding sophisticated measurements. The communication systems provide connectivity in the smart home ecosystems, whereas the contextual systems provide intelligent decision-making capabilities that have been moderated by the rules provided by the administrators.

Fig. 9 also shows a use case of a smart home in a smart city scenario that uses edge computing infrastructure for offloading the latency-sensitive tasks. The home appliances are connected with the gateway that further connects to the distributed controllers and the edge infrastructure. The smart home is connected to the fire stations and hospitals, which communicate using the wireless infrastructure. Smart home applications continuously generate data, where the analysis of the data provides opportunities for understanding the dynamics of the homes. As smart home appliances do not contain sophisticated data storage capabilities, they need cloud storage services. The edge nodes are capable of processing the data streams continuously from smart homes and providing storage at the edge of the devices. Moreover, it offers processing capabilities in near real-time. The unification of edge computing in smart homes solves the challenge of latency-sensitive computations, which was previously caused by the transport protocol of the cloud [168], [210].

B. Intelligent Healthcare

SDIoT-Edge can be extensively used in the healthcare industry to control diseases, perform skilled diagnoses, and help in executing complex surgical treatments (e.g., spinal surgery) at remote locations. It facilitates sensors to collect medical data and assist in critical decision making [211]. It addresses the critical healthcare requirements of latency, fault-tolerance, traffic dissemination, data classification, and security. It supports the mobility of healthcare and AR infrastructure by providing mobility-aware VM migration. The immense amount of data generated by intelligent healthcare infrastructure can be adequately managed by edge cloudlets. It invokes the AR concept to perform medical treatments at remote locations. Moreover, SDIoT-Edge supports innovation in intelligent healthcare by application development to flexibly manage critical healthcare

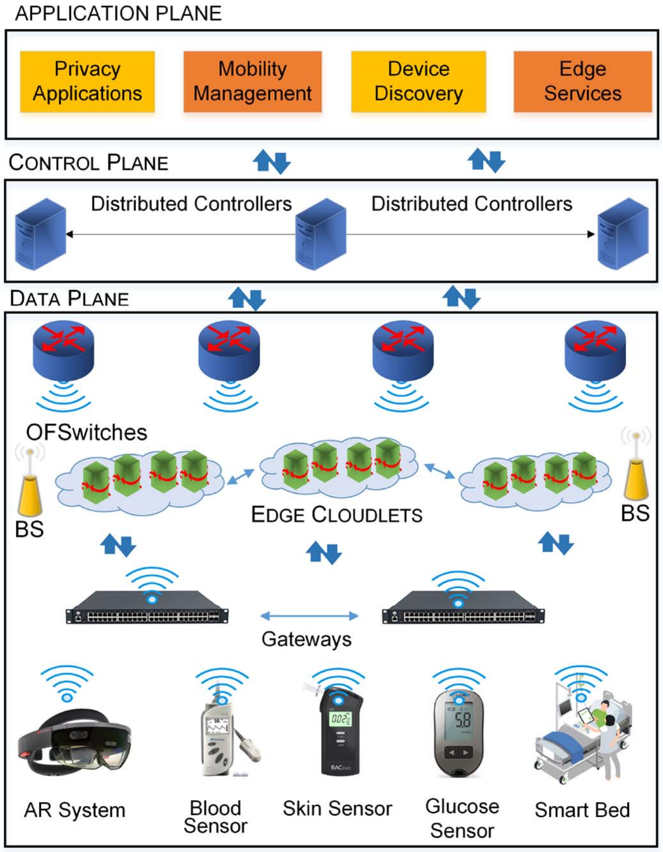

Fig. 10. A smart healthcare paradigm employing medical sensors at the data plane including AR, patient management, and other IoT sensors.

traffic. We provide SDIoT-Edge use cases in healthcare in the following.

1) Smart Health Monitoring Systems: Smart healthcare is envisioned by offering low-cost and effective real-time healthcare facilities ubiquitously. Smart health monitoring is a promising field that lies on the intersection of medical informatics, public healthcare, and cognitive capabilities using the IoT. The SDIoT-Edge-enabled smart healthcare system has been shown in Fig. 10. This architecture supports medical devices to share data using gateways, whereas the edge cloudlets provide offloading services to the resource-limited healthcare devices. The smart health monitoring system uses the concept of ubiquitous computing and ambient intelligence that is augmented in the personalized healthcare systems. Smart health monitoring is associated with the wellbeing of humans by smart decision making based on collected data from biomedical sensors, genomics, wearable sensors, and social media. The data from biomedical sensors is critical, which needs prompt processing to support intelligent decision making. The smart health monitoring systems connect tiny nodes having sensing and actuating capacities embedded inside or outside the human body. As the data generation and processing devices have been immensely increased, the smart healthcare data needs to be retrieved, updated, and transferred efficiently. SDN in healthcare provides heterogeneous infrastructure management where the traffic from a large number of sensors can be efficiently directed using the OpenFlow protocol. Moreover, the network programmability of SDN facilitates data transfer among medical devices manufactured by different vendors. The real-time data processing needs can be handled using edge computing.

In SDIoT-Edge, the sensor nodes record and transmit data to the edge cloudlets using BS. SDN and edge computing can collaborate to execute latency-critical tasks from heterogeneous medical devices. The gateways are used as central hubs between medical devices and the edge cloudlets. SDN and edge computing empower the healthcare network to maintain mobility, scalability, low-latency, and load balancing to develop remote healthcare systems. SDIoT-Edge provides extensive capabilities in smart healthcare for infrastructure and data management. However, because of the critical nature of latency-sensitive data in healthcare, deciding the local and remote execution of tasks is imperative where hierarchical computation partitioning techniques can be employed for decision making. Network resources can be virtualized using SDN and NFV, where the latency-sensitive resource requests can be processed at the edge. Personalized resources can be allocated according to the health status of the patient using intelligent disease analysis. The issue of mobility and scalability arise due to a significant number of mobile medical devices, which can be managed using mobility management and resource discovery services in SDIoT-Edge. There is an immense potential of SDIoT-Edge in the healthcare technology where smart healthcare infrastructure can interact for aggregated decision making, which will enhance the patient monitoring and overall healthcare facilities.

2) Augmented Reality in Healthcare: AR-enabled smart healthcare facilities support users by offering smart medical care, remote live support, smart Web-based AR features, and patient monitoring facilities. The AR example using SDIoT-Edge has been shown in Fig. 10, which presents the communication of the AR system with the controller using the wireless gateways. Edge cloudlets are deployed near the edge of AR to support latency-sensitive data transfer to and from the AR applications. The application plane provides mobility-aware AR management for seamless service orchestration. AR supports medical specialists to inspect the patient, ask questions, and prescribe the medicines remotely. SDIoT-Edge can effectively enhance AR implementation in the medical industry. Currently, hospitals are digitizing the medical process; for instance, the patient's body can be digitally mapped for surgery or even venipuncture using AR. SDIoT-Edge can be used to manage patients' medical records, their current health status, and time to the next medical checkup. A software dashboard is utilized by the doctor, who receives information from the sensors attached to the patient's body to make real-time decisions. The same data is available to the on-sight medical staff in the form of AR solutions with diagnosis instructions. The doctor and the support staff collaborate using AR to provide medical services to the patients. AR requires latency-sensitive solutions that can be provided by the edge infrastructure. On-sight sensors use edge cloudlets near them to process latency-sensitive applications. Heterogeneity in the infrastructure can be handled by the virtualization offered by SDN and NFV. The controller in SDIoT-Edge uses OpenFlow messages to drive the network traffic and ensure end-to-end information delivery. Hence, smart sensors, edge cloudlets, and SDN offer extensive support for AR applications in healthcare.

AR requires compute-intensive computing capabilities due to having CPU and GPU-intensive AR algorithms. A novel signaling architecture is required to predict AR demands to support real-time network adaptation for varying resource demands of AR applications. The AR computation in SDIoT-Edge is distributed on several nodes comprising of access points, BS, gateways, traffic aggregation points, routers, and switches, etc. Here, the BSs are composed of the digital signal processors customized according to their workloads. SDIoT-Edge distributes the load at one BS to a nearby BS using centralized traffic management. Moreover, AR applications use hardware from different vendors and employ different communication protocols that can effectively be handled by SDN. The SDN controller can program the compute-intensive traffic from the AR sources using the OpenFlow protocol to the edge cloudlets' proximity. Moreover, the REST API at the northbound interface of the SDIoT-Edge is capable of providing an interface for application development to forward traffic in a customized way.

Moreover, SDIoT-Edge can be used to support AR in the industry by explicitly focusing on human-machine interactions. This paradigm includes AR to enhance physical and sensory experiences through digital graphics and computer-generated simulations. AR technology can be used to keep the track record of industrial products, their efficiency, and time to the next maintenance. By using a system dashboard, the managers can make decisions and adjust the values to prolong the maintenance schedule of industrial components. The same data is available to the on-sight technicians in the form of AR solutions with maintenance instructions. Technicians and the experts collaborate through three dimensional AR animations remotely.

C. Lessons Learned: Summary and Insights

The big data generated by smart infrastructure in SDIoT-Edge needs real-time treatment, which suffers from uncertain provenance challenges. Traditional data processing techniques, such as Structured Query Language (SQL), fails in this paradigm. Therefore, machine learning techniques can be applied to the generated data to ascertain key insights that help in critical decision making [212]. Consequently, data management is critical in the smart infrastructure as the quality of any smart system depends on precise decision-making capability. Moreover, the data generated by the smart city infrastructure needs cost-effective solutions closer to smart devices. The pre-processing of the data needs to meet the latency requirements, as the data from the surveillance infrastructure, healthcare measurements, and transaction records need real-time processing. This requirement can be fulfilled by the edge computing infrastructure. Data in smart cities and healthcare comes from heterogeneous resources at high volumes and velocity where edge computing may operate as a pre-processing resource. Similarly, smart healthcare requires performance integration, and virtualization services to support the latency-critical and compute-intensive tasks of AR. Furthermore, the protocols in the smart infrastructure need to be aligned with the available M2M, and WSN standards as the development of novel protocols involve higher costs and interoperability issues.

This section provided two case studies using the SDIoT-Edge for seamless service orchestration, including smart cities and smart healthcare. Besides the tremendous achievements in the smart city domain, enormous challenges still exist, including heterogeneity in communication infrastructure, diverse QoS requirements, and scalability issues. In this regard, the Software-Defined Internet of Vehicles (SDIoV) architecture can provide effective management in the heterogeneous IoV paradigm. Vehicles in the IoV change locations from one RSU to another, which needs mobility-aware service provisioning. Moreover, the security of IoV is vital, which can have damaging consequences. For instance, an adversary can access the vehicle and modify the lane information, which might result in accidents. Edge cloudlets in smart healthcare might be placed in the eNodeB or near user equipment. The eNodeB needs to provide coverage to a large number of users, which require higher computational power, whereas placing edge nodes near the user equipment will require less computational resources; however, it will impact the coverage. Data in IoV suffers from intense spatial and temporal variations that may congest edge computing BS. Moreover, the dynamic resource provisioning might suffer from under-load or overload resource utilization, which provokes QoS, performance degradation, and loss of revenue. The resources in the edge computing are virtualized to fulfill the application needs. However, the VM energy consumption of the idle state is 60% than that of the active state; so, underutilization of the resources must be avoided.

Future smart cities and healthcare will produce an immense amount of data that will need efficient management, where edge computing and the central cloud will play a pivotal role. However, there is still a need for seamless integration of distributed edge cloudlets and the central cloud for continuous service orchestration, which can be achieved by standardization efforts. There is a need to devise protocols at different levels of SDIoT-Edge for efficient resource delivery.

VI. Software-Defined Internet of Things and Edge Computing Standardization

Standardization is a key enabling factor to achieve high adoption due to the complexity in the integration of diverse platforms. A comprehensive standardization framework is required for the effective realization of IoT in the current communication-intensive paradigm [213]. During the start of Internet technology, TCP/IP was an essential standard toward the revolution of the Internet. However, by analyzing TCP/IP from the IoT perspective, we can find that most of its protocols at different layers are not implementable in IoT [214]. The computation resource constraint must be considered before any IoT standardization. Moreover, an extensive study from all prospects must be performed before any IoT standardization because the security element in the most widely used IP protocol has not yet been utilized due to the high cost associated with it. We discuss the separate standardization

TABLE VI
STANDARDIZATION EFFORTS IN IOT, SDN, AND EDGE COMPUTING

Technology	Standards	Characteristics	Applications Scope
IoT	CoAP, CoRE, MQTT, AMPQ, XMPP	• Application specific service provisioning • Authentication • Flexible innovation • VM mobility • Security and privacy	Application layer
	UDP, DTLS	• Data transfer • Traffic dissemination • Data classification	Transport layer
	RPL(RoLL)	• Traffic dissemination • Data transfer	Internet layer
	6LoWPAN		
	IEEE 802.15.4	• Physical connectivity • Interoperability • Fault-tolerance	Network/link layer
	IEEE 802.15.4 PHY/ Physical Radio		
SDN	OpenFlow	• Southbound communication • Interoperability • Flexible innovation • VM mobility	C-DPI communication
	REST API	• Northbound communication • Flexible innovation • Fault-tolerance • VM mobility • Security and privacy • Authentication	A-CPI communication
	OVSDB	• Southbound communication • Interoperability • Flexible innovation • VM mobility	C-DPI communication
	OF-CONFIG	• OpenFlow switch communication • Interoperability • Flexible Innovation • VM Mobility • Security and privacy	OpenFlow configuration of switches
Edge Computing	Multi-access edge computing	• Cloud services at edge within RAN in 5G and 6G • Fault-tolerance • Latency-sensitive applications • Traffic dissemination • Data classification	Edge-device communication
	OpenFog reference architecture	• Resource visibility and control • Localized command control, and processing • Autonomy at all levels • Orchestration and analytics • Virtualization and multi-tenant	Cloud-IoT resource-provision
	Mist computing	• Mobility support • Decreases latency • Higher efficiency • Minimal connectivity • Privacy-preserving	Parallel computation on IoT

measures in the SDIoT-Edge ecosystem. Table VI presents the standards in SDN, IoT, and Edge computing, their characteristics, and application scope. The characteristics correspond to the type of requirements that the protocol can address in the heterogeneous SDIoT-Edge computing.

A. *Internet of Things Standardization*

IoT comprises of low-cost and resource-limited devices as compared to the traditional computing infrastructure. Most of the IoT devices possess lower energy and employ low-end microcontrollers with limited memory. The traditional Internet protocols are not supported by these devices, which pose a greater challenge on the communication of IoT. Recently several IETF working groups have been created to address these challenges. Fig. 11 shows the IoT protocol stack representing protocols at different layers [215]. It shows the communication standards at all the layers that are responsible for the service orchestration. IETF standardization efforts have played a key role in establishing light-weight communication protocols for the IoT, deployed over the prevailing IP network. IEEE 802.15.4 standard has been designed for the Wireless Personal Area Networks (WPANs), which defines the interaction among the physical and media access control layers under the low-resource constraints, as shown in Fig. 11. For instance, 6LoWPAN is a lightweight protocol that ensures the delivery of IPv6 packets over the IEEE 802.15.4 wireless networks. Similarly, Low-Power-Wide-Area Network (LPWAN) supports long-distance IoT communication at lower bit rates. Long Range (LoRa) is an LPWAN technology based on the spread spectrum modulation technique. It is a non-cellular LPWAN wireless communication network protocol that operates in the license-free spectrum like 169 MHz, 433 MHz, 868 MHz (Europe), and 915 MHz (North America) supporting long-range communication with lower power consumption. It distinguishes itself from Wireless Wide Area Network (WWAN), which was designed to provide connectivity to the businesses carrying more data, which consumes more power.

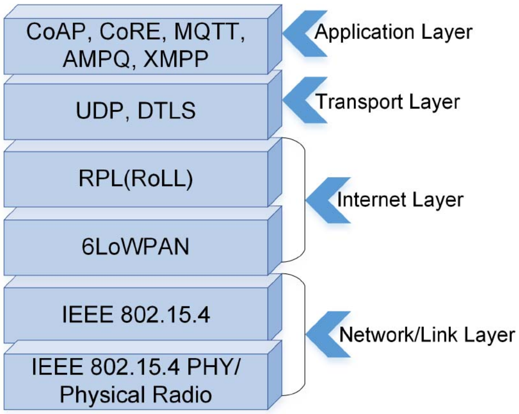

Fig. 11. The IoT protocol stack.

Various IoT standards have been developed including IEEE [216], Thread Group [217], and Open Interconnected Consortium [218], [219]. Any protocol for IoT standardization must consider resource limitation of the IoT infrastructure. Various working groups have been devised to standardize IoT communication protocols that are adaptable to the current TCP/IP paradigm. CoAP and Constrained RESTful Environments (CoRE) are the prevailing examples of such protocols, where CoAP is a lightweight HTTP version, whereas CoRE is a RESTful API for the application layer of IoT [220]. The Datagram Transport Layer Security (DTLS) is one such secure transport-level protocol that is suitable for resource-limited IoT devices employing User Datagram Protocol (UDP) for transport layer communication [221]. The Routing Over Low Power and Lossy Networks (ROLL) working group has designed a novel protocol for IPv6 for Low Power and Lossy Networks (RPL) at the Internet layer of the IoT protocol stack. In the same way, IEEE has developed 802.15.4, constituting MAC, and the physical layer of the TCP/IP protocol. Bluetooth Low Energy (BLE) is a standard that is characterized by low energy and the use of a fair data rate, which makes it appropriate for IoT applications [222].

