Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Integrating Edge Computing and Software Defined Networking in Internet of Things: A Systematic Review

Imran Edzereiq Kamarudin[1], Mohamed Ariff Ameedeen[1]*, Mohd Faizal Ab Razak[1], Azlee Zabidi[1],

[1]Faculty of Computing, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, MALAYSIA

*Corresponding Author: Mohamed Ariff Ameedeen

DOI: https://doi.org/10.52866/ijcsm.2023.04.04.011
Receive July 2023; Accepted October 2023; Available online November 2023

ABSTRACT: The Internet of Things (IoT) has transformed our interaction with the world by connecting devices, sensors, and systems to the Internet, enabling real-time monitoring, control, and automation in various applications such as smart cities, healthcare, transportation, homes, and grids. However, challenges related to latency, privacy, and bandwidth have arisen due to the massive influx of data generated by IoT devices and the limitations of traditional cloud-based architectures. Moreover, network management, interoperability, security, and scalability issues have emerged due to the rapid growth and heterogeneous nature of IoT devices. To overcome such problems, researchers proposed a new architecture called Software Defined Networking for Edge Computing in the Internet of Things (SDN-EC-IoT), which combines Edge Computing for the Internet of Things (EC-IoT) and Software Defined Internet of Things (SDIoT). Although researchers have studied EC-IoT and SDIoT as individual architectures, they have not yet addressed the combination of both, creating a significant gap in our understanding of SDN-EC-IoT. This paper aims to fill this gap by presenting a comprehensive review of how the SDN-EC-IoT paradigm can solve IoT challenges. To achieve this goal, this study conducted a literature review covering 74 articles published between 2019 and 2023. Finally, this paper identifies future research directions for SDN-EC-IoT, including the development of interoperability platforms, scalable architectures, low latency and Quality of Service (QoS) guarantees, efficient handling of big data, enhanced security and privacy, optimized energy consumption, resource-aware task offloading, and incorporation of machine learning.

Keywords: Edge Computing (EC), Software Defined Networking (SDN), Internet of Things (IoT),

1. INTRODUCTION

The Internet of Things (IoT) has garnered significant attention from both industry and academia in recent years, driven by the increasing demand for IoT devices. It has emerged as one of the fundamental underlying trends towards digital transformation. IoT facilitates the convergence of the digital and physical worlds. Benefiting from this convergence, a wide range of smart ecosystems and devices have been introduced in the market, including homes, vehicles, transportation, health care, and industrial products. It is estimated that by 2030, the economic value IoT could generate is between $5.5 trillion and $2.6 trillion globally, including products and services [1]. Also, according to Cisco [2], IoT devices will account for 50 percent, or an estimated 14.7 billion, of all globally networked devices by 2023.

With the rapid development of IoT, significant efforts have been made towards active development and deployment to address its limitations. IoT's main critical success factor depends on interoperability and open access between different platforms [3]. Each platform solution provides its IoT infrastructure, devices, APIs, and data formats. As a result, this creates a challenge for platforms with closed ecosystems to work with each other. Furthermore, IoT devices come from heterogeneous network environments [4]. This environment and the vertically fragmented network platform increase the complexity of supporting a large-scale IoT network. Thus, both interoperability and supporting heterogeneous networks play a vital role in the scalability of IoT. With an increase in connected IoT devices, the massive amount of data generated presents another challenge for today's networks to handle efficiently [5]. As a result, new ways to filter, classify, and select IoT data are needed before transmitting it to centralized cloud storage. It involves not only data management but also the security and privacy of data [6].

On top of that, it is important to note that IoT devices and applications are highly sensitive to network latency [7]. Each application requires a specific level of tolerance, with high-sensitivity applications such as industrial automation, smart grids, and remote surgery requiring latency of 30-40 milliseconds or less [8], whereas applications such as smart wearables and wastewater management can tolerate a higher latency rate, preferably around 40–60 milliseconds or less [9]. Exceeding the latency limit can significantly impact on the overall performance of the service, referred to as

Quality of Service (QoS). To overcome this limitation, the IoT network must be able to adapt automatically to the ever-changing applications and device requirements. In addition to facilitating the auto adaptation, emerging network technology such as Edge Computing (EC) and Software Defined Networking (SDN) are the potential key enablers. By integrating both technologies, the IoT network will have a full view of network resources and device requirements. Network resources can then be efficiently assigned based on the device's requirements during a specific time. Both EC and SDN serve distinct roles in the IoT environment, with EC providing processing capabilities closer to the edge devices and SDN enabling the network to be programmable and dynamically adjustable.

EC has the capability to process computer-intensive tasks from resource-limited IoT devices that cannot be performed locally [10]. This allows for the distribution of computation load across multiple edges, referred to as edge nodes or cloudlets. One of the primary advantages of EC is its proximity to IoT devices, which is closer to IoT devices than that of Cloud Computing (CC). It has received much attention over the past several years, to the point where researchers, industry leaders, and government organizations are concentrating on expanding the use of EC for IoT networks. An Edge Computing Internet of Things (EC-IoT) architecture has been proposed to leverage EC capability in IoT networks [11]. Edge-based services have been deployed across multiple smart applications such as transportation, homes, healthcare, cities, and buildings [12]. While utilizing the edge computational infrastructure, the performance of some IoT services suffers, especially from sending data to the edge and then waiting for the response. Therefore, there is a need to identify and classify delay-sensitive and highly compute-intensive applications in IoT environments. Ideally, latency-sensitive applications are sent to edge infrastructure for faster responses, while high compute-intensive applications are sent to cloud infrastructure for further processing [13].