The protocols mentioned above may further increase complexity and not yield improvement for IoT; hence, there is a need for dedicated communication platforms designed after the consideration of the challenges and needs of the IoT infrastructure.

B. Software-Defined Networking and Network Function Virtualization Standardization

OpenFlow is one of the first SDN standards released by the Open Networking Foundation (ONF), which defines the interaction of the controller with the forwarding devices [110]. OpenFlow is governed by the SDN controller; it is a powerful protocol that enables the management of forwarding tables of the remote network switches. Efforts in the standardization of SDN and NFV have not been performed by a single entity, where many organizations, open development institutions, and industry consortiums, including ETSI, IETF, ONF, 3GPP, and IEEE have developed SDN standards [122].

IEEE has been widely involved in developing standards for the SDN/NFV ecosystem. Such an effort to standardize the services life-cycle is the Next Generation Overlay Networks (NGSON) standard, which defines the protocols for service composition [223], self-organizing management [224], and content delivery [225]. A framework as a reference has already been developed for collaborative and customer-centric service delivery. IEEE standard for SDN/NFV security [226], performance [227], and reliability [228] defines security, performance, and reliability models. Each model includes standard terminology, analytics, and essential components of SDN/NFV. The IEEE standard for software-defined quantum communication defines an interface to quantum communication devices for the reconfiguration and implementation of diverse protocols [229]. Moreover, SDN bootstrapping procedures [230], SDN-based middleware solutions for control and management networks [231], recommended best practices for network reference model, and functional description of IEEE 802 access network standard [232] have also been developed by IEEE.

Extensive NFV standardization efforts are going on where IETF RFC 7665 was published in 2015, describing the framework to create and maintain SFC operations [233], [234]. ETSI NFV working group was formulated in 2012 to standardize the virtualization of network functions and define a framework to overcome the challenges of the novel architecture. Moreover, ETSI has published more than 50 group reports on security, service orchestration, and use cases [118]. Two working groups, IETF and IRTG, in the Internet Society (ISOC) are working on SDN standards. Two further working groups, i.e., Interface to Routing Systems (I2RS) and SFC, have been working on SDN standardization under the IFTF organization. In the same way, IRTG has been involved in publishing Request for Comments (RFC) titled "Software-Defined Networking: Layers and Architecture Terminology" standard [163]. The ITU-T has four working groups SG11, SG13, SG15, and SG16 that are working on SDN standard development projects. Furthermore, various open-source projects are being developed, including an open network operating system, OpenDaylight, Open vSwitch, OPNFV, and Floodlight open SDN controller [144].

IPsec protocol is a fundamental component of the Software-Defined Wide Area Networks (SD-WAN). It protects IP traffic at the network level. The Internet Key Exchange (IKEv2) is a key management protocol that is used in collaboration with the IPsec protocol. However, it suffers from scalability challenges due to the increase in IPsec entities. A flow-based solution develops security associations to defend against unauthorized access and scalability challenges [235]. ONF has also formed a working group to standardize the northbound interface of SDN [236]. It will accelerate the whole SDIoT-Edge paradigm because a northbound interface is integral in accomplishing the application-to-control plane communication. SDIoT-Edge intends to employ the prevailing standards devised by ONF for northbound and southbound interfaces. It enforces a distinct separation among the control and data planes by following the reference ONF architecture, where the flows are controlled by the flow entries at the data plane devices.

C. Edge Standardization

Edge computing for IoT service requirements is a vital paradigm; however, there is a need to standardize edge computing services, including identification of risks, responsibilities, and relationships during its operation. Recently, multiple efforts in cloud standardization have been performed, including IEEE Standard Association [237], ITU [238], ISO [239], NIST [240], [241], and Cloud Standard Customer Council (CSCC) [242]. Numerous working groups are involved in edge computing standardization; one of them is the MEC, which is an initiative in the industry specification group within the European Telecommunications Standards Institute (ETSI) that mainly works on edge computing [243]. The standardization aims at modifying specifications for uniting IT and telecom efforts at the level of the Radio Access Network (RAN). In the same way, the OpenFog consortium developed by giant tech companies is working on creating an architecture for applying fog in the IoT area [244]. Mist computing standard has also been introduced by Cisco to perform computation at the extreme edges of the IoT infrastructure [245]. It allows the computation at dispersed nodes of autonomous systems, which is extended through the edge to the IoT devices [246]. Edge standards need to be developed for the diverse set of operating platforms that have evolved from public and private partnerships in providing IoT-Edge services. Challenges in edge standardization arise from non-standardized operating environments. Therefore, the need to benchmark edge nodes still exists based on standard metrics provided by different researchers [247]–[249] and organizations such as the Standard Performance Evaluation Corporation (SPEC) [250].

Table VI shows the edge computing standards including multi-access edge computing, OpenFog reference architecture, and mist computing. An ISG within ETSI has been organized, which published GS MEC-IEG 006 standard in 2017, focusing on the service deployment using edge, [251]. This standard defines functional and non-functional performance metrics' improvement using edge computing. The capabilities of newly developed technologies for edge computing are tested on the ETSI ISG MEC proof of concept framework [252] where the edge video service orchestration [253], service delivery, and service chaining [254] are some of the examples.

The next-generation networks must be capable of satisfying the imperative requirements of latency, energy efficiency, bandwidth, and continuous mobility. These requirements can be fulfilled by radio access technology using edge computing and the core network management using SDN. OpenStack is a tool for cloud management that supports the centralized management of SDN. Although there are immense efforts in the standardization of edge computing, many challenges still exist, which are directives for further efforts. The services' complexity and management become a challenge due to the involvement of multiple third-party stakeholders, including application developers, device manufacturers, and network operators. Although the standardization efforts are extensively going on, the failure of the edge computing servers due to overloading is inevitable, which can induce huge costs for the network operators. ETSI ISG proof of concept [252] proclaims that the success of edge computing lies in continuous coordination with the central cloud. Therefore, a continuous interaction among the central cloud and edge computing is imperative for seamless service orchestration in edge paradigm.

D. Lessons Learned: Summary and Insights

In this section, we have presented current efforts in the standardization of Edge, IoT, and SDN infrastructure. There are several standardization bodies, forums, and corporations that are working on the standardization of the smart infrastructure. However, a joint effort in this scenario is still missing. We have illustrated that merely connecting many things to the Internet does not disseminate into smart infrastructure; there is a need to develop relevant communication standards and protocols.

SDN offers communication services independent of device vendors due to the extensive standardization efforts by ONF. It is capable of managing heterogeneous networks where sensors from different vendors can be operated in a single network. OpenFlow is capable of supporting interoperability among the data plane devices. Although edge computing is not directly associated with SDN, the requirements of edge servers can be mapped to the characteristics of SDN, which makes it a promising solution for edge computing. The northbound RESTful interface in the SDIoT-Edge has the same level of significance as the southbound OpenFlow interface. Seamless communication among the applications at the application plane and the controller in SDIoT-Edge needs vendor-independent interfaces. Therefore, standardization of the northbound interface will significantly enhance network management using customized applications.

Immense computational and network resources are required to achieve a smart transformation that can be provided by SDN and edge computing. Therefore, standardization of SDIoT-Edge is a key challenge in its implementation as the early IoT products have been expeditiously developed by equipping sensors to collect and transmit data. However, with an extremely large increase in IoT products, standardization has become a key challenge to gain optimum benefits. Currently, key technologies are present; however, most of the future work lies in the standardization of the diverse platforms in SDIoT-Edge. Many parallel efforts have been going on in the standardization that lacks a coordinated effort. Rapid development in this field also poses challenges to standardization. Similarly, most of the IoT solutions only consider a limited operational paradigm to reduce the time to market and increased benefits. Large-scale solutions need immense investments; moreover, the return on investment cannot be guaranteed during the development process, where most of the business owners opt to avoid risks. There is a need for the development of a standardization body that includes representatives from different vendors and service providing agencies. The research in the SDIoT-Edge paradigm is in its emerging phase; hence, many opportunities exist in developing standards, protocols, and operational environments.

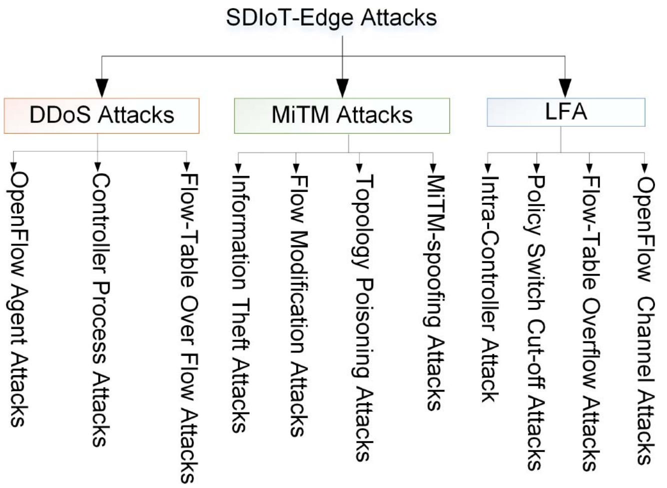
SDIoT-Edge Attacks
DDoS Attacks
MiTM Attacks
LFA
OpenFlow Agent Attacks
Controller Process Attacks
Flow-Table Over Flow Attacks
Information Theft Attacks
Flow Modification Attacks
Topology Poisoning Attacks
MiTM-spoofing Attacks
Intra-Controller Attack
Policy Switch Cut-off Attacks
Flow-Table Overflow Attacks
OpenFlow Channel Attacks

Fig. 12. Attack taxonomy in the SDIoT-Edge ecosystem.

VII. Security and Privacy in Software-Defined Internet of Things Using Edge Computing Ecosystem

Security and privacy are the most important issues in the SDIoT-Edge paradigm [255]. SDIoT-Edge incorporates a diverse set of devices composed of a multitude of vendors, all having different mechanisms to deal with security challenges. Therefore, addressing the security and privacy of SDIoT-Edge poses a vital challenge. Similarly, IoT and edge devices are limited in computing resources; therefore, providing specialized security arrangements becomes a difficult task. Many security vulnerabilities exist in SDIoT-Edge that can be leveraged by attackers. We discuss security and privacy by critically considering the vulnerabilities arise due to the integration of heterogeneous platforms in SDIoT-Edge. A major threat emerges because of the resource limitation, which invokes increased attack space in SDIoT-Edge. We categorize the SDIoT-Edge attacks based on the STRIDE framework, which provides a systematic terminology to classify the attacks. Fig. 12 shows the taxonomy of the attack vulnerabilities in SDIoT-Edge [256]. Moreover, a taxonomy of

TABLE VII
SDIoT-Edge Attack Taxonomy Based on the STRIDE Framework

Threat	Definition	Affected Services	Attack Examples
Spoofing	Impersonating	Edge-IoT data spoofing	• Malicious services [258]. • Malicious information [259]. • Critical information access [192]. • Unauthorized access e.g., MiTM [256], [260].
Tampering	Malicious data update	Data integrity in SDIoT-Edge	• Modifying critical data in edge [261]. • Authentication key update [139]. • Integrity attack on user data [141]. • Illegal access [262]. • Cross-service attack [41].
Repudiation	IoT repudiation	IoT-Edge device authentication	• Repudiation on device accountability [263]. • Edge data access [264].
Information Disclosure	Information theft	Service confidentiality in SDIoT-Edge	• Data leakage on edge [16]. • Data theft on IoT-Edge interface [265]. • Communication spoofing [66], [256], [260]. • Digital signature attack on IoT [256], [260]. • Device's signature access [221].
DoS	Service unavailability	IoT-Edge service orchestration	• Flow table overflow [266]. • C-DPI attack [198], [199]. • A-CPI attack [267]. • Application plane attack [267]. • Control plane attack [268].
Elevation of Privilege	Unauthorized access	Authorization	• Resources access by unauthorized IoT [67]. • Impersonating [260].

attacks based on the STRIDE framework has been presented in Table VII.

A. Threat Modeling Using STRIDE Framework

We have performed threat modeling in SDIoT-Edge using the STRIDE attack modeling framework developed by Microsoft [257]. It constitutes six security aspects, including spoofing, tampering, repudiation, information disclosure, DoS, and elevation of privileges that are critical in any network setting. We map the SDIoT-Edge vulnerabilities using the STRIDE framework and analyze their consequences. Table VII discusses the taxonomy of STRIDE using attack definition, security services affected by the attack, and the consequences of these attacks on SDIoT-Edge. Security solutions need to be developed by considering the STRIDE attack taxonomy, which provides an overview of the security aspects in SDIoT-Edge. Spoofing attacks capture the information among connecting hosts, which can provoke vital data loss and illegal access to critical information. Tampering attacks are directed toward the data integrity in SDIoT-Edge, causing illegal data access, cross-service irregularities, and malicious updates of the authentication keys. The repudiation attacks cause anomalies in the IoT-Edge authentication, which invoke inconsistencies in data transfer at the edge and malicious device accountability issues. The information disclosure attacks target the vital information exchanged between the smart infrastructures, which may include healthcare, vehicular, and fire-safety data. The DoS attack can have devastating outcomes on SDIoT-Edge where several use cases can be presented, e.g., flow table overflow, C-DPI, A-CPI, application plane, control plane, and data plane attacks. In the elevation of privilege attack, an IoT device maliciously accesses unauthorized resources, which can spoof the information and provoke other information leakage issues.

Although the STRIDE framework encompasses an extensive attack taxonomy, there are still other attack vulnerabilities that arise specifically due to the multi-device interaction, resource limitation, and increased attack space. We explicitly discuss these attack vulnerabilities in the next section.

B. Security Vulnerabilities

The distributed nature of edge cloudlets in the SDIoT-Edge reveals novel security risks that were not present in the centralized cloud. SDIoT-Edge deals with heterogeneous entities, including virtualization platforms, multiple IoT variations, distributed systems, and wireless networking technologies. To ensure security, we need to safeguard all the involved platforms using a security solution. Due to the distributed nature of infrastructure and mobility of devices, autonomous security solutions are required, which induce lower-latency as compared to the centralized security solutions. The network endpoints are prone to security challenges where an attacker can easily access the devices and install malicious software to generate attacks. Data processing on edge causes privacy leakage issues due to multi-device interactions. Moreover, lack of standardization and competition among the firms to introduce novel IoT devices for market survival is contributing immensely in increasing the security risks. The authentication mechanisms at different levels of SDIoT-Edge is one of the typical security problems in heterogeneous networks. For instance, a smart temperature meter in smart homes can be distinguished by a separate IP address. An adversary can manipulate this device, report false information, and tamper data, which will disrupt temperature management in the smart home.

Limited computation power in these devices makes them vulnerable to be exploited as bots for a diverse set of attacks, including Distributed Denial of Service (DDoS) [267]–[271], Man in the Middle (MiTM) [256], location-based privacy concerns [141], [272], [273], and Link Flooding Attacks (LFA) [274]–[278]. SDIoT-Edge nodes are highly prone to DDoS attacks due to the lack of resources to implement a local security solution to defend against these attacks [33], [279]–[281]. In addition to these challenges, LFA can exploit weaknesses of SDIoT-Edge and flood important links in a variety of ways, including policy switch attacks [199], OpenFlow channel attacks [198], [282], and data plane resource saturation attacks [270]. The OpenFlow channel attacks can disconnect C-DPI in SDIoT-Edge by using link flooding and DDoS. The policy switches in SDIoT-Edge manage the policy of the network, including the firewall and security [283]. LFA on the policy switch causes the disruptions in per-packet consistency, security, and privacy. DDoS attacks on the data plane cause flow table memory saturation in the SDN switches [284]. This attack sends a flood of flows to the switches that causes packet miss in the switch flow table and trigger a new flow rule installation. Multiple flow rule installation requests cause limited Ternary Content-Addressable Memory (TCAM) in the switches to overflow and disrupt the network communication. Strong NIDS can be developed for a specific edge node and can be deployed on similar nodes. In the same way, lightweight solutions against flooding attacks can also be developed specifically, providing defense against MiTM-spoofing attacks.

C. Countermeasures, Solutions, and Security Protocols

Data transfer in SDIoT-Edge suffers from immense attack vulnerabilities. Attacks like worm propagation, sniffer attacks, and resource saturation attacks can be launched to congest the network resources or bring down the controllers. Edge computing utilizes a variety of different networks, including wireless, Wi-Fi, and ultra-dense networks, which introduce immense management traffic. Thus isolation of management and data traffic poses challenges of data management. Moreover, adversaries can utilize this data to generate DoS attacks.

Developing on-device security solutions for IoT is hard because of the resource limitations [285]. The programmability characteristic of SDN can be used to develop network-level security solutions. SDN provides the capability to program the network for customized traffic forwarding and network policy enforcement. Recently, a few techniques have been proposed to use SDN's programmability feature to secure IoT [198], [199]. For instance, the Floodlight open-source controller provides the flexibility to deploy custom security applications [282]. It is easy to analyze network traffic due to centralized traffic management in SDN, which is an imperative characteristic for developing security solutions. Different mechanisms can be used to detect the malicious traffic (e.g., machine learning, bloom filtering), which can then be mitigated by customized flow rule installation on the forwarding devices [286]. Data protection during the transmission process can be efficiently achieved using the SDN's capability of VLAN ID to separate traffic in different VLAN groups, which can further be mitigated. Homomorphic encryption techniques can be used to solve the challenge of data modification on distributed edge nodes [287]. Authors in [288] propose privacy-preserving public auditing to protect data stored in the cloud via third-party auditors. Two authentication protocols (i.e., for the same and different access privileges) are designed to verify file search results. Finally, a third-party auditor ensures the security of the data storage, where the auditor itself uses homomorphic encryption and random mask for protection. IPSec is a security protocol that is used to protect network traffic in SD-WAN [235].

Security vulnerabilities arise while transferring data to and from the edge nodes in the SDIoT-Edge. Verifiable computation [289] uses compute-nodes to offload the task. A public encryption key is generated by the IoT where key-value, and computation can be compared to authenticate the data [290]. Trust between edge nodes and the end devices must be available to ensure security; where Clemens *et al.* proposed a trust-based authentication solution to solve this challenge [291]. This method ensures the authentication using integrity measurement and attestation to verify integrity evidence from edge devices. Echeverría *et al.* proposed a trust identity solution in disconnected environments using identity-based cryptography and secure key exchange. This solution provides application, OS, network, and site-level controls, which can efficiently safeguard the disconnected environments [292].

D. Privacy Concerns

IoT and edge nodes can be manipulated in vehicular networks, which can be transformed into adversaries to spread disinformation to the vehicles driving through the edge node's range [293], [294]. The user location can be exposed using IoT devices when they offload their data on edge nodes, for example, wearable devices send data to the edge, which performs the computation and may further forward this data to the cloud [295]. During the data transmission, edge nodes and central cloud can ascertain the location of the user even in the presence of anonymity solutions. Fake, comprised, and manipulated IoT and edge nodes are threats to the network infrastructure due to the difference in trust models being used in different devices that require a massive investment in trust management [296]. Similarly, it is challenging to develop blacklists for these suspicious devices in a massively distributed paradigm.

Individuals are usually reluctant to share their personal information; however, with more IoT-Edge deployment, personal information is exposed to unnecessary individuals. Privacy breaches such as those arising from the exploitation of credit card information leaks at payment on edge nodes, in addition to customers' health information by the pharmacies, financial information by the insurance companies, and travel information at the refueling stations can be easily exposed in SDIoT-Edge. Consequently, there is a need for comprehensive privacy mechanisms to secure user data leakage in SDIoT-Edge.

E. DDoS Attacks

In the SDIoT-Edge paradigm, most of the devastating attacks are the DDoS attacks. IoT comprises a heterogeneous range of devices; hence, the manufacturers often use this as an excuse to provide adequate security measures. Therefore, IoT devices act as readily available objects that can be manipulated for DDoS attacks. The most devastating DDoS in SDIoT-Edge includes the following.

- Flow-table overflow attacks
- Controller process attacks
- OpenFlow agent attacks

The first DDoS attack against smart home-based IoT devices was observed in 2014 when the attackers broke into one hundred thousand IoT devices by sending malicious emails targeting the enterprises and individual customers around the world [297]. During the past several years, DDoS attacks have become a routine in the IoT infrastructure, where more than 550 attacks have been observed per year, generating a peak traffic load of 800 Gbps at a time. These attacks are growing rapidly, and the yearly growth of 150% is observed that has incurred a cost of approximately 30,000$ per hour [298]. The most severe is the Mirai malware attack, which had blocked access to the IoT services and ISPs in the U.S by injecting Mirai malware in the IoT devices. This malware manipulated the IoT devices into bots that attacked important servers [281], [299]–[302].

IoT networks are prone to DDoS attacks because they are not supported by the policy mechanisms for traffic handling. In the same way, the devices connected by the SDIoT-Edge network are heterogeneous in nature, employing different computation and battery capacities. Limited computation resources in IoT makes the deployment of state-of-the-art security solutions a challenging task. The availability of service provisioning has also been limited and only a specific number of requests can be fulfilled at a certain time. Another vulnerable nature of IoT is that it is designed as an open framework for multiple connected devices; therefore, considerably less control exists over simplified connected objects. The workflow in IoT is also dependent on multiple devices that can have a cascading effect on the attack area during DDoS.

F. MiTM Attacks

In the MiTM attacks, the adversary intercepts the communication channel between two hosts and access, modify, or replace the ongoing traffic [260]. The victims continue the regular interaction believing that the communication channel is protected. The aim of the MiTM attack is to compromise confidentiality, integrity, and availability of the network by eavesdropping, intercepting, and spoiling the network traffic. In SDIoT-Edge, the OpenFlow channel can become the target of the MiTM attacks where the information exchanged between the switches and the controller can be intercepted by a smart adversary.

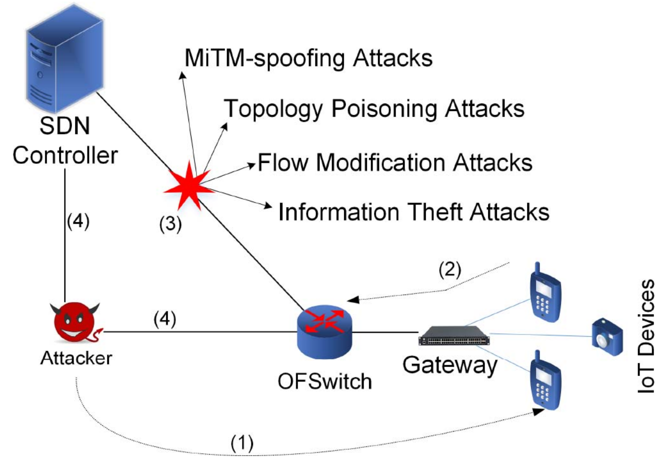

Fig. 13. An example of the MiTM attacks on the OpenFlow channel of SDIoT-Edge.

Li *et al.* [256] and Stojmenovic [303] propose the feasibility of MiTM attacks against the cloud and IoT infrastructure. A considerable number of IoT devices suffer from the challenge of firmware update attacks where the adversary updates the IoT device's firmware using a legitimate update method. If there is a device with such a vulnerability: 1) the adversary comprehends the device by firmware modification; 2) then deploys a client certificate at the gateway and OFSwitch claiming that both the nodes need to use this certificate for the future communication; 3) the adversary then spoofs the communication between the controller and OFSwitch; and 4) launches MiTM attack. Li *et al.* [256] present the idea of an SDN-based centralized controller and fog computing to alleviate MiTM attacks. They investigate MiTM attacks on the OpenFlow channel and provide defense using Bloom filters in order to examine stealthy malicious updates in the packets. However, the problem with their approach is that if all the switches in the path of a flow are comprehended, then this approach becomes invalid. Fig. 13 shows the sequence of possibilities of MiTM attacks on the SDIoT-Edge infrastructure. To avoid MiTM attacks, lightweight encryption/decryption schemes are needed to authenticate the devices prior to serving the requests. The figure shows that four types of attacks are possible in this scenario, as given in the following.

- MiTM-spoofing attacks
- Topology poisoning attacks
- Flow modification attacks
- Information theft attacks

In the MiTM-spoofing attacks, the attacker can intercept the OpenFlow channel and modify the victim's switch forwarding table using spoofing techniques such as ARP spoofing. The Link Layer Discovery Protocol (LLDP) packets are utilized by

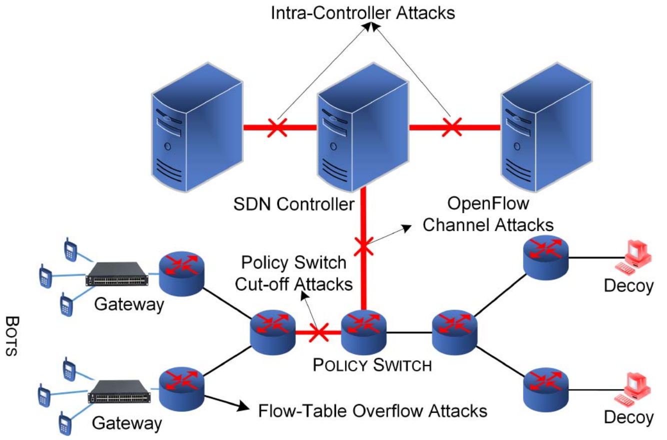

Fig. 14. An example of LFA in the SDIoT-Edge paradigm.

the controller in order to learn the topology of the network. The attacker can customize the LLDP packets by modifying the maximum length and output port in a PACKET_OUT message. The switch will not be able to access the message due to the malicious modifications and will respond by sending a fake topology to the controller. In the flow modification attacks, the attacker redirects traffic from a host to an irrelevant host by modifying flows. Similarly, during the process of redirecting network traffic, the adversary has the ability to collect vital information over the network, causing vital information loss.

G. LFAs

LFA has been involved in causing an outage in the major Internet infrastructures [304]. The adversary can exploit security limitations in the IoT devices and launch devastating LFA on the Internet infrastructure. Flooding attacks have the ability to congest specific links connected to important nodes. LFAs are dangerous in SDIoT-Edge because they use low-rate traffic to flood important servers. This traffic behaves the same as the legitimate traffic; so, detection and defense against LFA become a complex challenge. Different variations of LFA exist, such as Coremelt [305], Crossfire [306], and Spamhaus [307]. Fig. 14 shows that IoT nodes can be exploited to send low-rate traffic toward decoys, which can disrupt the network and shut it down in extreme conditions. The figure shows that LFA causes four types of vulnerabilities in SDIoT-Edge networks, which are as follows.

- OpenFlow channel attacks
- Policy switch cut-off attacks
- Intra controller attacks
- Flow-table overflow attacks

In the open flow channel attacks, the adversary can manipulate IoT devices to send low-rate traffic toward the switches to cause a packet miss. Furthermore, the switch will send the traffic packets to the controller, which can cause flooding on the OpenFlow channel. This attack has the ability to disconnect the OpenFlow channel from the rest of the network. The policy consistency is an essential characteristic in the SDNs to provide seamless services that are performed by the policy switch in the network. A smart adversary can cause flooding on the policy switch links in order to disconnect them from the rest of the network. In the same way, the switches are equipped with low-size storage known as TCAM to maintain flow tables. The adversary can send a flood of low-rate packets to the target switch causing a packet miss that triggers the installation of new flow rules [266]. Consequently, after a significant rule installation, a point will come when the TCAM memory will be exhausted. Therefore, LFA has the ability to disrupt communication in SDIoT-Edge; therefore, adequate security must be provided in order to defend these potentially devastating attacks.

H. Lessons Learned: Summary and Insights

In summary, the IoT-Edge infrastructure suffers from many security threats that can leverage the resource limitation of these devices. Research in IoT-Edge has been in its evolutionary phase; in this regard, the solution developers and industry must put extensive efforts into realizing the underlying threats before providing any solution for the SDIoT-Edge ecosystem. We have provided a taxonomy of the attacks in SDIoT-Edge by considering the vulnerabilities that arise due to the integration of heterogeneous infrastructure. The lack of standardization and immense product delivery needs have instigated core security risks in IoT that pose a severe threat to the network infrastructure. The lack of real-time simulation for IoT experiments poses a challenge for the researchers to develop effective security solutions. The vulnerabilities of the controller in SDN have been well established and can become a critical challenge in the SDIoT-Edge paradigm. Research on providing a solution for the IoT devices can be of great benefit to the current network paradigm. The security strategies must also consider a holistic approach that leverages the benefits of multiple solutions to cater to the security issues in IoT. For example, researchers proposed solutions against OpenFlow channel DDoS attacks, data-plane attacks, and controller attacks separately. However, a comprehensive security mechanism that incorporates the characteristics of all the solutions can be extremely beneficial for the SDIoT-Edge.

VIII. Open Issues and Future Research Directions

Edge computing is continuously being deployed for SDIoT service orchestration. It is a novel paradigm that has been creating widespread opportunities for efficient resource deployment, interoperability, and management. However, multiple areas in SDIoT-Edge still need to be addressed adequately. This section provides insights toward open research issues and future research directions. Table VIII provides the current challenges in the SDIoT-Edge paradigm and suggests guidelines to solve these issues.

A. Resource Provision at Edge Nodes

Traditional cloud computing provides powerful resources to address compute-intensive tasks. Applications that can tolerate latency and cost associated with transferring and receiving back the computation can successfully utilize the central cloud. However, IoT devices are resource-limited with real-time processing needs to control sophisticated infrastructure. Moreover,

TABLE VIII
A DETAILED DISCUSSION ON OPEN CHALLENGES AND SOLUTION GUIDELINES

Challenges	Causes	Guidelines
Resource Provision at Edge Nodes	• Dynamic workload handling at BS. • Compatibility issues at BS. • Platform integration issues. • Lack of virtualization capabilities.	• Hardware-independent solutions e.g., SDN/NFV, VMs. • Virtualization at different layers. • VM management using overlays at BS. • VM caching for rapid future integrations.
Cloud Service Discovery for IoT	• Edge nodes discovery in heterogeneous networks. • Data management issues. • Production, partitioning, classification, and delivery of data. • Workflow related issues. • Workflow execution, fault-tolerance, and integration.	• Service discovery using ETSI MEC stage 3 level APIs [29]. • Use of data distribution service [308][310]. • Storage using Luby Transform [309]. • Interoperability of the services using [29].
Heterogeneous Service Architecture	• Networking among heterogeneous devices. • Heterogeneous infrastructures for services. • Heterogeneous security and privacy challenges. • Data naming issues.	• Softwarization of services NFV/OpenFlow. • Customized VM management at the edge cloudlets. • Multi-innovation using current networking and IPv6. • Employing the NDN technique for naming.
Provision of a Marketplace	• Lack of edge services marketplace. • Nonavailability of standards and SLAs. • Lack of efficient resource utilization. • Lack of coordination among edge cloudlets. • Distributed request handling challenges.	• Development using the central cloud example. • Seamless coordination among edge cloudlets. • Handling temporal variation in service demands. • Low-cost service borrowing from edge cloudlets. • Auction-based service-provisioning [310], [311].
5G/6G Networks	• Realization of wireless SDN. • Wireless separation of C-DPI has not been optimized. • Effective spectrum sharing. • Heterogeneous message exchange and resource over-provisioning.	• Protocols and standards for wireless SDN. • Novel high spectrum-capable hardware. • Device accessibility using 5G/6G high coverage. • Synchronization protocols for message exchange.
Lightweight Algorithms for IoT-Edge nodes	• Lack of resources to handle compute-intensive libraries and algorithms. • Energy consumption in the data processing. • Traditional algorithms need higher computation resources.	• Lightweight machine learning algorithms to perform data classification and partitioning [14], [154]. • Apache Quarks [312] provides a baseline. • Data filtering before delivering data to the cloud.
Lightweight OS for IoT-Edge nodes	• Resource-limitation in IoT-Edge nodes. • Rapid deployment issues over diverse platforms. • Missing essential OS components in IoT-Edge. • Updates and maintenance issues.	• Novel lightweight OS development. • Enhancement in the Docker-based solutions. • Development of lightweight components in the IoT. • Reliable updates patching and rollback.
Unified Architectures	• Multiplatform synchronization. • Communication heterogeneity during service orchestration. • Distributed integration among edge computing and IoT. • Mobility of IoT e.g., smart vehicles.	• Unified architecture for different devices. • Standards and protocols for interoperability [9]–[11]. • Coordinator-based solution for resource sharing and scheduling [313]. • Mobility-aware architecture for service orchestration [314].
Security Handling Mechanisms	• Lack of trusted administering capabilities in IoT-Edge. • Data transfer among heterogeneous platforms. • Data leakage issues. • Constraints on high-cost encryption algorithms i.e., 1024-bit RSA [315]. • Resource constraints.	• Privacy-preserving techniques e.g., differential privacy [316]. • Authenticating gateways at multi-levels [317]. • Homomorphic encryption for security [287]. • Provision of lightweight cryptography solutions. • Blockchain-based solutions [318], [319].
Eavesdropping, Flooding, and DDoS Attacks	• Increased attack surface. • Lack of resources increases vulnerability. • Non-availability of IoT security solutions. • Security element is neglected for higher performance. • Simplified communication models.	• Network solutions using SDN [282]. • Edge-based IoT device authentication [261]. • Secure frameworks in SDN [198], [199]. • Use of authentication for a secure connection. • Standardized network protocols for security.
Controller Bottleneck	• Controller-switch communication delay. • Central management causes single point of failure. • Scalability of a single controller. • The OpenFlow channel vulnerability caused by flooding flows.	• Seamless communication between controller and data plane devices. • Distributed controllers with abstraction. • Virtualization using NFV and VM management. • OpenFlow channel security [320].

the mobility requirements, throughput, big data needs, low-latency, precise control, and data aggregation pose a challenge on the use of traditional cloud computing. Therefore, the computation processing can be brought at the edge of IoT to address these challenges.