On the other hand, SDN provides centralized control and an overview of the whole network [14]. Centralized control enables network resources to be optimized effectively and adjusted dynamically while ensuring interoperability across heterogeneous IoT networks. SDN provides a layered framework by separating the data and the control plane to facilitate network resource administration, traffic management, network evolution, and flexible network programmability. The flexibility of SDN to manage the network has been seen as one of the key enablers to solve IoT challenges, especially managing the complexity of an IoT network. A Software Defined Internet of Things (SDIoT) architecture has been proposed for effective network resource management [15]. By combining both EC-IoT and SDIoT, Software Defined Internet of Things for Edge Computing (SDN-EC-IoT) architecture creates a new communication perspective for IoT. This architecture benefits from combining the ability to process computer-intensive computation by EC with the effective management of network resources capability of SDN. SDN-EC-IoT provides promising results in solving some IoT challenges [16]–[18].

1.1 MOTIVATION OF SURVEY

IoT is anticipated to experience rapid growth, with the potential for full-scale adoption across various industries. For example, this can be seen when the concept of smart cities was introduced. Smart cities rely on information from multiple sensors and devices to make smart decisions on buildings, traffic, waste, energy, and water management [19]. Fast and responsive decisions are made based on the data gathered from and based on the IoT applications. Similarly, development can also be seen in smart healthcare. For example, patient management has become more accessible because doctors can monitor patients remotely based on the health-related sensors' data [20]. This capability enables healthcare professionals to make critical and accurate decisions by leveraging real-time patient data.

The IoT applications mentioned earlier require an optimal IoT network. The data generated is enormous over time, starting from IoT sensors and devices. This large amount of data requires an infrastructure to process and store if necessary [21]. EC is advantageous because it has more computation power and storage than IoT sensors and devices. Also, EC can store selected mission-critical data; the rest can be sent to CC for more extensive storage capacity. Besides, EC's position near the IoT sensors and devices compared to CC's provides optimal solutions for latency-sensitive applications. On the other hand, the demand for innovative approaches to network management is growing because of the exponential growth of the IoT infrastructure. To cater to this, a centralized management view of the overall IoT network infrastructure and resources is vital [22]. SDN provides an overall view of the whole network, which offers greater control, flexibility, and efficiency in centralized network management.

This paper provides in-depth literature on how IoT applications and services can benefit from integrating EC and SDN into IoT architecture. The disjointed development of IoT infrastructure and services creates non-standardized solutions, which creates issues such as interoperability [3] and scalability [23] among IoT devices. In terms of IoT services, various applications require different latency levels [24]. The overall IoT architecture is required to manage network resources dynamically to provide sufficient QoS. Managing huge numbers of data from large-scale IoT sensors efficiently has been challenging [25]. This large amount of data strains storage and computation power, and data leakage can also happen at the intermediate nodes during the data transfer, posing higher privacy issues [26]. Furthermore, IoT suffers more security threats than traditional networks due to its sensors, devices, and network heterogeneity. Due to the abovementioned challenge, this paper is designed to provide a detailed approach highlighting the significance of EC and SDN integrating with IoT architecture to address IoT challenges in various IoT applications.

1.2 RELATED SURVEY

Several surveys focusing on different aspects of the architecture for IoT have been conducted during the past few years, including EC-IoT [12], [27]–[29], SDIoT [30]–[33] and SDN-EC-IoT [34]. Most of these surveys address individual aspects of architecture. However, there is a lack of survey publications on the SDN-EC-IoT architecture from a comprehensive perspective, including requirements, standardization, and application. Moreover, most did not consider combining EC and SDN with IoT as an overall architecture. For example, in [34], authors evaluated the implementation of the SDN-EC-IoT architecture. However, they did not consider the implementation of the architecture in IoT application scenarios such as smart cities, homes, transportation, healthcare, and grids.

This paper focuses on how EC, SDN, and IoT can collaborate by leveraging the advantages of EC and SDN in IoT. Both technologies offer novel services besides complementing existing IoT applications. Table 1 outlines the most recent surveys on the IoT taxonomy from the perspective of various architectures and IoT applications. A detailed study of the literature available on the SDIoT, EC-IoT, and SDN-EC-IoT paradigms is performed, including a comprehensive discussion on key requirements for the SDN-EC-IoT paradigm based on existing works done on IoT applications. The key contributions of this paper are as follows:

- Present the foundation of SDN-EC-IoT by individually reviewing relevant literature on IoT, EC, and SDN. Furthermore, we explore the literature on combined architecture: EC-IoT, SDIoT, and SDN-EC-IoT.
- Discuss and analyze existing IoT and its challenges to determine how they can be addressed and the benefits of combining both EC and SDN in IoT architecture.
- Critically discuss, analyze, and evaluate current implementation efforts of SDN-EC-IoT in IoT applications such as smart cities, homes, transportation, healthcare, and grids.
- Present open research issues, challenges, limitations, and future research directions.