Edge cloudlets are placed between IoT and the central cloud using communication infrastructure, including switches, gateways, access points, routers, and BSs. Heterogeneous communication paradigms and multiple platforms provoke resource provisioning challenges and integration issues. Edge nodes contain limited resources as compared to the central cloud; therefore, intelligent decision making is needed to identify the services, which can be handled at the edge and those that should be transferred to the central cloud. The workload at BSs is handled by the signal processors that are configured for specific operating conditions. Hence, they are not able to handle dynamic workloads as they are not developed to provide generic computation resources. Several solutions to address this problem have been provided by industry, including Nokia's solutions for edge computing [321], OCTEON Fusion [322], and Cisco's IOx [323]; however, these solutions are dependent on the use of specific hardware and may not be suitable for heterogeneous environments. Virtualization techniques using SDN, NFV, and VMs can be exploited for dynamic workload handling, platform integration challenges, and compatibility issues at the BSs. Further, VM caching techniques can be exploited for swift integrations in future. Consequently, the need for resource provision at edge nodes is still a challenge in the SDIoT-Edge ecosystem. A scalable solution for IoT resource provision at the edge, employing SDN can be an impactful research direction in this paradigm.

Due to the lower computation power at the edge nodes, the requesting devices may suffer from resource underprovisioning, which can be addressed by employing distributed edge computing. In this paradigm, multiple edge nodes can cooperate in providing services where SDN can address communication management issues. SDN applications in the context of Software-Defined Wireless Networks (SDWN) can provide the baseline to implement such solutions.

B. Cloud Service Discovery for Internet of Things

Edge device discovery in a heterogeneous and decentralized IoT environment is challenging because it needs special arrangements to search specific edge nodes over the network. The exponential growth of IoT devices and massive data generation provoke data management issues, where novel techniques will be required to handle heterogeneous workloads and diverse infrastructure. This challenge can be addressed using distributed data management techniques [308] where storage using Luby Transform [309] provides a use case deployment. Seamless service provisioning in a heterogeneous environment is also a key challenge, where proactive fault-tolerant mechanisms are required to deal with failover situations. IoT devices need special arrangements for edge service discovery as they lack compute-intensive processing capabilities; therefore, research in rapid service discovery needs special attention.

Using edge services to fulfill widespread IoT demands, requires the use of switches, routers, gateways, and BSs that are owned by the public and private organizations, which provoke workflow execution, management, and integration issues. However, different institutions may have organizational-level security parameters that need to be standardized for disparate firms providing services. Subsequently, as several infrastructures from different organizations are collaborating to provide services, there is a need to furnish standardized pricing models for service provisioning.

For efficient edge cloud service discovery, the edge of the network should be incorporated in the communication ecosystem where the edge cloudlets will be accessible to the IoT devices, which can offload the compute-intensive tasks. Another solution to this challenge is using ETSI MEC stage 3 level APIs, which use token endpoint URI for communication [29]. Moreover, public cloud resources are available for service discovery and orchestration like EC2 and Amazon Web Services (AWS). However, there is a lack of similar platforms for edge computing services discovery. Edge-as-a-service (EaaS) has been proposed as a first effort in this paradigm by Varghese *et al.* [115]. A discovery protocol is used, which identifies the edge nodes and makes them publicly available using a controller. This platform owns a three-tier architecture where the discovery layer is at the bottom, whereas the top layer constitutes the application servers. There is still a need to develop edge discovery mechanisms to publish the edge services, which can be utilized by any requesting device on a pay as per use pricing model.

C. Heterogeneous Service Architecture

Supporting heterogeneous infrastructures, devices, and diverse service demands are critical challenges in SDIoT-Edge. This architecture integrates a diverse combination of platforms, servers, network topologies, and protocols. It constitutes a heterogeneous architecture causing complexities to program, operate, manage, and secure applications operating on different platforms and locations. Security challenges can be handled by providing network-based solutions, which provide an efficient way to secure resource-limited IoT devices.

The distributed edge nodes possess sufficient resources to handle service requests at the resource-limited devices' edge. The application developers face challenges in application development for end-to-end service orchestration in a heterogeneous SDIoT-Edge paradigm. Although a few techniques address the challenge of programmability in edge computing, they did not consider the specific IoT characteristics [314], [324], [325]. The device discovery is complex in IoT-Edge as the IoT devices are unaware of the nearby edge platforms. Moreover, the edge nodes need to deploy multiple server-side programs; however, the deployment and management of these programs is another challenge because of the distributed nature of edge nodes. These challenges can be addressed by virtualization services provided by SDN/NFV and VM management.

The versatility in the service architecture also poses a challenge on data management, where different storage servers having different operating systems increase complexities in file naming, resource allocation, and reliability management. The naming convention of data becomes another critical issue, due to the generation of data by multiple resources, where the URI and DNS schemes are not suitable for the dynamic edge networks and IoT. The IP-based naming conventions are not applicable for multi-source and multi-task edge nodes as they induce huge implementation costs. Schemes like Named Data Network (NDN) [325] provide a hierarchical naming strategy for the distributed networks, which is easy for the network owner to manage. However, it requires extra proxy servers in the network to integrate heterogeneous communication protocols. Additionally, it needs source hardware information, which increases privacy and security issues. The MobilityFirst technique [314] separates the device-identity from the IP and MAC addresses to ensure mobility-aware device discovery. However, this technique requires a globally unique device-identification, which is not user-friendly.

D. Provision of a Marketplace

Similar to a cloud marketplace, there is still a need to develop an edge marketplace where edge services can be acquired as pay-per-demand and pay-per-use pricing standards of the central cloud. Edge computing has been tremendously involved in providing services to the latency-sensitive IoT devices, which suffer from the nonavailability of standards and SLAs to develop a marketplace. Due to the heterogeneity in the edge infrastructure providers, lack of coordination becomes an inevitable situation that results in unbalanced resource utilization. The temporal variation in the resource demands can be analyzed to provide resources to the requesting stations. Moreover, coordination strategies among distributed edge cloudlets should be developed, which can deal with the resource over provisioning challenges. Edge cloudlets can be utilized to provide services in terms of VMs, where resource requests over the distributed edge nodes can be addressed via a collocated market. The edge market place should have the capability to provide on-demand and on-path services for the customers. As the edge cloudlet resources are limited, they cannot provide boundless services to the requesting stations. More often, the demand for a resource can exceed as compared to the available resources. Therefore, a scheduling mechanism is needed to allocate the resources to the incoming requests from the customers, especially for the latency-sensitive applications [326]. Moreover, these applications cannot be transferred to other distributed edge cloudlets due to the QoS requirements, which pose a challenge on distributed request handling. Hence, the market place for edge computing should be devised considering the fact that the resource provisioning over the edge cloudlets is distributed and uncoordinated. A solution to this challenge would be the deployment of auction-based service provisioning to the requesting stations based on the strategies developed in [310], [311].

Moreover, immense challenges still remain in developing SLAs and pricing standards for such a diverse marketplace. These challenges may include the allocation of resources over time to the requesting applications and the pricing mechanisms to service the requests. Therefore, a pricing model can be assigned to every edge cloudlet considering on-demand, decentralized, and uncoordinated requests.

E. 5G/6G Networks

5G/6G networks are key enablers for the SDIoT-Edge, which offer a high-speed communication spectrum that can support the need for seamless interaction among different platforms. Although 5G networks have been increasingly deployed as a communication resource, an effective realization of wireless SDN is still a challenge where the wireless separation of C-DPI has not yet been optimized. In this regard, the development of standards and protocols for the wireless SDN will provoke immense opportunities using 5G/6G in SDIoT-Edge. Device-to-Device (D2D) communication using spectrum sharing is expected to grow using 5G/6G technology. However, D2D communication in SDIoT-Edge needs high spectrum hardware and interference management to optimally utilize the throughput and provide reliable communication. The convergence of 5G/6G, edge computing, IoT, and SDN with AI can provide analytics capabilities to enable better user experience in communication, digital content access, automotive IoT, smart homes, and VR. Moreover, edge analytics and AI give rise to autonomous networks that enhance the user experience. However, spectrum sharing and energy harvesting become a critical challenge in SDIoT-Edge. Further research in 5G/6G implementation in SDIoT-Edge will provide novel spectrum sharing solutions, which can considerably enhance the coverage where the high-speed switches at the data plane and edge resources can interact seamlessly. Further, there is a dire need of the service infrastructure that can optimally harvest the capabilities provided by highspeed 5G/6G communication.

In order to optimally utilize the benefits provided by high-speed 5G/6G communication, there is a need for high-spectrum capable hardware on both ends (e.g., client and carrier). In the high device density paradigms, the 5G/6G networks are capable of providing context-aware middleware solutions that can overcome the challenges of scalability, heterogeneity, and mobility. They support the realization of autonomous networks, which can provide fault-tolerance during dynamic network changes. Although 5G/6G networks provide faster communication, the network scalability is still a major issue as managing state information of large-scale IoT devices requires heterogeneous message exchange, which needs synchronization protocols. Although solutions for cloud services provision for IoT over 5G/6G communication channels have been proposed, energy-efficient resource provision, security, privacy, and VM management are still key challenges in SDIoT-Edge [137], [327], [328]. 5G/6G networks can support wireless-NFV for the entire network, which has the capability to simplify the service orchestration in the SDIoT-Edge paradigm. 5G/6G networks, with the help of NFV, aim at providing scalable cloud resources using customized network slicing for IoT applications. Though 5G/6G networks are key enablers for the SDIoT-Edge, trusted communication over the high-speed network in the presence of eavesdroppers is a significant challenge.

F. Lightweight Algorithms for the Internet of Things and Edge Nodes

Lightweight algorithms and libraries become imperative when there are constraints on computational power, battery life, memory, and storage. As edge nodes and IoT devices encompass fewer computation resources, for example, the Intel T3K processor on an edge node supports four core CPUs having a meager memory that is not capable of supporting cloud-level services such as Apache Spark [329], with an 8-core CPU, supports 8 GB of memory for operation. Alternatively, Apache Quarks [312] enables real-time analytics on edge nodes; however, it only provides basic data filtering and aggregation capabilities that are not sufficient for current edge nodes.

As the systems deploying IoT devices are increasing continuously, resource constraints like data management, low processing capabilities, smaller memory, and lower battery power are posing a critical challenge on the performance of these systems. In this situation, lightweight algorithms for data filtering, classification, and partitioning (e.g., [14], [154]) are needed to operate on resource-limited devices. Moreover, data filtering can be performed before transmitting the data to the cloud to reduce network resource consumption. Traditional compute-intensive algorithms may become invalid in the IoT context; for instance, the RSA 1024-bit cryptography algorithm cannot be deployed in the IoT context. The resource limitation in IoT requires lightweight security solutions that can operate on IoT infrastructure. Similarly, lightweight VM management techniques are required to orchestrate distributed edge cloudlets' services. IoT realization is based on devising connections and operating with heterogeneous infrastructure where IoT devices are often operated autonomously. Similarly, data produced by the IoT is used to control sophisticated infrastructure. Therefore, lightweight data classification algorithms are required to process the data. Lightweight algorithms are necessary to enable seamless connection, computation offloading, interoperability, and security among the heterogeneous architecture. Moreover, resource limitation in IoT needs the development of lightweight algorithms, OS, and solutions that deal with scarce resources.

G. Lightweight Operating Systems for Software-Defined Internet of Things and Edge Computing Nodes

In SDIoT-Edge, the cloud service nodes lack enough resources as compared to central cloud servers; therefore, lightweight OS for edge nodes is required to enable efficient offloading services. The ideal characteristics for edge OS include less boot time, rapid deployment over diverse platforms, multitasking, less resource consumption, and fewer startup delays [330], [331]. Technologies such as Docker [145] can provide such services; however, these container-based solutions can hardly provide rapid deployments over diverse platforms.

There is a high demand for lightweight OS for low-cost sensor technologies like 5G/6G and advanced LTE. Considering the lower memory footprints in the IoT and edge infrastructure, any OS must be portable, needs lower computational resources, support heterogeneous deployment, and should seamlessly integrate with other solutions, and already available IoT software.

Moreover, the resource-limitation requirements in the IoT, need special attention from OS developers to eliminate the compute-intensive elements from the operating systems. The developers usually provide novel packages that are often called snaps. The snaps are software package-images, which can be downloaded from the multiple network resources instead of an app store. Any OS for IoT should support low-end IoT devices, which suffer from low-computational constraints. Therefore, there is still a need to develop a lightweight OS for IoT and edge nodes, which consider the modularity, scheduling, memory allocation, and network buffer management constraints.

Furthermore, there is a strong need to develop effective maintenance and update mechanisms for SDIoT-Edge solutions. The update patching suffers from reliable installation issues, due to the presence of resource-limited infrastructure. For example, power failure during update patching creates anomalies in the device operation. Therefore, effective rollback mechanisms should be developed for updates' patching in the resource-limited devices [332], [333]. SOA provides software services through message exchange between different layers of IoT, which can be used to develop effective solutions for the updates and maintenance challenges.

H. Unified Architectures

Current research in platform and framework development has been targeted toward fulfilling specific requirements. However, many similar requirements arise due to diverse IoT infrastructure that needs to be addressed for efficient solution development including multiplatform synchronization, communication heterogeneity, distributed integration, and mobility. Consequently, there is a need to develop a unified architecture for versatile devices that will encourage interoperability, ease of synchronization, and leverage standardization. We have discussed many solutions in this paper; however, most of the available architectures do not consider the prevailing IoT demands, which makes it complex to develop an adaptable solution. Generalization among different IoT solutions, such as architectures, protocols for interoperability (e.g., [9]–[11]), and devices will leverage interoperability among IoT for novel applications.

The existing solutions hardly provide distributed integration among edge computing nodes and the IoT infrastructure. A unified architecture using a coordinator might be employed where the function of the coordinator could be periodically querying the edge nodes regarding the available resources, job status, and scheduling. The coordinator can provide seamless operation among distributed edge cloudlets such as discussed in [313]. The location-awareness of the IoT also poses challenges as the computation offloading and transferring results back to the IoT (For example, IoV) needs location assessment of the underlying device. Therefore, a unified architecture employing the mobility-awareness is needed to effectively orchestrate IoT services [314].

I. Security Handling Mechanisms

Edge nodes and IoT devices are part of a decentralized architecture where nodes can join and leave the network at any time. This characteristic makes SDIoT-Edge a security bottleneck, where no single entity acts as a trusted administrator and controls the security of the infrastructure. Data transfer among these heterogeneous devices in the absence of security solutions creates data leakage issues. The resource constraints pose limitations on employing encryption algorithms like 1024-bit RSA [315]. The distributed privacy-preserving techniques like differential privacy can solve the challenge in the multiplatform paradigm. An example of distributed authentication is the gateway authentication at multi-levels [317] and the homomorphic encryption [287]. In the distributed environment, each edge node stores the credentials of every IoT device, which creates inefficient utilization of resources; similarly, a centralized credential repository on a powerful edge node results in communication overhead on the network. To deal with these issues, blockchain technology has recently been proposed, which provides efficient storage for growing records [265], [334]–[337]. Blockchain in IoT can gage the security problems being faced by IoT devices as blockchain allows only trusted participants to interact with each other [338]–[340]. A blockchain-based data-sharing framework uses compute-efficient proof-of-collaboration, transaction filtering, and offloading to reduce storage overhead for robust communication [336]. EdgeChain is a security framework that uses permissioned blockchain and the underlying currency mechanism to securely associate edge cloud resources pool with the IoT devices [319]. Edge nodes can be used to store a log file of records in a distributed manner. This blockchain-based log implementation stores the information of the edge nodes' behavior using exchanged messages between edge nodes and IoT devices. If an edge or IoT device misbehaves, other edge nodes can discover its behavior easily. A decentralized security architecture using SDN blockchain, edge, and fog computing is proposed in [318]. In this architecture, SDN offers continuous monitoring of the network, whereas blockchain provides decentralized security to avoid a single point of failure. In addition, blockchains can be used to develop authentication mechanisms for IoT devices and edge nodes. Blockchains can be utilized to develop a secure layer among edge nodes and IoT infrastructure to ensure security and privacy.

Alternatively, symmetric and asymmetric algorithms can be developed to handle security issues. However, both of these algorithms suffer from potential drawbacks as the symmetric solutions lack in providing sufficient authentication, whereas the asymmetric solutions contain larger key sizes and consume more memory. Therefore, there is still a need to develop cryptography solutions that offer lower-key size, increased processing speed, and need fewer computation resources. Moreover, many device designers use less-secure Bluetooth or ZigBee for the connection where the adversaries can break through the Bluetooth passwords and mac addresses on ZigBee. Therefore, the device designers need to employ standardized Public Key Infrastructure (PKI) authentication methods and utilize standard network protocols like TCP/IP to enhance security from the nodes to the edge infrastructure.

J. Eavesdropping, Flooding, and DDoS Attacks

The attack surface for DDoS and flooding attacks has been increased due to the development of a huge number of smart devices. As the IoT devices contain a small memory, the implementation of device-specific security solutions becomes a challenge. It makes IoT vulnerable to different attacks, including eavesdropping, flooding, and DDoS [65]. The security of the Internet depends on securing the whole Internet infrastructure; so, the network connectivity of attack-vulnerable IoT devices exposes the global network toward attacks. Internet entities, such as hosts, servers, and the service infrastructure contain limited resources that can be saturated by a finite number of users. This fact increases the possibility of DDoS and LFA in the SDIoT-Edge. Moreover, simplified communication models of IoT and non-availability of security solutions create increased vulnerability of attacks in SDIoT-Edge. Lack of resources poses limitations on compute-intensive security solutions; therefore, security frameworks can be developed at the application plane to secure SDIoT-Edge [198], [199]. However, they pose overhead on the network due to the extra packet inspection and mitigation processes. Device authentication using auxiliary edge infrastructure can provide a solution to authenticate resource-limited IoT devices. EdgeSec [341] and ReSIoT [342] ensure security against eavesdropping attacks, which offloads the security function to the edge cloudlets. However, it becomes riskier when edge nodes are down, making the IoT devices prone to attacks. Furthermore, as IoT devices collect imperative data and transfer it to the edge nodes for processing, the possibilities of eavesdrop attacks increases. Although the eavesdropping attacks can be secured using the edge nodes, there is still a need to secure the communication channels. The development of lightweight security protocol for the edge nodes and end devices has great potential in future research. Moreover, enhanced authentication of IoT and standardization of network security protocols will help in securing SDIoT-Edge from eavesdropping, flooding and DDoS Attacks.

K. Controller Bottleneck

In the SDIoT-Edge paradigm, SDN provides scalability by leveraging the performance of the controller, where many studies are available that address the performance of SDN controllers based on different workloads, architectures, and implementations. These studies perform an evaluation of the SDN controller based on different metrics, including link utilization, path installation time, flow rule installation, the throughput of the controller, and latency, which corresponds to the delay in completing a flow request [54], [262], [343]–[345]. Nevertheless, centralized management in SDIoT-Edge provides immense opportunities and benefits; however, key challenges of a single point of failure, OpenFlow channel vulnerabilities caused by flooding flows, and scalability issues are associated with this integration. One of the main aspects of the scalability of the SDN controller is achieved by the separation of planes, which instigates the mechanism of controlling the data plane devices from a different location. As devices at the data plane cannot decide traffic flow, a seamless communication mechanism is necessary between the controller and data plane devices to efficiently manage the network traffic. This presents a communication overhead between the controller and data plane devices depending on the architecture of the network, in addition to applications at the application plane.

The controller is a central component of the network that can be overloaded with extensive flow rule installation requests. This forging of the network flows on the controller make it a bottleneck due to the limitation of the computation resources, e.g., memory and processing power. To address this challenge, distributed controllers can be deployed to avoid the single point of failure. In this situation, enhanced virtualization of network resources using NFV and VM management has the potential to provide an abstraction of a single controller. Moreover, security solutions can be developed to secure the OpenFlow channel from flooding attacks [320]. Another latency requirement is instigated by the difference between the architectural positioning of the control and data plane. The latency of flow setup is associated with the duration of the switch packet processing, the round trip time among the controllers, and the time taken by the controller in handling the request. Higher time consumption in the controller-switch communication, directly affects the flow setup latency, resulting in prolonged time delays in updating (e.g., add, delete, update) the flow rules. This setup causes congestion in the network and ultimately triggers the failure. A well-organized synchronization between the data and control plane enhances the control of the network in handling failover issues.

SDN provides the basis to revisit the deployment of network functions. It promotes the idea of softwarization that supports heterogeneity and dynamicity. Alternatively, the centralized management poses issues of throughput, latency, availability, and single point of failure of the network. Unified architectures for SDIoT-Edge are needed where service provision, security, interoperability, and pricing can be effectively handled. Moreover, security solutions for the diverse SDIoT-Edge ecosystem need to be developed considering constraints from the diversity and resource limitations in the infrastructure.

IX. Conclusion

Programmable networks enable flexible network evolution and management that leverage the SDN characteristics of separation of the control and data plane. A massive increase in IoT infrastructure has been observed due to the advancements in the field of wireless sensor networks. IoT weaves the fabric of the current smart world infrastructure; however, the resource-limited IoT devices bring novel challenges of service orchestration, management, scalability, and heterogeneity. In this context, SDN provides virtualization for effective IoT implementation, whereas edge computing acts as a gateway between latency-sensitive IoT infrastructure and the traditional cloud. Although SDN adaptation with the IoT networks seems promising, there are still many challenges that need to be addressed for an efficient implementation of the SDIoT-Edge. The most critical challenge is the integration of the heterogeneous infrastructure, where the trust, security, and privacy among the computation endpoints needs to be established.

This survey proposed SDN and edge computing for effective IoT service orchestration and infrastructure management. We provide extensive discussion on the areas where SDN can be beneficial for efficient IoT implementation using edge computing. A taxonomy of the available literature has been discussed based on different performance metrics, which can support researchers in selecting the relevant solutions according to their demand. We propose that the standardization in SDIoT-Edge needs extensive consideration due to the heterogeneity in the infrastructure. The security and privacy vulnerabilities arising from the multi-device exposure of data are vital threats to the effective realization of SDIoT-Edge. We postulate that the attack vectors in IoT are greater than the traditional networks due to the lack of security solutions. SDIoT-Edge is the key enabling factor for future generation computing systems because of intense demand for smart devices. Therefore, security, privacy, integration, and standardization requirements need to be adequately established for an effective SDIoT-Edge realization.

References

[1] M. Weiser, “Computer of the 21st century,” *IEEE Pervasive Comput.*, vol. 1, no. 1, pp. 19–25, Jan.–Mar. 2002.

[2] Q. Ni, I. Cleland, C. Nugent, A. B. G. Hernando, and I. P. de la Cruz, “Design and assessment of the data analysis process for a wrist-worn smart object to detect atomic activities in the smart home,” *Pervasive Mobile Comput.*, vol. 56, pp. 57–70, May 2019.

[3] G. Wilson *et al.*, “Robot-enabled support of daily activities in smart home environments,” *Cogn. Syst. Res.*, vol. 54, pp. 258–272, May 2019.

[4] (2018). *50 Billion Connections 2020*. [Online]. Available: http://www.ericsson.com/thecompany/press/releases/2010/04/1403231

[5] L. Atzori *et al.*, “SDN&NFV contribution to IoT objects virtualization,” *Comput. Netw.*, vol. 149, pp. 200–212, Feb. 2019.

[6] B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, “Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges,” *Future Gener. Comput. Syst.*, vol. 92, pp. 718–731, Mar. 2019.

[7] B. W. Wirtz, J. C. Weyerer, and F. T. Schichtel, “An integrative public IoT framework for smart government,” *Govt. Inf. Quart.*, vol. 36, no. 2, pp. 333–345, 2019.

[8] A. Chaudhary and P. Tomar, “Big data and IoT applications in real life environment,” in *Proc. Handbook Res. Big Data IoT*, 2019, pp. 1–21.

[9] S. Habib, J. Qadir, A. Ali, D. Habib, M. Li, and A. Sathiaseelan, “The past, present, and future of transport-layer multipath,” *J. Network Comput. Appl.*, vol. 75, pp. 236–258, Jun. 2016.

[10] J.-P. Vasseur and A. Dunkels, “Chapter 3—Why IP for smart objects?” in *Interconnecting Smart Objects With IP*, J.-P. Vasseur and A. Dunkels, Eds. Boston, MA, USA: Morgan Kaufmann, 2010, pp. 29–38.

[11] M. A. Gallo and W. M. Hancock, “Chapter 3—The Internet and TCP/IP,” in *Networking Explained*, 2nd ed., M. A. Gallo and W. M. Hancock, Eds. Woburn, MA, USA: Digital, 2002, pp. 55–138.

[12] M. Aly, F. Khomh, Y.-G. Guéhéneuc, H. Washizaki, and S. Yacout, “Is fragmentation a threat to the success of the Internet of Things?” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 472–487, Feb. 2019.

[13] Z. Xu, L. Chao, and X. Peng, “T-REST: An open-enabled architectural style for the Internet of Things,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4019–4034, Jun. 2019.

[14] X. Xu *et al.*, “A computation offloading method over big data for IoT-enabled cloud-edge computing,” *Future Gener. Comput. Syst.*, vol. 95, pp. 522–533, Jun. 2019.

[15] L. Tu, S. Liu, Y. Wang, C. Zhang, and P. Li, “An optimized cluster storage method for real-time big data in Internet of Things,” *J. Supercomput.*, to be published.

[16] K. R. Sollins, “IoT big data security and privacy vs. innovation,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1628–1635, Apr. 2019.

[17] A. J. Ferrer, J. M. Marquès, and J. Jorba, “Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing,” *ACM Comput. Surveys*, vol. 51, no. 6, p. 111, 2019.

[18] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[19] L. Baresi, D. Mendonça, M. Garriga, S. Guinea, and G. Quattrocchi, “A unified model for the mobile-edge-cloud continuum,” *ACM Trans. Internet Technol.*, vol. 19, no. 2, p. 29, 2019.

[20] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, “QoS prediction for service recommendations in mobile edge computing,” *J. Parallel Distrib. Comput.*, vol. 127, pp. 134–144, May 2019.

[21] E. Ahmed and M. H. Rehmani, “Mobile edge computing: Opportunities, solutions, and challenges,” *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.

[22] R. Olaniyan, O. Fadahunsi, M. Maheswaran, and M. F. Zhani, "Opportunistic edge computing: Concepts, opportunities and research challenges," *Future Gener. Comput. Syst.*, vol. 89, pp. 633–645, Dec. 2018.

[23] E. Ahmed *et al.*, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017.

[24] A. H. Sodhro, S. Pirbhulal, and V. H. C. de Albuquerque, "Artificial intelligence driven mechanism for edge computing based industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4235–4243, Jul. 2019.

[25] E. Jonas *et al.*, "Cloud programming simplified: A Berkeley view on serverless computing," 2019. [Online]. Available: arXiv:1902.03383.

[26] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging SDN and NFV security mechanisms for IoT systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 812–837, 1st Quart., 2019.

[27] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[28] 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.

[29] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, 4th Quart., 2018.

[30] F. Javed, M. K. Afzal, M. Sharif, and B. Kim, "Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2062–2100, 3rd Quart., 2018.

[31] C. Mouradian *et al.*, "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.

[32] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Comput. Netw.*, vol. 143, no. 2, pp. 221–246, 2018.

[33] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog *et al.*: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, no. 2, pp. 680–698, 2018.

[34] 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.

[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.

[36] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[37] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 56, pp. 684–700, Mar. 2016.

[38] H. Elazhary, "Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions," *J. Netw. Comput. Appl.*, vol. 128, pp. 105–140, Feb. 2019.

[39] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1826–1857, 3rd Quart., 2018.

[40] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: A primer," *Digit. Commun. Netw.*, vol. 4, no. 2, pp. 77–86, 2018.

[41] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.

[42] J. Hendler and J. Golbeck, "METCALFE's law, Web 2.0, and the semantic Web," *Web Semantics Sci. Serveys Agents World Wide Web*, vol. 6, no. 1, pp. 14–20, 2008.

[43] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in defined networks: Threats, taxonomy, and state-of-the-art," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 303–324, 1st Quart., 2017.

[44] K. Poularakis, Q. Qin, E. M. Nahum, M. Rio, and L. Tassiulas, "Flexible SDN control in tactical ad hoc networks," *Ad Hoc Netw.*, vol. 85, pp. 71–80, Mar. 2019.

[45] A. M. Zarca *et al.*, "Security management architecture for NFV/SDN-aware IoT systems," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8005–8020, Oct. 2019.

[46] A. Akhunzada and M. K. Khan, "Toward secure defined vehicular networks: Taxonomy, requirements, and open issues," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 110–118, Jul. 2017.

[47] A. Darabseh and N. M. Freris, "A software-defined architecture for control of IoT cyberphysical systems," *Clust. Comput.*, vol. 22, pp. 1107–1122, Dec. 2018.

[48] Y. Jararweh, M. Al-Ayyoub, and E. Benkhelifa, "An experimental framework for future smart cities using data fusion and software defined systems: The case of environmental monitoring for smart healthcare," *Future Gener. Comput. Syst.*, vol. 107, pp. 883–897, Jun. 2020.

[49] I. Haque, M. Nurujjaman, J. Harms, and N. Abu-Ghazaleh, "SDSense: An agile and flexible SDN-based framework for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1866–1876, Feb. 2019.

[50] I. Alam *et al.*, "IoT virtualization: A survey of software definition & function virtualization techniques for Internet of Things," 2019. [Online]. Available: arXiv:1902.10910.

[51] M. Satyanarayanan *et al.*, "Edge analytics in the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 24–31, Apr.–Jun. 2015.

[52] Y. Jararweh *et al.*, "Software-defined system support for enabling ubiquitous mobile edge computing," *Comput. J.*, vol. 60, no. 10, pp. 1443–1457, 2017.

[53] S. Almajali, H. B. Salameh, M. Ayyash, and H. Elgala, "A framework for efficient and secured mobility of IoT devices in mobile edge computing," in *Proc. 3rd IEEE Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Barcelona, Spain, 2018, pp. 58–62.

[54] R. Sairam, S. S. Bhunia, V. Thangavelu, and M. Gurusamy, "NETRA: Enhancing IoT security using NFV-based edge traffic analysis," *IEEE Sensors J.*, vol. 19, no. 12, pp. 4660–4671, Jun. 2019.

[55] M. Uddin, T. Nadeem, and S. Nukavarapu, "Extreme SDN framework for IoT and mobile applications flexible privacy at the edge," *Heart*, vol. 200, p. 250, Feb. 2019.

[56] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.

[57] A. J. Kadhim and S. A. H. Seno, "Maximizing the utilization of fog computing in Internet of Vehicle using SDN," *IEEE Commun. Lett.*, vol. 23, no. 1, pp. 140–143, Jan. 2019.

[58] J. Dizdarević, F. Carpio, A. Jukan, and X. Masip-Bruin, "A survey of communication protocols for Internet of Things and related challenges of fog and cloud computing integration," *ACM Comput. Surveys*, vol. 51, no. 6, p. 116, 2019.

[59] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of Things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Gener. Comput. Syst.*, vol. 92, pp. 265–275, Mar. 2019.

[60] V. Alfonso, P. Earl, L. Hung, J. F. Hines, and R. M. Satish. (2015). *Predicts 2015: The Internet of Things*. [Online]. Available: https://www.gartner.com/en/documents/2952822

[61] V. Afshar. (2017). *Cisco: Enterprises Are Leading the Internet of Things Innovation*. [Online]. Available: https://www.huffingtonpost.com/entry/cisco-enterprises-are-leading-the-internet-of-things-us-59a41fcee4b0a62d0987b0c6

[62] L. Calderoni, A. Magnani, and D. Maio, "IoT manager: An open-source IoT framework for smart cities," *J. Syst. Archit.*, vol. 98, pp. 413–423, Sep. 2019.

[63] O. Bello and S. Zeadally, "Toward efficient smartification of the Internet of Things (IoT) services," *Future Gener. Comput. Syst.*, vol. 92, pp. 663–673, Jan. 2019.

[64] M. B. M. Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, Jan. 2019.

[65] M. De Donno, N. Dragoni, A. Giaretta, and A. Spognardi, "DDoS-capable IoT malwares: Comparative analysis and Mirai investigation," *Security Commun. Netw.*, vol. 2018, pp. 1–30, Feb. 2018.

[66] C. Bodei, S. Chessa, and L. Galletta, "Measuring security in IoT communications," *Theor. Comput. Sci.*, vol. 764, pp. 100–124, Apr. 2019.

[67] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, and J. Lopez, "A survey of IoT-enabled cyberattacks: Assessing attack paths to critical infrastructures and services," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3453–3495, 4th Quart., 2018.

[68] *Scopus*. Accessed: May 29, 2019. [Online]. Available: www.scopus.com/search/form.uri?display=basic

[69] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[70] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 1–20, Feb. 2017.

[71] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[72] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the Internet of Things (IoT)," Politecnico di Torino, Turin, Italy, Rep., May 2015.

[73] G. M. Lee, P. Jungsoo, K. Ning, and C. Noel. (Jul. 2011). *The Internet of Things—Concept and Problem Statement*. [Online]. Available: tools.ietf.org/html/draft-lee-iot-problem-statement-02

[74] S. Debroy, P. Samanta, A. Bashir, and M. Chatterjee, "SPEED-IoT: Spectrum aware energy efficient routing for device-to-device IoT communication," *Future Gener. Comput. Syst.*, vol. 93, pp. 833–848, Apr. 2019.

[75] P. K. Verma *et al.*, "Machine-to-machine (M2M) communications: A survey," *J. Netw. Comput. Appl.*, vol. 66, pp. 83–105, May 2016.

[76] I. Bojanova, G. Hurlburt, and J. Voas, "Imagineering an Internet of anything," *Computer*, vol. 47, no. 6, pp. 72–77, 2014.

[77] *The Internet of Everything Global Public Sector Economic Analysis*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/business-insights/docs/ioe-value-at-stake-public-sector-analysis-faq.pdf

[78] D. McFarlane. *Industrial Internet of Things Applying IoT in the Industrial Context*. Accessed: Sep. 28, 2018. [Online]. Available: https://www.ifm.eng.cam.ac.uk/uploads/DIAL/industrial-internet-of-things-report.pdf

[79] B. Chen, J. Wan, Y. Lan, M. Imran, D. Li, and N. Guizani, "Improving cognitive ability of edge intelligent IIoT through machine learning," *IEEE Netw.*, vol. 33, no. 5, pp. 61–67, Sep./Oct. 2019.