Table 1. – Comparison of previous surveys on EC and SDN in IoT

Ref	EC	SDN	IoT	EC-IoT	SDIoT	SDN-EC-IoT	IoT Application	Publication Year
Rafique et al.[12]	✓	✓	✓	✓	✓	×	✓	2020
Hamdan et al.[29]	✓	×	✓	✓	×	×	✓	2020
Li et al. [32]	×	✓	✓	×	✓	×	✓	2020
Jazaeri et al.[34]	✓	✓	✓	✓	✓	✓	×	2021
Ja'afreh et al.[35]	×	✓	✓	×	✓	×	×	2021
Laroui et al.[36]	✓	×	✓	✓	×	×	×	2021
Imran et al.[28]	✓	×	✓	✓	×	×	✓	2021
Alam et al.[37]	×	✓	✓	×	✓	×	×	2021
Our Survey	✓	✓	✓	✓	✓	✓	✓	2023

1.3 ORGANIZATION OF THE SURVEY

The rest of this paper is structured as follows. Section 2 explains the study's research methodology. Section 3 describes the underlying technologies. Section 4 presents the challenges faced by IoT applications. Section 5 describes EC and SDN involvement in mitigating challenges in IoT applications. Section 6 details the SDN-EC-IoT approach. Section 7 highlights the future research directions. Lastly, Section 8 concludes the study.

2. METHODOLOGY

The search strategy is based on a combination of keywords to identify EC and SDN in the IoT. The keywords are divided into three groups. Each search will apply one keyword from each group; for example, we used the following combination of keywords: (i) "Edge Computing"; (ii) "IOT" OR ``Internet of Things"; (iii) "Smart cities". The keywords used for the search are listed in Table 2.

To demonstrate the growth of the SDN-EC-IoT domain, this paper conducts an investigation of the domain by presenting a comprehensive evaluation of SDN-EC-IoT research practices published in the Web of Science (WOS), Elsevier's Scopus, IEEE, MDPI, and ACM Digital Library from 2019 and 2023.

Over 127 articles were identified and scrutinized using "Edge Computing," "Software Defined Networking," and "Internet of Things" as the primary keywords before being classified into 74 main related articles. The selection also combines the keywords "smart cities", "smart healthcare", "smart transportation", "smart homes" and "smart grid". Fig. 1 illustrates the paper selection process.

Table 2. – Keywords in respective group

Group 1	Group 2	Group 3
“Edge Computing"	“IoT” OR “Internet of Things”	“Smart cities”
“SDN OR “Software Defined Networking”		“Smart healthcare”
“Edge Computing" AND		“Smart transportation”
(“SDN” OR “Software Defined Networking”)		“Smart homes”
		“Smart grid”

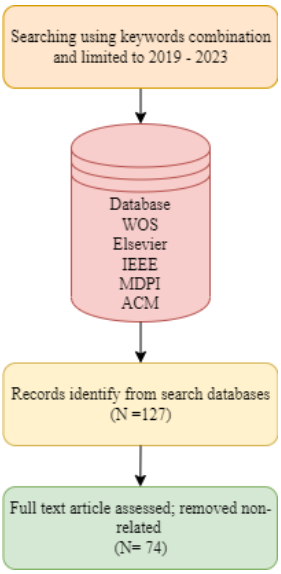

FIGURE 1. – Paper selection process. N represents the number of papers

3. UNDERLYING TECHNOLOGY

This section discusses the underlying technology and key concepts mentioned in this paper, which include the Internet of Things (IoT), Edge Computing (EC), Software Defined Networking (SDN), Edge Computing Internet of Things (EC-IoT), Software-Defined Internet of Things (SDIoT), and Software-Defined Networking for Edge Computing in the Internet of Things (SDN-EC-IoT).

3.1 INTERNET OF THINGS

During the early stage, several international bodies came up with their definitions and standardization to facilitate the development and advancement of IoT. The ITU Telecommunication Standardization Sector (ITU-T) has defined the IoT as a global infrastructure designed for the information society. This infrastructure enables advanced services through the interconnection of physical and virtual entities using interoperable information and communication technologies [38]. The IoT leverages capabilities like identification, data capture, processing, and communication to utilize physical entities fully. In doing so, it offers services to a wide range of applications while prioritizing the fulfillment of security and privacy requirements. On the other hand, The Internet Engineering Task Force (IETF) defines IoT as a network of physical objects or things embedded with electronics, software, sensors, actuators, and connectivity to enable objects to exchange data with the manufacturer, operator, and other connected devices [39]. In contrast, the Institute of Electrical and Electronics Engineers (IEEE) defines IoT as a network of items, each embedded with sensors and connected to the internet. All three well-known bodies have their working groups, participating in various areas such as generating IoT-related standards for architecture and framework, QoS and Quality of Experience (QoE) performance, protocol, test specifications, and many more. In addition, when adopted, these standards will guarantee interoperability and security support across the IoT architecture [40].