[80] H. Tang, D. Li, J. Wan, M. Imran, and M. Shoaib, "A reconfigurable method for intelligent manufacturing based on industrial cloud and edge intelligence," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4248–4259, May 2020.

[81] M. H. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera, "The role of big data analytics in industrial Internet of Things," *Future Gener. Comput. Syst.*, vol. 99, pp. 247–259, Oct. 2019.

[82] D. V. Dimitrov, "Medical Internet of Things and big data in healthcare," *Healthcare Inf. Res.*, vol. 22, no. 3, pp. 156–163, 2016.

[83] D. Zeng, S. Guo, and Z. Cheng, "The Web of Things: A survey," *J. Clin. Microbiol.*, vol. 6, no. 6, pp. 424–438, 2011.

[84] M. S. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions," *Comput. Commun.*, vol. 139, pp. 32–57, May 2019.

[85] M. G. Sarowar, M. S. Kamal, and N. Dey, "Internet of Things and its impacts in computing intelligence: A comprehensive review—IoT application for big data," in *Proc. Big Data Anal. Smart Connected Cities*, 2019, pp. 103–136.

[86] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things-based smart cities: Recent advances and challenges," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 16–24, Sep. 2017.

[87] M. Gusev and S. Dustdar, "Going back to the roots—The evolution of edge computing, an IoT perspective," *IEEE Internet Comput.*, vol. 22, no. 2, pp. 5–15, Mar./Apr. 2018.

[88] A. Taivalsaari and T. Mikkonen, "Cloud technologies for the Internet of Things: Defining a research agenda beyond the expected topics," in *Proc. 41st IEEE Euromicro Conf. Softw. Eng. Adv. Appl.*, 2015, pp. 484–488.

[89] X. Zheng, A. Chen, G. Luo, L. Tian, and Z. Cai, "Privacy-preserved distinct content collection in human-assisted ubiquitous computing systems," *Inf. Sci.*, vol. 493, pp. 91–104, Aug. 2019.

[90] S. Goudarzi *et al.*, "A hybrid intelligent model for network selection in the industrial Internet of Things," *Appl. Soft Comput.*, vol. 74, pp. 529–546, Jan. 2019.

[91] R. F. Babiceanu and R. Seker, "Cyber resilience protection for industrial Internet of Things: A software-defined networking approach," *Comput. Ind.*, vol. 104, pp. 47–58, Jan. 2019.

[92] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[93] J. G. An *et al.*, "Towards global IoT-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5753–5765, Jun. 2019.

[94] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6TiSCH: Deterministic IP-enabled industrial Internet (of Things)," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 36–41, Dec. 2014.

[95] L. Doyle. *Facebook, Google Use SDN to Boost Data Center Connectivity*. Accessed: Sep. 28, 2019. [Online]. Available: https://searchnetworking.techtarget.com/tip/Facebook-Google-use-SDN-to-boost-data-center-connectivity

[96] M. Yang, H. Rastegarfar, and I. B. Djordjevic, "Physical-layer adaptive resource allocation in software-defined data center networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 12, pp. 1015–1026, Dec. 2018.

[97] *Software Defined Networking: The New Norm for Networks*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/··· /wp-sdn-newnorm.pdf

[98] F. de Oliveira Silva, J. H. de Souza Pereira, P. F. Rosa, and S. T. Kofuji, "Enabling future Internet architecture research and experimentation by using software defined networking," in *Proc. IEEE Eur. Workshop Softw. Defined Netw.*, 2012, pp. 73–78.

[99] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[100] M. Casado *et al.*, "Rethinking enterprise network control," *IEEE/ACM Trans. Netw.*, vol. 17, no. 4, pp. 1270–1283, Aug. 2009.

[101] L. Wang, Q. Li, R. Sinnott, Y. Jiang, and J. Wu, "An intelligent rule management scheme for software defined networking," *Comput. Netw.*, vol. 144, pp. 77–88, Oct. 2018.

[102] I. I. Awan, N. Shah, M. Imran, M. Shoaib, and N. Saeed, "An improved mechanism for flow rule installation in-band SDN," *J. Syst. Archit.*, vol. 96, pp. 1–19, Feb. 2019.

[103] M. Casado, N. Foster, and A. Guha, "Abstractions for software-defined networks," *Commun. ACM*, vol. 57, no. 10, pp. 86–95, 2014.

[104] K. M. Modieginyane, B. B. Letswamotse, R. Malekian, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportunities for efficient network management: A survey," *Comput. Elect. Eng.*, vol. 66, pp. 274–287, Feb. 2018.

[105] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "SAFE: SDN-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 469–480, Jan. 2019.

[106] F. A. Zaman, A. Jarray, and A. Karmouch, "defined network-based edge cloud resource allocation framework," *IEEE Access*, vol. 7, pp. 10672–10690, 2019.

[107] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[108] *SDN Architecture*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.opennetworking.org/wp-content/uploads /2013/02/TRSDNARCH-1.0-06062014.pdf

[109] W. Zhou, L. Li, M. Luo, and W. Chou, "Rest API design patterns for SDN northbound API," in *Proc. 28th IEEE Int. Conf. Adv. Inf. Netw. Appl. Workshops*, 2014, pp. 358–365.

[110] *OpenFlow Switch Specification Version 1.5.0*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[111] S. A. Hossain, M. A. Rahman, and M. A. Hossain, "Edge computing framework for enabling situation awareness in IoT based smart city," *J. Parallel Distrib. Comput.*, vol. 122, pp. 226–237, Dec. 2018.

[112] I. Farris, A. Orsino, L. Militano, A. Iera, and G. Araniti, "Federated IoT services leveraging 5G technologies at the edge," *Ad Hoc Netw.*, vol. 68, pp. 58–69, Jan. 2018.

[113] M. Satyanarayanan *et al.*, "An open ecosystem for mobile-cloud convergence," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 63–70, Mar. 2015.

[114] 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, 1st Quart., 2015.

[115] B. Varghese, W. Nan, L. I. Jianyu, and D. S. Nikolopoulos, "Edge-as-a-service: Towards distributed cloud architectures," in *Proc. Int. Conf. Parallel Comput. (ParCo)*, Bologna, Italy, 2017, pp. 1–10.

[116] S. Sahhaf *et al.*, "Network service chaining with optimized network function embedding supporting service decompositions," *Comput. Netw.*, vol. 93, no. 3, pp. 492–505, 2015.

[117] *Huawei Observation to NFV*. Accessed: Sep. 28, 2019. [Online]. Available: http://www.huawei.com/ilink/en/download/HW_399662

[118] *European Telecommunications Standards Institute Industry Specifications Group, Mobile-Edge Computing—Network Functions Virtualisation*. Accessed: Oct. 6, 2019. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv

[119] F. B. Jemaa, G. Pujolle, and M. Pariente, "Cloudlet- and NFV-based carrier Wi-Fi architecture for a wider range of services," *Ann. Telecommun.*, vol. 71, nos. 11–12, pp. 1–8, 2016.

[120] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Gener. Comput. Syst.*, vol. 97, pp. 219–235, Aug. 2019.

[121] M. Alenezi, K. Almustafa, and K. A. Meerja, "Cloud based SDN and NFV architectures for IoT infrastructure," *Egypt. Inf. J.*, vol. 20, no. 1, pp. 1–10, 2019.

[122] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surveys*, vol. 51, no. 6, pp. 1–39, 2019.

[123] A. Tzanakaki, M. P. Anastasopoulos, and D. Simeonidou, "Converged optical, wireless, and data center network infrastructures for 5G services," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, no. 2, pp. A111–A122, Feb. 2019.

[124] N. Y. Kim, J. H. Ryu, B. W. Kwon, Y. Pan, and J. H. Park, "CF-CloudOrch: Container fog node-based cloud orchestration for IoT networks," *J. Supercomput.*, vol. 74, no. 12, pp. 7024–7045, 2018.

[125] A. Blenk, A. Basta, M. Reisslein, and W. Kellerer, "Survey on network virtualization hypervisors for software defined networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 655–685, 1st Quart., 2016.

[126] H. A. Alameddine, S. Sharafeddine, S. Sebbah, S. Ayoubi, and C. Assi, "Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 668–682, Mar. 2019.

[127] J. Wang, J. Pan, and F. Esposito, "Elastic urban video surveillance system using edge computing," in *Proc. ACM Workshop Smart Internet Things*, New York, NY, USA, 2017, p. 7.

[128] Y.-B. Lin, H.-C. Tseng, Y.-W. Lin, and L.-J. Chen, "NB-IoTtalk: A service platform for fast development of NB-IoT applications," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 928–939, Feb. 2019.

[129] E. A. Mazied *et al.*, "The wireless control plane: An overview and directions for future research," *J. Netw. Comput. Appl.*, vol. 126, pp. 104–122, Jan. 2019.

[130] J. Huang, Q. Duan, S. Guo, Y. Yan, and S. Yu, "Converged network-cloud service composition with end-to-end performance guarantee," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 545–557, Apr.–Jun. 2018.

[131] B. N. Silva, M. Khan, and K. Han, "Internet of Things: A comprehensive review of enabling technologies, architecture, and challenges," *IETE Tech. Rev.*, vol. 35, no. 2, pp. 205–220, 2018.

[132] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 475–488, Mar. 2018.

[133] A. A. Abdelltif, E. Ahmed, A. T. Fong, A. Gani, and M. Imran, "SDN-based load balancing service for cloud servers," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 106–111, Aug. 2018.

[134] M. K. Jaiswal, "Introduction to OpenFlow," in *Proc. Innov. Softw. Defined Netw. Functions Virtual.*, 2018, pp. 52–71.

[135] S. Secci, P. Raad, and P. Gallard, "Linking virtual machine mobility to user mobility," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 927–940, Dec. 2016.

[136] A. Raza and S. Lee, "Gate switch selection for in-band controlling in software defined networking," *IEEE Access*, vol. 7, pp. 5671–5681, 2019.

[137] R. Muñoz *et al.*, "Integration of IoT, transport SDN, and edge/cloud computing for dynamic distribution of IoT analytics and efficient use of network resources," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1420–1428, Apr. 1, 2018.

[138] H. Huang, J. Zhu, and L. Zhang, "An SDN_based management framework for IoT devices," in *Proc. 25th IET Irish Signals Syst. Conf.*, 2014, pp. 175–179.

[139] J. Ni, X. Lin, and X. S. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 644–657, Mar. 2018.

[140] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, 2015, pp. 1–6.

[141] P. Gope, R. Amin, S. H. Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment," *Future Gener. Comput. Syst.*, vol. 83, pp. 629–637, Jun. 2018.

[142] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in Internet of Things," *Future Gener. Comput. Syst.*, vol. 100, pp. 144–164, Nov. 2019.

[143] Q. Li, X. He, M. Xu, Y. Jiang, and L. Wang, "Unified middlebox model design and deployment with dynamic resources," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 3, pp. 1035–1048, Sep. 2018.

[144] P. Neves *et al.*, "Future mode of operations for 5G—The SelfNet approach enabled by SDN/NFV," *Comput. Stand. Interfaces*, vol. 54, no. 4, pp. 229–246, 2017.

[145] A. J. Ferrer *et al.*, "OPTIMIS: A holistic approach to cloud service provisioning," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 66–77, 2012.

[146] M. Bastam, M. Sabaei, and R. Yousefpour, "A scalable traffic engineering technique in an SDN-based data center network," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 2, 2018, Art. no. e3268.

[147] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[148] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 39–45, May 2018.

[149] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.

[150] M. Mahalingam *et al.*, "Virtual extensible local area network (VXLAN): A framework for overlaying virtualized layer 2 networks over layer 3 networks," Internet Eng. Task Force, RFC 7348, Sep. 2019.

[151] A. Javed, K. Heljanko, A. Buda, and K. Främling, "CEFIoT: A fault-tolerant IoT architecture for edge and cloud," in *Proc. 4th IEEE World Forum Internet Things (WF-IoT)*, Singapore, 2018, pp. 813–818.

[152] C.-A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, Jan.–Mar. 2015.

[153] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Gener. Comput. Syst.*, vol. 70, pp. 138–147, May 2017.

[154] A. A. Abdellatif, A. Emam, C.-F. Chiasserini, A. Mohamed, A. Jaoua, and R. Ward, "Edge-based compression and classification for smart healthcare systems: Concept, implementation and evaluation," *Expert Syst. Appl.*, vol. 117, pp. 1–14, Mar. 2019.

[155] L. Valerio, M. Conti, and A. Passarella, "Energy efficient distributed analytics at the edge of the network for IoT environments," *Pervasive Mobile Comput.*, vol. 51, pp. 27–42, Dec. 2018.

[156] *Put the Most Trusted, Independent Location Data and Technology Platform to Work for Your Business*. Accessed: Sep. 28, 2019. [Online]. Available: https://foursquare.com/

[157] *The Right Information at Just the Right Time*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.google.co.uk/landing/now/

[158] I. Yaqoob, L. U. Khan, S. M. A. Kazmi, M. Imran, N. Guizani, and C. S. Hong, "Autonomous driving cars in smart cities: Recent advances, requirements, and challenges," *IEEE Netw.*, vol. 34, no. 1, pp. 174–181, Jan./Feb. 2020.

[159] E. Ahmed, A. Naveed, A. Gani, S. H. A. Hamid, M. Imran, and M. Guizani, "Process state synchronization-based application execution management for mobile edge/cloud computing," *Future Gener. Comput. Syst.*, vol. 91, pp. 579–589, Feb. 2019.

[160] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE Acess*, vol. 4, pp. 2751–2763, 2016.

[161] C. C. Byers, "Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 14–20, Aug. 2017.

[162] A. Karim *et al.*, "Big data management in participatory sensing: Issues, trends and future directions," *Future Gener. Comput. Syst.*, vol. 107, pp. 942–955, Jun. 2020.

[163] Z. Wen, X. Liu, Y. Xu, and J. Zou, "A restful framework for Internet of Things based on software defined network in modern manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, nos. 1–4, pp. 361–369, 2016.

[164] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1351–1360, Jun. 2018.

[165] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Gener. Comput. Syst.*, vol. 90, pp. 149–157, Jan. 2019.

[166] M. Boussard *et al.*, "The Majord' Home: A SDN approach to let ISPS manage and extend their customers' home networks," in *Proc. 10th IEEE Int. Conf. Netw. Service Manag. (CNSM) Workshop*, Rio de Janeiro, Brazil, 2014, pp. 430–433.

[167] M. Boussard *et al.*, "Software-defined LANs for interconnected smart environment," in *Proc. 27th IEEE Int. Teletraffic Conf.*, 2015, pp. 219–227.

[168] A. Yassine, S. Singh, M. S. Hossain, and G. Muhammad, "IoT big data analytics for smart homes with fog and cloud computing," *Future Gener. Comput. Syst.*, vol. 91, pp. 563–573, Feb. 2019.

[169] O. Briante *et al.*, "A social and pervasive IoT platform for developing smart environments," in *Proc. Internet Things Smart Urban Ecosyst.*, 2019, pp. 1–23.

[170] M. Lee, Y. Kim, and Y. Lee, "A home cloud-based home network autoconfiguration using SDN," in *Proc. 12th IEEE Int. Conf. Netw. Sens. Control*, Taipei, Taiwan, 2015, pp. 444–449.

[171] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," 2017. [Online]. Available: arXiv:1711.05036.

[172] Y. Zhang, H. Zhou, and J.-L. Chen, "Cross-layer access control in publish/subscribe middleware over software-defined networks," *Comput. Commun.*, vol. 134, pp. 1–13, Jan. 2019.

[173] J. Chen, E. Cañete, D. Garrido, M. Díaz, and K. Piotrowski, "PICO: A platform independent communications middleware for heterogeneous devices in smart grids," *Comput. Stand. Interfaces*, vol. 65, pp. 1–14, Jul. 2019.

[174] M. A. da Cruz, J. J. Rodrigues, P. Lorenz, P. Solic, J. Al-Muhtadi, and V. H. C. Albuquerque, "A proposal for bridging application layer protocols to HTTP on IoT solutions," *Future Gener. Comput. Syst.*, vol. 97, pp. 145–152, Aug. 2019.

[175] T. Lin, J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "Enabling SDN applications on software-defined infrastructure," in *Proc. IEEE Netw. Operat. Manag. Symp. (NOMS)*, Krakow, Poland, 2014, pp. 1–7.

[176] A. Galis *et al.*, "Software enabled future Internet—Challenges in orchestrating the future Internet," in *Proc. Int. Conf. Mobile Netw. Manag.*, Tokyo, Japan, 2013, pp. 228–244.

[177] P. Bull, R. Austin, and M. Sharma, "Pre-emptive flow installation for Internet of Things devices within software defined networks," in *Proc. IEEE 3rd Int. Conf. Future Internet Things Cloud*, 2015, pp. 124–130.

[178] V. R. Tadinada, "Software defined networking: Redefining the future of Internet in IoT and cloud era," in *Proc. IEEE Int. Conf. Future Internet Things Cloud*, Barcelona, Spain, 2014, pp. 296–301.