The core components within the architecture of IoT consist of the perception, network, and application layer. In the perception layer, sensors and actuators are deployed to collect data from the physical world. The collected data is then subjected to filtering, preprocessing, and real-time analysis, provided that the gateways possess the requisite

capabilities. Alternatively, the data is directed to the subsequent application layer. The application layer, such as CC infrastructure, typically has extensive storage and computational resources. IoT applications come into play within the application layer, frequently relying on cloud resources and associated services to store and process data. Diverse IoT applications, spanning domains like smart cities, transportation, healthcare, homes, and grids, leverage this data to deliver valuable services to end-users. Fig. 2 shows the core components of IoT architecture: the perception, network, and application layers [41]. It provides a foundation for IoT technology to evolve.

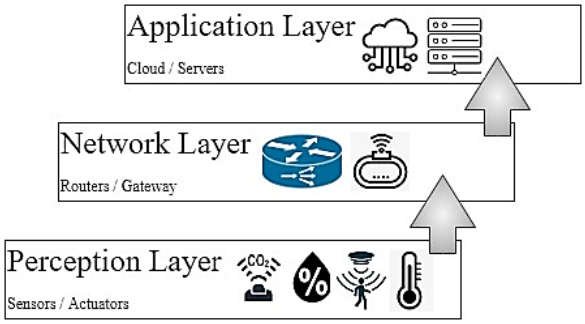

FIGURE 2. – IoT architecture

In recent years, new inventions and technological advancements in IoT have been made through research. This evolution has led to new IoT branches [42], named based on technologies they used during the development and innovation process. Fig. 3 depicts several types of IoT branches.

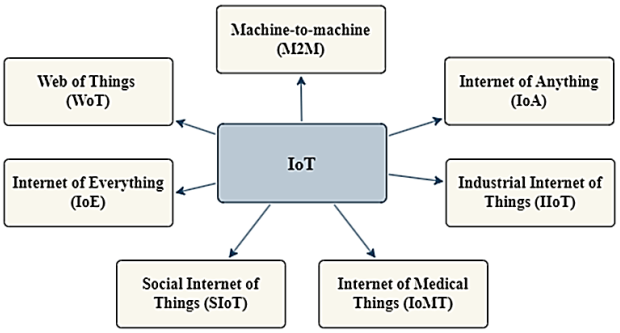

FIGURE 3. – Different branches of IoT

3.2 EDGE COMPUTING

The EC is a decentralized computing model designed to bring computation and data storage closer to the data source. It enables data processing at greater speeds and storage by extending CC capabilities nearer the network's edge. This feature benefits IoT, especially in real-time applications, by reducing the latency and response time [29]. EC and CC work hand in hand to complement each other. Depending on the type of IoT services, EC manages and ensures the continuity of these real-time services while maintaining enough data for the services to run. If a less critical service needs higher computation power and storage size, it will be offloaded to CC.

EC consists of three common layers: cloud, edge, and device [36]. The top layer is the cloud layer, which can either be a public or private cloud infrastructure. This layer hosts and runs computation-intensive applications that the edge layer cannot handle. Furthermore, it also runs applications that orchestrate and manage the different edge nodes. The workload from the edge layer will be offloaded and interact with the cloud. The cloud can also be the primary source of storage when needed.

The second layer is the edge layer. This layer has four common implementation forms: edge node, cloudlet, fog computing, and mobile edge computing (MEC). However, they differ in location, functionality, and architecture [29]. Edge nodes refer to any edge servers or gateways where the EC function is performed. This node can be divided into virtual partitions according to the services provided [43], for example, edge nodes for serving smart city facilities in the first partition, edge nodes for serving medical facilities in the next partition, and so on. This individual virtual partition will communicate with its respective device layer.

A cloudlet functions similarly to a conventional cloud by offering computing services, but it operates on a smaller scale with limited resources compared to a traditional cloud. Users are directly linked to other devices within a specific

cloudlet, often localized to a particular region. This regional specificity is commonly seen in mobile devices, significantly reducing response latency for user requests. Furthermore, when a mobile device moves from one region to another, it connects from one cloudlet to another. Therefore, cloudlets can be used in environments requiring superior situation awareness, decision-making, and reliable connectivity [44].

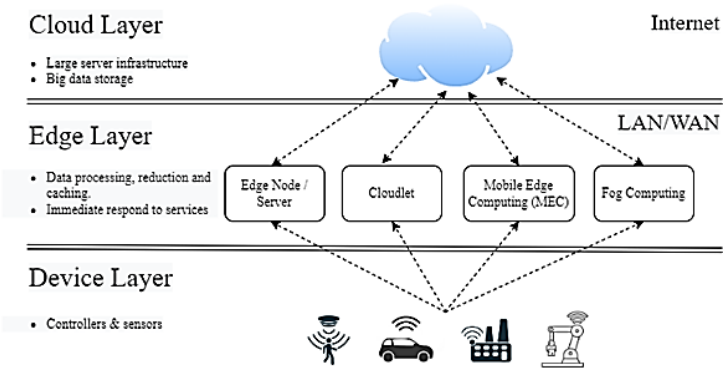

FIGURE 4. – EC architecture

Fog computing represents a decentralized computing infrastructure characterized by a network of computing nodes. In this architecture, the services offered to end-users are strategically positioned between the end-users themselves and the cloud [29]. It extends the CC function to the edge of an enterprise's network. The fog layer is made up of fog nodes, which are network-connected computing and storage devices. Fog nodes being close to the IoT devices helps avoid cross-network traffic and addresses the latency issue in the traditional cloud architecture. The OpenFog Alliance was created in 2016 by a collaboration between Intel, Microsoft, Dell, ARM, Cisco, and Princeton University to support the development of fog computing-related technical standards and promote technological transformation in the industry [45].

MEC is EC applied to mobile devices. The primary purpose is to reduce mobile devices' latency and power consumption by allowing them to offload their computation-intensive tasks to nearby servers [46]. This is possible by integrating computing, storage, and networking resources with the 3G, LTE, and 5G base stations.

The third layer is the device layer, which consists of edge devices such as sensors, cameras, and controllers. Edge devices can gather and transmit data to the edge layer. However, this capability is limited due to the device's small size, which limits the computation and storage capacity. Fig. 4 shows the overall architecture of EC architecture with the functions of each layer.

3.3 SOFTWARE DEFINED NETWORKING

The SDN concept is based on the approach of network programmability. Unlike a traditional network, the main idea is to separate network hardware and software. This separation enables the generalization of network hardware and decoupling the network control software from the implementation network devices [47]. SDN separates the control plane from the network devices, enabling the data control from a centralized and external software entity called an SDN controller. This controller can dynamically initialize, control, modify, and oversee network behaviour through open interfaces.

In 2015, the Internet Research Task Force (IRTF) issued its initial guidelines through its Software Defined Networking Research Group (SDNRG). These guidelines aimed to clarify the nature of SDN, define the layer structure within an SDN architecture, and elucidate the interactions between these layers [48]. To further the advancement and adoption of SDN, The Open Networking Foundation (ONF) was established as a consortium dedicated to developing, standardizing, and commercializing SDN for the transport and IP network layers. As a result, the SDN architecture was defined, consisting of forwarding plane, management plane, control plane, and application plane.

The forwarding plane, or data plan, handles and forwards packets. It performs essential functions like switching, routing, packet transformation, and filtering. Additionally, it processes packets within the data stream based on instructions relayed from the control plane [49]. The data packets can be forwarded, discarded, or modified depending on the action required. The data plane presents a forwarding table that forwards the incoming packets to a network device. Typically, the forwarding plane serves as the end point for control plane services and applications. Network devices within the forwarding plane, including switches and routers, can be implemented using hardware or software in physical or virtual forms. Virtual network devices are software-based switches, for example, Open vSwitch[50] and Stratum[51]. On the other hand, physical switches are hardware-based switches implemented either on open network

hardware such as Stordis[52] or on a commercial switch from networking hardware vendors such as Cisco[53] and HP[54].

The management plane is responsible for overseeing, configuring, and upkeeping network devices, including tasks like assessing the status of these devices [55]. Furthermore, it offers intelligent provisioning and orchestration systems for comprehensive network management, guaranteeing optimal performance through communication with network devices.

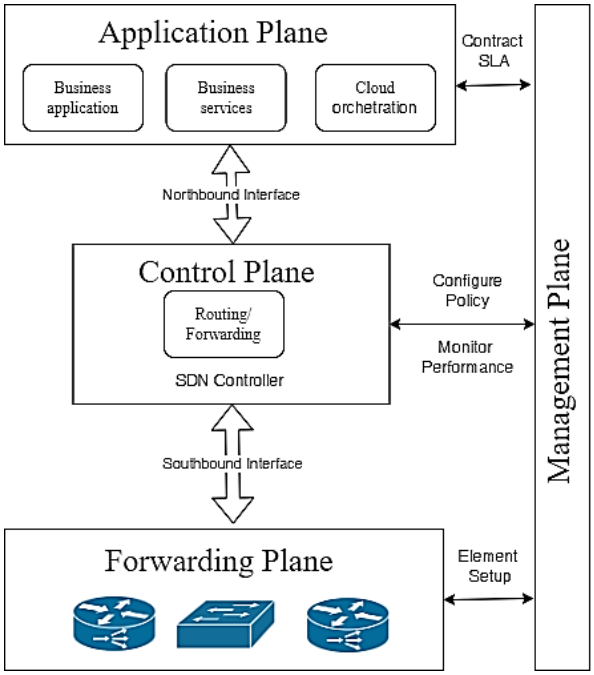

FIGURE 5. – SDN architecture

The intelligence behind SDN is often called the control plane, acting as its central decision-making component. It dictates the path packets should follow through one or more network devices and conveys these decisions for execution by the network devices[56]. While it does pay some attention to the operational aspects of the device, its primary focus lies on the forwarding plane. The control plane might take an interest in operational-plane data, like a specific port's current status or capabilities. Its primary responsibility involves refining the forwarding tables within the plane based on network topology or external service requirements. Examples of the SDN controller architectures are NOX [57], Floodlight [58], Ryu [59], and Beacon [60]. Furthermore, SDN controllers have three interfaces to facilitate communication: northbound towards the application plane, southbound towards the forwarding plane, and eastbound towards other SDN controllers [61].