[179] E. Patouni, A. Merentitis, P. Panagiotopoulos, A. Glentis, and N. Alonistioti, "Network virtualisation trends: Virtually anything is possible by connecting the unconnected," in *Proc. IEEE Conf. SDN Future Netw. Services (SDN4FNS)*, 2013, pp. 1–7.

[180] A.-C. G. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "Towards a software-defined network operating system for the IoT," in *Proc. 2nd IEEE Int. World Forum Internet Things (WF-IoT)*, 2015, pp. 579–584.

[181] D. Bendouda, A. Rachedi, and H. Haffaf, "Programmable architecture based on software defined network for Internet of Things: Connected dominated sets approach," *Future Gener. Comput. Syst.*, vol. 80, pp. 188–197, Mar. 2018.

[182] Y. B. Zikria, H. Yu, M. K. Afzal, M. H. Rehmani, and O. Hahm, "Internet of Things (IoT): Operating system, applications and protocols design, and validation techniques," *Future Gener. Comput. Syst.*, vol. 88, pp. 699–706, Nov. 2018.

[183] A.-C. G. Anadiotis, S. Milardo, G. Morabito, and S. Palazzo, "Toward unified control of networks of switches and sensors through a network operating system," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 895–904, Apr. 2018.

[184] P. Martinez-Julia and A. F. Skarmeta, "Empowering the Internet of Things with software defined networking," in *Proc. FP7 Eur. Res. Future Internet Things*, 2014, pp. 1–4.

[185] M. Ojo, D. Adami, and S. Giordano, "A SDN-IoT architecture with NFV implementation," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 2016, pp. 1–6.

[186] P. Hu, "A system architecture for software-defined industrial Internet of Things," in *Proc. IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Montreal, QC, Canada, 2015, pp. 1–5.

[187] R. Morabito, I. Farris, A. Iera, and T. Taleb, "Evaluating performance of containerized IoT services for clustered devices at the network edge," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1019–1030, Aug. 2017.

[188] Y. Jararweh *et al.*, "SDIoT: A software defined based Internet of Things framework," *J. Ambient Intell. Humanized Comput.*, vol. 6, no. 4, pp. 453–461, 2015.

[189] Q. Xiaofeng, L. Wenmao, G. Teng, H. Xinxin, W. Xutao, and C. Pengcheng, "WOT/SDN: Web of Things architecture using SDN," *China Commun.*, vol. 12, no. 11, pp. 1–11, 2015.

[190] M. Steiner, G. Tsudik, and M. Waidner, "Diffie–Hellman key distribution extended to group communication," in *Proc. 3rd ACM Conf. Comput. Commun. Security*, Delhi, India, 1996, pp. 31–37.

[191] X. Zhang *et al.*, "Reliable multiservice delivery in fog-enabled VANETs: Integrated misbehavior detection and tolerance," *IEEE Access*, vol. 7, pp. 95762–95778, 2019.

[192] J.-W. Xu, K. Ota, M.-X. Dong, A.-F. Liu, and Q. Li, "SIoTFog: Byzantine-resilient IoT fog networking," *Front. Inf. Technol. Electron Eng.*, vol. 19, no. 12, pp. 1546–1557, 2018.

[193] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[194] A. J. Jara, S. Varakliotis, A. F. Skarmeta, and P. Kirstein, "Extending the Internet of Things to the future Internet through IPV6 support," *Mobile Inf. Syst.*, vol. 10, no. 1, pp. 3–17, 2014.

[195] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPV6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals," IETF, Fremont, CA, USA, Rep. RFC 4919, 2007.

[196] B. T. De Oliveira, L. B. Gabriel, and C. B. Margi, "TinySDN: Enabling multiple controllers for software-defined wireless sensor networks," *IEEE Latin America Trans.*, vol. 13, no. 11, pp. 3690–3696, Nov. 2015.

[197] Q. Duan, Y. Yan, and A. V. Vasilakos, "A survey on service-oriented network virtualization toward convergence of networking and cloud computing," *IEEE Trans. Netw. Service Manag.*, vol. 9, no. 4, pp. 373–392, Dec. 2012.

[198] W. Rafique, M. Khan, N. Sarwar, and W. Dou, "A security framework to protect edge supported defined Internet of Things infrastructure," in *Proc. 15th EAI Int. Conf. Collaborative Comput. Netw. Appl. Worksharing*, London, U.K., 2019, pp. 71–88.

[199] W. Rafique, X. He, Z. Liu, Y. Sun, and W. Dou, "CFADefense: A security solution to detect and mitigate crossfire attacks in software-defined IoT-edge infrastructure," in *Proc. 21st IEEE Int. Conf. High Perform. Comput. Commun. IEEE 17th Int. Conf. Smart City IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, 2019, pp. 500–509.

[200] P. Garraghan, X. Ouyang, R. Yang, D. McKee, and J. Xu, "Straggler root-cause and impact analysis for massive-scale virtualized cloud datacenters," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 91–104, Jan./Feb. 2019.

[201] Y. Liu, H.-N. Dai, H. Wang, M. Imran, X. Wang, and M. Shoaib, "UAV-enabled data acquisition scheme with directional wireless energy transfer for Internet of Things," *Comput. Commun.*, vol. 155, pp. 184–196, Apr. 2020.

[202] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "ENORM: A framework for edge node resource management," *IEEE Trans. Services Comput.*, early access, Sep. 18, 2017, doi: 10.1109/TSC.2017.2753775.

[203] M. Körner, T. M. Runge, A. Panda, S. Ratnasamy, and S. Shenker, "Open carrier interface: An open source edge computing framework," in *Proc. Workshop Netw. Emerg. Appl. Technol.*, Budapest, Hungary, 2018, pp. 27–32.

[204] *The Linux Foundation Projects*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.edgexfoundry

[205] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st ed. ACM MCC Workshop Mobile Cloud Comput.*, New York, NY, USA, 2012, pp. 13–16.

[206] B. Varghese, N. Wang, J. Li, and D. S. Nikolopoulos, "Edge-as-a-service: Towards distributed cloud architectures," 2017. [Online]. Available: arXiv:1710.10090.

[207] B.-W. Chen, M. Imran, N. Nasser, and M. Shoaib, "Self-aware autonomous city: From sensing to planning," *IEEE Commun. Mag.*, vol. 57, no. 4, pp. 33–39, Apr. 2019.

[208] H. Teng *et al.*, "A novel code data dissemination scheme for Internet of Things through mobile vehicle of smart cities," *Future Gener. Comput. Syst.*, vol. 94, pp. 351–367, May 2019.

[209] "Charging infrastructure for electric vehicles in Germany progress report and recommendations 2015," Nationale Platform Elektromobilitat, Berlin, Germany, Rep., 2015.

[210] H. El-Sayed *et al.*, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, 2018.

[211] T. Hayajneh, K. Griggs, M. Imran, and B. J. Mohd, "Secure and efficient data delivery for fog-assisted wireless body area networks," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1289–1307, 2019.

[212] M. I. Razzak, M. Imran, and G. Xu, "Big data analytics for preventive medicine," *Neural Comput. Appl.*, vol. 32, pp. 4417–4451, May 2020.

[213] G. Fortino *et al.*, "Towards multi-layer interoperability of heterogeneous IoT platforms: The inter-IoT approach," in *Proc. Integr. Interconnect. Interoper. IoT Syst.*, 2018, pp. 199–232.

[214] C. Gomez, A. Arcia-Moret, and J. Crowcroft, "TCP in the Internet of Things: From ostracism to prominence," *IEEE Internet Comput.*, vol. 22, no. 1, pp. 29–41, Jan./Feb. 2018.

[215] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, p. 44, 2016.

[216] *Internet of Things Related Standards*. Accessed: Sep. 28, 2019. [Online]. Available: http://standards.ieee.org/innovate/iot/stds.html

[217] (2019). *Connecting Devices Where We Live and Work*. [Online]. Available: https://www.threadgroup.org/thread-group

[218] *Open Connectivity Foundation: Unlocking the Massive Opportunity in the Internet of Things*. Accessed: Sep. 28, 2019. [Online]. Available: http://openinterconnect.org/

[219] R. Stackowiak, A. Licht, V. Mantha, and L. Nagode, *Big Data and the Internet of Things: Enterprise Information Architecture for a New Age*. Berkeley, CA, USA: Apress, 2015.

[220] M. Iglesias-Urkia, D. Casado-Mansilla, S. Mayer, and A. Urbieta, "Validation of a COAP to IEC 61850 mapping and benchmarking vs. HTTP-rest and WS-SOAP," in *Proc. 23rd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1. Funchal, Portugal, 2018, pp. 1015–1022.

[221] C.-S. Park and W.-S. Park, "A group-oriented DTLS handshake for secure IoT applications," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1920–1929, Oct. 2018.

[222] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz, "Bluetooth 5: A concrete step forward toward the IoT," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 125–131, Jul. 2018.

[223] *P1903.2—Standard for Service Composition Protocols of Next Generation Service Overlay Network (NGSON)*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1903.2.html

[224] *Standard for Self-Organizing Management Protocols of Next Generation Service Overlay Network (NGSON)*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1903.3.html

[225] *P1903.1—Standard for Content Delivery Protocols of Next Generation Service Overlay Network (NGSON)*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1903.1.html

[226] *P1915.1—Standard for Software Defined Networking and Network Function Virtualization Security*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1915.1.html

[227] *P1916.1—Standard for Software Defined Networking and Network Function Virtualization Performance*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1916.1.html

[228] *P1917.1—Standard for Software Defined Networking and Network Function Virtualization Reliability*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1917.1.html

[229] *P1913.1—Standard for Software Defined Quantum Communication*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/develop/project/1913.1.html

[230] *IEEE P1921.1—Software-Defined Networking Bootstrapping Procedures*. Accessed: Sep. 28, 2019. [Online]. Available: http://standards.ieee.org/develop/project/1921.1.html

[231] *IEEE P1930.1—SDN Based Middleware for Control and Management of Networks*. Accessed: Sep. 28, 2019. [Online]. Available: https://standards.ieee.org/project/1930.1.html

[232] *IEEE P802.1CF—Recommended Practice for Network Reference Model and Functional Description of IEEE 802 Access Network*. Accessed: Sep. 28, 2019. [Online]. Available: http://standards.ieee.org/develop/project/802.1CF.html

[233] K. Gray and T. D. Nadeau, *Network Function Virtualization*, A. Ivernizzi, Ed. Cambridge, MA, USA: Morgan Kaufmann, 2016.

[234] J. Halpern *et al.*, "Service function chaining (SFC) architecture," Internet Eng. Task Force, RFC 7665, 2015.

[235] G. Lopez-Millan, R. Marin-Lopez, and F. Pereniguez-Garcia, "Towards a standard SDN-based IPsec management framework," *Comput. Stand. Interfaces*, vol. 66, Oct. 2019, Art. no. 103357.

[236] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Review*, vol. 44, no. 3, pp. 87–95, 2014.

[237] H. Hromic *et al.*, "Real time analysis of sensor data for the Internet of Things by means of clustering and event processing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., 2015, pp. 685–691.

[238] S. A. De Chaves, R. B. Uriarte, and C. B. Westphall, "Toward an architecture for monitoring private clouds," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 130–137, Dec. 2011.

[239] C. Meurisch, A. Seeliger, B. Schmidt, I. Schweizer, F. Kaup, and M. Mühlhäuser, "Upgrading wireless home routers for enabling large-scale deployment of cloudlets," in *Proc. Int. Conf. Mobile Comput. Appl. Services (MOBICASE)*, Berlin, Germany, 2015, pp. 12–29.

[240] M. Shiraz and A. Gani, "A lightweight active service migration framework for computational offloading in mobile cloud computing," *J. Supercomput.*, vol. 68, no. 2, pp. 978–995, 2014.

[241] W. Li, Y. Zhao, S. Lu, and D. Chen, "Mechanisms and challenges on mobility-augmented service provisioning for mobile cloud computing," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 89–97, Mar. 2015.

[242] J. Povedano-Molina, J. M. Lopez-Vega, J. M. Lopez-Soler, A. Corradi, and L. Foschini, "DARGOS: A highly adaptable and scalable monitoring architecture for multi-tenant clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 8, pp. 2041–2056, 2013.

[243] 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.

[244] *OpenFog Reference Architecture for Fog Computing*, OpenFog Consortium, Fremont, CA, USA, 2017.

[245] *Octeon Fusion-mIntegrated Baseband Processors*. Accessed: Apr. 3, 2020. [Online]. Available: https://rethinkresearch.biz/articles/cisco-pushes-iot-analytics-extreme-edge-mist-computing-2/

[246] A. Yousefpour *et al.*, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Architect.*, vol. 98, pp. 289–330, Sep. 2019.

[247] M. Aazam, E.-N. Huh, and M. St-Hilaire, "Towards media inter-cloud standardization–evaluating impact of cloud storage heterogeneity," *J. Grid Comput.*, vol. 16, no. 3, pp. 425–443, 2018.

[248] T. Halabi and M. Bellaiche, "A broker-based framework for standardization and management of cloud security-SLAS," *Comput. Security*, vol. 75, pp. 59–71, Jun. 2018.

[249] R. Gracia-Tinedo *et al.*, "Giving wings to your data: A first experience of personal cloud interoperability," *Future Gener. Comput. Syst.*, vol. 78, pp. 1055–1070, Jan. 2018.

[250] A. Limaye and T. Adegbija, "A workload characterization of the SPEC CPU2017 benchmark suite," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, 2018, pp. 149–158.

[251] *European Telecommunications Standards Institute Industry Specifications Group. Mobile-Edge Computing—MEC Metrics Best Practice and Guidelines*. Accessed: Sep. 28, 2019. [Online]. Available: http://www.etsi.org/deliver/etsi-gs/MEC-IEG/001099/004/01.01.0160/gs-MEC-IEG004v010101p.pdf

[252] *European Telecommunications Standards Institute Industry Specifications Group, MEC Proofs of Concept*. Accessed: Sep. 28, 2019. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing/mec-poc

[253] *European Telecommunications Standards Institute, POC 2 Edge Video Orchestration and Video Clip Replay*. Accessed: Sep. 28, 2019. [Online]. Available: http://mecwiki.etsi.org/index.php?title=PoC-2-Edge-Video-Orchestration-and-Video-Clip-Replay-via-MEC

[254] *European Telecommunications Standards Institute, POC 7 Multi-Service MEC Platform for Advanced Service Delivery*," Accessed: Sep. 28, 2019. [Online]. Available: http://mecwiki.etsi.org/index.php?title=PoC-7-Multi-Service-MEC-Platform-for-Advanced-Service-Delivery

[255] S. Deng, X. Gao, Z. Lu, Z. Li, and X. Gao, "DoS vulnerabilities and mitigation strategies in software-defined networks," *J. Netw. Comput. Appl.*, vol. 125, pp. 209–219, Jan. 2019.

[256] C. Li, Z. Qin, E. Novak, and Q. Li, "Securing SDN infrastructure of IoT–fog networks from MITM attacks," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1156–1164, Oct. 2017.

[257] R. McRee, "Microsoft threat modeling tool 2014: Identify & mitigate," *ISSA J.*, vol. 39, p. 42, Feb. 2014.

[258] Z. Khan, Z. Pervez, and A. G. Abbasi, "Towards a secure service provisioning framework in a smart city environment," *Future Gener. Comput. Syst.*, vol. 77, pp. 112–135, Dec. 2017.

[259] L. Liu, Z. Ma, and W. Meng, "Detection of multiple-mix-attack malicious nodes using perceptron-based trust in IoT networks," *Future Gener. Comput. Syst.*, vol. 101, pp. 865–879, Dec. 2019.

[260] R. A. Sowah, K. B. Ofori-Amanfo, G. A. Mills, and K. M. Koumadi, "Detection and prevention of man-in-the-middle spoofing attacks in MANETs using predictive techniques in artificial neural networks (ANN)," *J. Comput. Netw. Commun.*, vol. 2019, pp. 1–14, Jan. 2019.

[261] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[262] A. Muthanna *et al.*, "Secure and reliable IoT networks using fog computing with software-defined networking and blockchain," *J. Sensor Actuator Netw.*, vol. 8, no. 1, p. 15, 2019.

[263] M. Badra and R. Borghol, "Long-term integrity and non-repudiation protocol for multiple entities," *Sustain. Cities Soc.*, vol. 40, pp. 189–193, Jul. 2018.

[264] F. Wang, L. Xu, H. Wang, and Z. Chen, "Identity-based non-repudiable dynamic provable data possession in cloud storage," *Comput. Elect. Eng.*, vol. 69, pp. 521–533, Jul. 2018.

[265] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019.

[266] J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, and J. Zheng, "Disrupting SDN via the data plane: A low-rate flow table overflow attack," in *Proc. Int. Conf. Security Privacy Commun. Syst.*, 2017, pp. 356–376.

[267] D. Yin, L. Zhang, and K. Yang, "A DDoS attack detection and mitigation with software-defined Internet of Things framework," *IEEE Access*, vol. 6, pp. 24694–24705, 2018.