The Northbound Interface (NBI) provides abstract network views and enables the direct expression of network behaviour and requirements. Applications utilize NBIs to obtain an abstract view of the network to ease automation, study specific network behaviours, and assess network needs. Based on the service requirement, NBI can take many forms of service interface [62], such as RESTful APIs, NETCONF, Common Object Request Broker Architecture (CORBA) interfaces, and Remote Procedure Call (RPC). On the other hand, the Southbound Interface (SBI) is used for forwarding devices to exchange control policies and network state information between the control and forwarding plane. Furthermore, SBI APIs allow the end-user to gain better control over the network and promote the SDN controller's efficiency level to evolve based on real-time demands and needs. SBI offers a few abstraction model APIs, such as Openflow [30], YANG [58], NETCONF [63], and OVSDB [58]. Lastly, a multi-SDN controller solution uses the eastbound and westbound interfaces to exchange information among controllers. This is important between controllers to provide a global network view to the application.

The application plane is where applications and services that define network behaviour reside. This may include business applications that provide management and optimization of business services. NBIs are used by each application to communicate its network requirements and behaviour to the SDN controller [55]. Examples of these applications include those focused on network topology discovery, network provisioning, and path reservation. Fig. 5 shows the overall architecture of SDN, consisting of four planes: the forwarding plane, management plane, control plane, and application plane [64].

3.4 EC-IOT

The EC-IoT architecture comprises three layers: the IoT access, edge, and cloud computing layers [65]. The IoT access layer serves as the communication link between IoT devices. Examples of IoT objects include actuators, sensors, gateways, and controllers. These objects monitor equipment, appliances, services, and activities for smart cities, smart healthcare, smart transportation, smart homes, and smart grids. Due to their limited computation and memory, IoT devices can only do preliminary processing on the data before offloading the remaining of the computation to the higher levels. Fig 6. shows the detailed architecture of each layer.

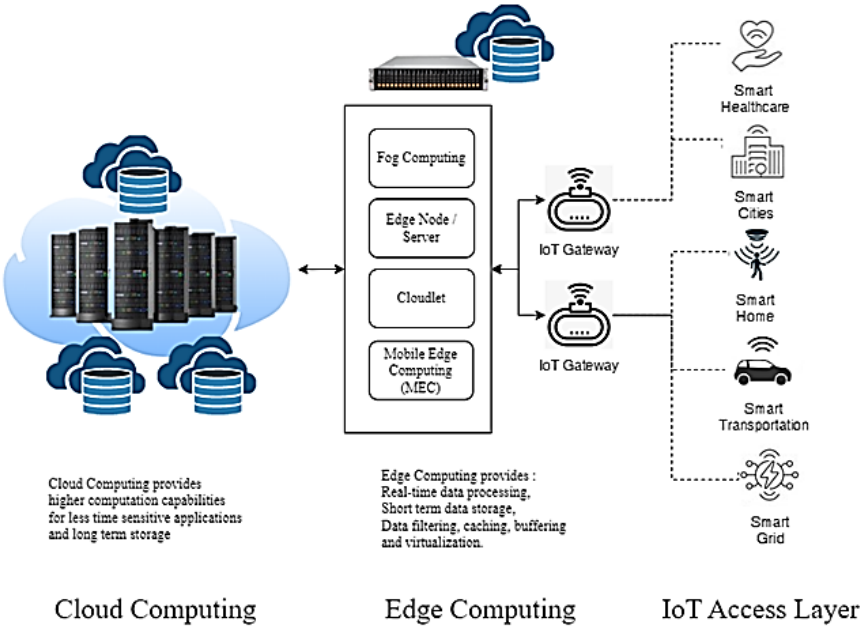

FIGURE 6. – EC-IoT architecture

The EC layer is the central layer of the EC-IoT architecture. The EC layer comprises several edge gateways, such as fog computing, servers, cloudlets, and MEC, located at the edge of the IoT network. In this layer, data offloaded by devices in IoT access are filtered and preprocessed before being sent to the CC layer [66]. Real-time data is transmitted from several edge nodes in different parts of the IoT access layer to the edge server. Next, this data can be processed, filtered, cached, buffered, and visualized for real-time IoT services. If needed, the edge server performs the required computation and delivers the results to the cloud. Edge gateways are high-capacity computers that serve as the intersection point between the IoT layer devices and the cloud. They are responsible for performing the function of the EC. The EC devices ensure that diverse items in the IoT access layer can be accessed safely, and the CC layer can collaborate effectively using APIs. As edge servers are deployed close to the users, they provide storage and computational resources. This reduces the latency for end devices and provides faster real-time response for end-user applications [29]. Deploying EC in IoT architecture brings several benefits and solutions to IoT problems.