[268] M. Ambrosin, M. Conti, F. De Gaspari, and R. Poovendran, "LineSwitch: Tackling control plane saturation attacks in software-defined networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1206–1219, Apr. 2017.

[269] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other Botnets," *Comput.*, vol. 50, no. 7, pp. 80–84, 2017.

[270] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu, "Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1838–1853, Jul. 2018.

[271] K. Bhardwaj, J. C. Miranda, and A. Gavrilovska, "Towards IoT-DDoS prevention using edge computing," in *Proc. USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, Boston, MA, USA, 2018, pp. 1578–1584.

[272] G. Han, H. Wang, J. Jiang, W. Zhang, and S. Chan, "CASLP: A confused arc-based source location privacy protection scheme in WSNs for IoT," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 42–47, Sep. 2018.

[273] L. Chen *et al.*, "Robustness, security and privacy in location-based services for future IoT: A survey," *IEEE Access*, vol. 5, pp. 8956–8977, 2017.

[274] J. Wang, R. Wen, J. Li, F. Yan, B. Zhao, and F. Yu, "Detecting and mitigating target link-flooding attacks using SDN," *IEEE Trans. Depend. Secure Comput.*, vol. 16, no. 6, pp. 944–956, Nov./Dec. 2019.

[275] L. Xue, X. Ma, X. Luo, E. W. W. Chan, T. T. Miu, and G. Gu, "LinkScope: Toward detecting target link flooding attacks," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2423–2438, Oct. 2018.

[276] L. Wang, Q. Li, Y. Jiang, X. Jia, and J. Wu, "WoodPecker: Detecting and mitigating link-flooding attacks via SDN," *Comput. Netw.*, vol. 147, pp. 1–13, Dec. 2018.

[277] X. Ma, J. Li, Y. Tang, B. An, and X. Guan, "Protecting Internet infrastructure against link flooding attacks: A techno-economic perspective," *Inf. Sci.*, vol. 479, pp. 486–502, Apr. 2019.

[278] C. Liaskos and S. Ioannidis, "Network topology effects on the detectability of crossfire attacks," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 7, pp. 1682–1695, Jul. 2018.

[279] C. Baker. *Recent IoT-Based Attacks: What Is the Impact on Managed DNS Operators?* Accessed: Nov. 2, 2019. [Online]. Available: http://dyn.com/blog/dyn-analysis-summary-of-fridayoctober-21-attack/

[280] M. Lyu, D. Sherratt, I. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, "Quantifying the reflective DDoS attack capability of household IoT devices," in *Proc. 10th ACM Conf. Security Privacy Wireless Mobile Netw.*, Montreal, QC, Canada, 2017, pp. 46–51.

[281] T. L. Seals. *IoT Botnet Bursts on the Scene With Massive DDoS—Attack*. Accessed: Sep. 28, 2019. [Online]. Available: https://www.infosecurity-magazine.com/news/leet-iot-botnet-bursts-on-the-scene/

[282] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, and Z. Anwar, "CyberPulse: A machine learning based link flooding attack mitigation system for defined networks," *IEEE Access*, vol. 7, pp. 34885–34899, 2019.

[283] B. Yan, Y. Xu, and H. J. Chao, "BigMaC: Reactive network-wide policy caching for SDN policy enforcement," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2675–2687, Dec. 2018.

[284] C. Li *et al.*, "Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN," *Int. J. Commun. Syst.*, vol. 31, no. 5, 2018, Art. no. e3497.

[285] J. Wan, J. Li, M. Imran, and D. Li, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3652–3660, Jun. 2019.

[286] M. A. Amanullah *et al.*, "Deep learning and big data technologies for IoT security," *Comput. Commun.*, vol. 151, pp. 495–517, Feb. 2020.

[287] D. Li, Q. Yang, W. Yu, D. An, X. Yang, and W. Zhao, "A strategy-proof privacy-preserving double auction mechanism for electrical vehicles demand response in microgrids," in *Proc. 36th IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, San Diego, CA, USA, 2017, pp. 1–8.

[288] F. Chen, T. Xiang, X. Fu, and W. Yu, "User differentiated verifiable file search on the cloud," *IEEE Trans. Services Comput.*, vol. 11, no. 6, pp. 948–961, Nov./Dec. 2018.

[289] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Proc. Conf. Adv. Cryptology CRYPTO*, 2010, pp. 465–482.

[290] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, 2013, pp. 238–252.

[291] J. Clemens, R. Pal, and P. Philip, "Extending trust and attestation to the edge," in *Proc. IEEE/ACM Int. Symp. Edge Comput. (SEC)*, Washington, DC, USA, 2016, pp. 101–102.

[292] S. Echeverría, D. Klinedinst, K. Williams, and G. A. Lewis, "Establishing trusted identities in disconnected edge environments," in *Proc. IEEE/ACM Int. Symp. Edge Comput. (SEC)*, Washington, DC, USA, 2016, pp. 51–63.

[293] J. Kang, R. Yu, X. Huang, and Y. Zhang, 'Privacy-preserved pseudonym scheme for fog computing supported Internet of Vehicles," *IEEE Trans. Intelligent Transport. Syst.*, vol. 19, no. 8, pp. 2627–2637, Aug. 2018.

[294] Q. Kong, R. Lu, M. Ma, and H. Bao, "A privacy-preserving sensory data sharing scheme in Internet of Vehicles," *Future Gener. Comput. Syst.*, vol. 92, pp. 644–655, Mar. 2019.

[295] G. Wang, J. He, X. Shi, J. Pan, and S. Shen, "Analyzing and evaluating efficient privacy-preserving localization for pervasive computing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2993–3007, Aug. 2018.

[296] A. M. Alberti, G. D. Scarpioni, V. J. Magalhães, A. S. Cerqueira, Jr., J. J. P. C. Rodrigues, and R. da Rosa Righi, "Advancing NovaGenesis architecture towards future Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 215–229, Feb. 2019.

[297] *Proofpoint Uncovers Internet of Things (IoT)*. Accessed: Oct. 31, 2019. [Online]. Available: https://www.marketwatch.com/story/proofpoint-uncovers-internet-of-things-iot-cybera

[298] *13th World Wide Infrastructure Security Report*. Accessed: Nov. 1, 2019. [Online]. Available: https://pages.arbornetworks.com/rs/082-KNA-087/images/13th_Worldwide_Infrastructure_Security_Report.pdf

[299] S. Khandelwal. *Friday's Massive DDoS Attack Came From Just 100 000 Hacked IoT Devices*. Accessed: Oct. 31, 2019. [Online]. Available: https://thehackernews.com/2016/10/ddos-attack-mirai-iot.html

[300] C. Frank, C. Nance, S. Jarocki, and W. E. Pauli, "Protecting IoT from MIRAI botnets; IoT device hardening," *J. Inf. Syst. Appl. Res.*, vol. 11, no. 2, p. 33, 2018.

[301] *SmartCam Products: SNH-p6410bn*. Accessed: Oct. 15, 2019. [Online]. Available: https://www.samsungsmartcam.com/web/

[302] *Samsung Smart Things Hub*. Accessed: Oct. 22, 2019. [Online]. Available: https://www.smarthhings.com/works-with-smarthings/hubs-and-kits/samsung-smarthhings-hub/

[303] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput. Pract. Exp.*, vol. 28, no. 10, pp. 2991–3005, 2016.

[304] *New DDoS Attack LFA: From May 11th Netease Attacked*. Accessed: Oct. 30, 2019. [Online]. Available: www.freebuf.com/articles/network/67107.html

[305] A. Studer and A. Perrig, "The coremelt attack," in *Proc. Eur. Symp. Res. Comput. Security*, Saint Malo, France, 2009, pp. 37–52.

[306] M. S. Kang, S. B. Lee, and V. D. Gligor, "The crossfire attack," in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, 2013, pp. 127–141.

[307] L. John. *Biggest DDoS Attack in History Hammers Spamhaus*. Accessed: Oct. 7, 2019. [Online]. Available: http://www.theregister.co.uk/2013/03/27/spamhaus/ddos/megaflood

[308] K. Beckmann and M. Thoss, "A wireless sensor network protocol for the OMG data distribution service," in *Proc. 10th Int. Workshop Intell. Sol. Embedded Syst.*, 2012, pp. 45–50.

[309] C. Anglano, R. Gaeta, and M. Grangetto, "Securing coding-based cloud storage against pollution attacks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1457–1469, May 2017.

[310] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[311] B. Ramachandran, S. K. Srivastava, C. S. Edrington, and D. A. Cartes, "An intelligent auction scheme for smart grid market using a hybrid immune algorithm," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4603–4612, Oct. 2011.

[312] *Join the Community Accelerating Analytics at the Edge*. Accessed: Sep. 25, 2019. [Online]. Available: https://quarks-edge.github.io/

[313] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao, "Vehicle control system coordinated between cloud and mobile edge computing," in *Proc. 55th IEEE Annu. Conf. Soc. Instrum. Control Eng. Japan (SICE)*, 2016, pp. 1122–1127.

[314] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet," *Mobile Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[315] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *J. Ambient Intell. Humanized Comput.*, to be published.

[316] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, 2016, pp. 192–203.

[317] T. W. Chim, S.-M. Yiu, V. O. K. Li, L. C. K. Hui, and J. Zhong, "PRGA: Privacy-preserving recording gateway-assisted authentication of power usage information for smart grid," *IEEE Trans. Depend. Secure Comput.*, vol. 12, no. 1, pp. 85–97, Jan./Feb. 2015.

[318] S. Rathore, B. W. Kwon, and J. H. Park, "BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network," *J. Netw. Comput. Appl.*, vol. 143, pp. 167–177, Oct. 2019.

[319] J. Pan, J. Wang, A. Hester, I. Alqerm, Y. Liu, and Y. Zhao, "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4719–4732, Jun. 2019.

[320] G. Shang, P. Zhe, B. Xiao, A. Hu, and K. Ren, "FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, 2017, pp. 1–9.

[321] *Multi-Access Edge Computing (MEC)*. Accessed: Nov. 1, 2019. [Online]. Available: https://networks.nokia.com/solutions/multi-access-edge-computing

[322] *Octeon Fusion-M Integrated Baseband Processors*. Accessed: Nov. 1, 2019. [Online]. Available: https://www.marvell.com/embedded-processors/base-station-processors/octeon-fusion-m/index.jsp

[323] *Octeon Fusion-M Integrated Baseband Processors*. Accessed: Nov. 1, 2019. [Online]. Available: https://www.cisco.com/c/en/us/products/cloud-systems-management/iox/index.html

[324] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proc. IEEE Int. Conf. Intell. Syst. Control*, 2016, pp. 1–8.

[325] L. Zhang *et al.*, "Named data networking (NDN) project," Elect. Eng., Univ. California at Los Angeles, Los Angeles, CA, USA, Rep. NDN-0001, 2010.

[326] X. Li, J. Wan, H.-N. Dai, M. Imran, M. Xia, and A. Celesti, "A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4225–4234, Jul. 2019.

[327] I. F. Akyildiz, P. Wang, and S. C. Lin, "SoftAir: A software defined networking architecture for 5G wireless systems," *Comput. Netw.*, vol. 85, pp. 1–18, Jul. 2015.

[328] L. Tello-Oquendo, S.-C. Lin, I. F. Akyildiz, and V. Pla, "Software-defined architecture for QoS-aware IoT deployments in 5G systems," *Ad Hoc Netw.*, vol. 93, 2019, Art. no. 101911.

[329] A. Li, X. Zong, S. Kandula, X. Yang, and M. Zhang, "CloudProphet: Towards application performance prediction in cloud," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, 2011, pp. 426–427.

[330] L. Xu, Z. Wang, and W. Chen, "The study and evaluation of arm-based mobile virtualization," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 7, 2015, Art. no. 310308.

[331] M. Silva, A. Tavares, T. Gomes, and S. Pinto, "ChameLIoT: An agnostic operating system framework for reconfigurable IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1291–1292, Feb. 2019.

[332] W. Rafique, X. Zhao, S. Yu, I. Yaqoob, M. Imran, and W. Dou, "An application development framework for Internet of Things service orchestration," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4543–4556, May 2020.

[333] W. Rafique, M. Khan, and W. Dou, "Maintainable solution development using collaboration between architecture and requirements in heterogeneous IoT paradigm (short paper)," in *Proc. 15th EAI Int. Conf. Collaborative Comput. Netw. Appl. Worksharing*, 2019, pp. 489–508.

[334] M. Bartoletti, B. Bellomy, and L. Pompianu, "A journey into bitcoin metadata," *J. Grid Comput.*, vol. 17, pp. 1–20, Jan. 2019.

[335] P. K. Sharma, M. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.

[336] P. K. Sharma, S. Rathore, Y.-S. Jeong, and J. H. Park, "SoftEdgeNet: SDN based energy-efficient distributed network architecture for edge computing," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 104–111, Dec. 2018.

[337] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[338] S. Moin, A. Karim, Z. Safdar, K. Safdar, E. Ahmed, and M. Imran, "Securing IoTs in distributed blockchain: Analysis, requirements and open issues," *Future Gener. Comput. Syst.*, vol. 100, pp. 325–343, Nov. 2019.

[339] S. Hakak, W. Z. Khan, G. A. Gilkar, M. Imran, and N. Guizani, "Securing smart cities through blockchain technology: Architecture, requirements, and challenges," *IEEE Netw.*, vol. 34, no. 1, pp. 8–14, Jan./Feb. 2020.

[340] Z. Zheng *et al.*, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020.

[341] K. Sha, R. Errabelly, W. Wei, T. A. Yang, and Z. Wang, "EdgeSec: Design of an edge layer security service to enhance IoT security," in *Proc. 1st IEEE Int. Conf. Fog Edge Comput.*, Madrid, Spain, 2017, pp. 81–88.

[342] R. H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, "Reconfigurable security: Edge-computing-based framework for IoT," *IEEE Netw.*, vol. 32, no. 5, pp. 92–99, Sep./Oct. 2018.

[343] J. Ni, X. Lin, and X. S. Shen, "Toward edge-assisted Internet of Things: From security and efficiency perspectives," *IEEE Netw.*, vol. 33, no. 2, pp. 50–57, Mar./Apr. 2019.

[344] Q. Miao, W. Jing, and H. Song, "Differential privacy-based location privacy enhancing in edge computing," *Concurrency Comput. Pract. Exp.*, vol. 31, no. 8, 2019, Art. no. e4735.

[345] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Dec. 2017.

Wajid Rafique received the B.S. degree in computer science from the Virtual University of Pakistan and the M.S. degree in software engineering from the National University of Sciences and Technology, Pakistan. He is currently pursuing the Ph.D. degree in computer science with Nanjing University, China. He is also conducting research in collaboration with the Victoria University, Melbourne, VIC, Australia. His research works have been appeared in several prestigious international journals and the top-tier conferences. His research interests include big data services, machine learning, mobile cloud computing, and attacks–defense in virtualized network infrastructure, including IoT and SDN and network security.

Lianyong Qi (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, China, in 2011. In 2010, he visited the Department of Information and Communication Technology, Swinburne University of Technology, Australia. He is currently a Full Professor with the School of Information Science and Engineering, Qufu Normal University, China. He has published over 70 research papers. His research interests include big data and recommender systems.

Ibrar Yaqoob (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Malaya, Malaysia, in 2017. He is a Research Professor with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, where he completed his postdoctoral fellowship under the prestigious grant of Brain Korea 21st Century Plus. He worked as a Researcher and a Developer with the Centre for Mobile Cloud Computing Research, University of Malaya. His numerous research articles are very famous and among the most downloaded in top journals. He has reviewed over 200 times for the top ISI-Indexed journals and conferences. His research interests include big data, edge computing, mobile cloud computing, Internet of Things, and computer networks. He has been listed among top researchers by Thomson Reuters (Web of Science) based on the number of citations earned in last three years in six categories of Computer Science. He is currently serving/served as a guest/associate editor in various journals. He has been involved in a number of conferences and workshops in various capacities.

Muhammad Imran is working as an Associate Professor with the College of Applied Computer Science, King Saud University. His research is financially supported by several grants. He has published a number of research papers in refereed international conferences and journals. His research interests include mobile and wireless networks, Internet of Things, cloud/edge computing, big data analytics, and information security. He has received a number of national and international awards. He served as an Editor-in-Chief for *EAI Transactions on Pervasive Health and Technology*. He also serves as an Associate Editor for many international journals, including IEEE ACCESS, *IEEE Communications Magazine*, and *Future Generation Computer Systems*. He has been involved in more than 75 conferences and workshops in various capacities, such as a chair, a co-chair, and a technical program committee member. These include IEEE ICC, Globecom, AINA, LCN, IWCMC, IFIP WWIC, and BWCCA.

Raihan Ur Rasool is a Fulbright Alumnus with the University of Chicago, USA. He is currently affiliated with Victoria University, Melbourne. His research work, comprising over 60 articles, is published in various international conferences and journals. His research interests include large-scale systems, security, and computer architecture.

Wanchun Dou (Senior Member, IEEE) received the Ph.D. degree in 2001. He is a Full Professor with the State Key Laboratory for Novel Software Technology, Nanjing University. To date, he has chaired four National Natural Science Foundation of China projects and published more than 100 articles in international journals and conferences. His research interests include big data, cloud computing, and service computing.