- *Computation power*: Unlike IoT devices with limited computation and memory, EC can cater to IoT services requiring higher computation and memory capabilities. Smart cites, for example, require massive collaboration between services, which requires high computation power to process data [67].
- *Low latency*: Due to the proximity of EC and IoT devices, EC can process user requests faster by potentially reducing communication latency between IoT devices and offering intelligent decisions. This will produce faster response times and increased operational efficiency. Services such as dynamic video stream processing benefit from real-time information processing and decision-making from EC [68].
- *Customization of each IoT service*: By analyzing the data gathered from each IoT service, EC can efficiently allocate resources based on each service requirement. This will ensure that each IoT service is assigned EC resources as efficiently as possible [69].
- *Decentralization*: The existence of EC lessens the burden of network core infrastructure. Rather than centralizing all computation in CC, only selected data that require higher computation power than EC and long-term storage must be handled by CC. By doing so, fewer data needs to travel across the core network [70].

- *Geographical load distribution*: By deploying distributed EC, computation load can be shared among EC when needed. EC ensures the computation load is offloaded and distributed efficiently while preserving privacy and security [71]. This is important for large-scale IoT services such as smart cities, which involve massive collaboration across huge areas.

3.5 SDIOT

SDIoT extends the SDN approach to collect and aggregate data from the IoT perception layer. In comparison to the traditional IoT architecture, the SDIoT architecture offers facilities for managing security and network resources with a centralized control plane. SDIoT is built on the traditional IoT protocol stack, with improvisations of the control plane in the network layer. Fig. 7 depicts the SDIoT architecture, which consists of four main components: the forwarding plane, control plane, application plane, and management plane [31].

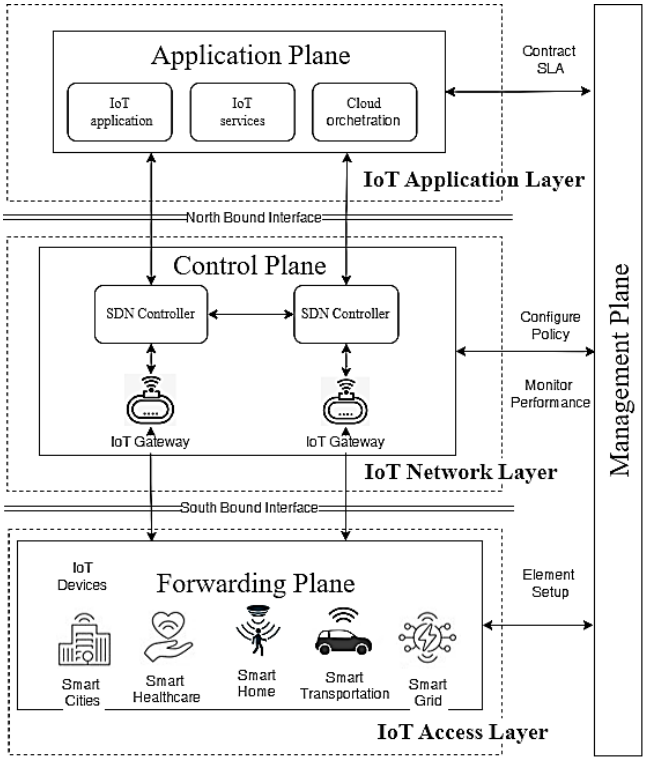

FIGURE 7. – SDIoT architecture

The architecture addresses the clear separation of concerns between services provided in the control plane and those provided in the forwarding plane. The control plane specifies network traffic management, while the data plane specifies the mechanisms for forwarding traffic to the desired destination. It determines how applications on top of the management layer interact with the control plane and how they collaborate [72]. It also lets the network administrator specify how the SDN controller and human users should govern the control process.

The lowest layer in the SDIoT is the forwarding plane. Like the IoT access layer, it consists of sensors and actuators that collect huge amounts of real-time data. The data is presented in various formats for IoT applications, such as smart cities, healthcare, homes, transportation, and grids [73]. Each sensor and actuator have numerous integration points at the middle layer, such as gateways, routers, and a centralized SDN controller for this device network.

The middle layer of the SDIoT architecture's control plane consists of IoT gateways and SDN controllers. Sensors and actuators from the forwarding plane are typically connected to a centralized IoT gateway. This IoT gateway is primarily responsible for forwarding data in the network. Furthermore, it can cache local data or process information based on instructions from an SDN controller. On the other hand, SDN controllers are responsible for handling data forwarding and data processing management [74]. These controllers can efficiently manage equipment, such as IoT gateway configuration, routers, virtual network components, and policy definition. In this case, the control plane is centralized and programmable, allowing the network to be tuned in any way it sees fit.

The top layer in the SDIoT is the application plane. Applications and services consumed by an end user are deployed in this layer. These applications and services interact with cloud data, communicate with devices via SDN

controllers, and process information based on IoT applications, such as smart cities, healthcare, homes, transportation, and grids [55]. Deploying SDN in IoT architecture brings several benefits and solutions to IoT problems.

- *Secure architecture*: Due to the heterogeneous nature of IoT devices, each perceives threats differently. As a result, creating a unified security policy to cover the entire IoT network becomes nearly impossible. A centralized SDN controller helps by providing visibility of the security threats across the whole network from a single point of view. Doing so can enforce a standard, centralized, and effective security policy for the entire network [75].
- *Scalability*: Scalability is a problem made worse by the significant number of IoT devices and the data generated by these devices over time. Not only must the network be able to handle this amount of traffic and data, but it also needs to cater to new devices and the additional volume of data that is constantly added to the network. By implementing an open-layer concept, SDN provides greater scalability in IoT networks without causing significant changes in the central hardware's software, tools, or protocols [76].
- *Load Balancing*: SDN can distribute the network traffic among multiple devices, such as virtual machines and servers within its network, to prevent the overloading of a particular host. Centralized control coupled with a load estimation technique helps to balance and distribute the traffic across the network by analyzing and determining the predicted load and designing the network's flow routes. This load-balancing solution helps to conserve the network bandwidth, decrease energy consumption, and reduce the redundant data packets in the IoT network [14].
- *Constrained Environment*: Deploying IoT applications in a real-time environment requires excessive resources. Each application has its own resource requirement sets because of the diverse architecture employed. SDN enhances network control and dynamically adapts by providing the required network resources to the IoT devices in real-time. This helps improve the overall QoS of the IoT network [77].
- *Fault tolerance*: In IoT networks, fault tolerance is crucial, especially for centralized and integration points like IoT gateway. If an integrated point goes down, the entire traffic component must be migrated to another similar component. This process is tedious and may also require downtime and an outage for the section until it is fully migrated. Due to the centralized implementation of routing rules and protocols at the controller level, SDN ensures smooth migration in such cases [78].

3.6 SDN-EC-IOT

SDN-EC-IoT comprises IoT access, SDN/EC, and CC layers [34]. Fig. 8 shows the detailed architecture and functions of each layer. The IoT access layer consists of sensors and actuators that provide real-time data readings for multiple parameters, depending on the applications. Examples of parameters such as temperature, humidity, and barometric pressure are captured for smart city applications. In addition, various sensors are typically connected to an IoT gateway to simplify the connection to the upper layer.

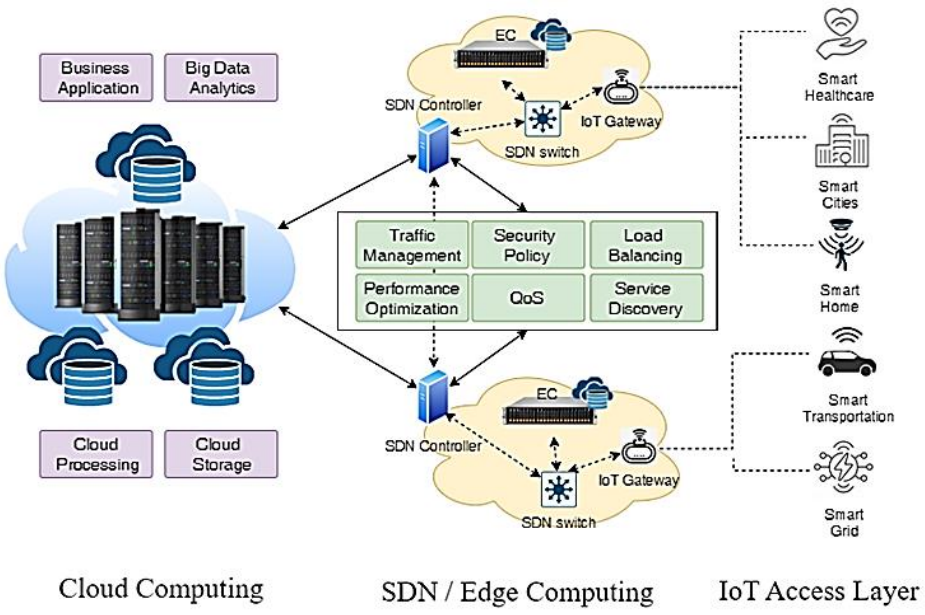

FIGURE 8. – SDN-EC-IoT architecture

The SDN/EC layer consists of both SDN and EC, each with distinct responsibilities. SDN's primary role is the management of IoT network resources. It achieves this by efficiently virtualizing IoT networks, enabling tasks such as automatic traffic rerouting, traffic management, load balancing, device reconfiguration, service discovery, and bandwidth allocation. These actions enhance performance and reduce network complexity [79]. Meanwhile, EC takes charge of data processing at the network edge of the IoT.

In a large-scale IoT network, distributed SDN controllers are deployed, where each SDN controller will manage its network resources [80]. In addition, each distributed SDN controller communicates with the other to share network status and resource updates. In terms of security, SDN provides enhanced network transparency by automating the detection of security threats, applying security policies, and implementing access control measures [81]. Additionally, SDN facilitates centralized management for various components such as sensors, terminals, communication modules, IoT gateways, and other devices. It also enables automatic deployment, security authentication, real-time status monitoring, and remote upgrades.

On the other hand, EC serves gateways and data capture facilities capable of operating on raw data gathered from the IoT access layer. It offers an open platform that handles tasks such as management, analysis, control, and data processing at the network edge of the IoT. [82]. The main advantage of EC is its capability to support real-time data processing, such as real-time data analysis, data visualization, simple analytics, data compression and buffering, and real-time data filtering. This benefits time sensitive IoT applications such as video processing and analytics for smart homes and cities, self-driving cars in smart transportation, and many more. The rest of the IoT applications will be offloaded to CC.

The massive storage and computation capabilities at the CC layer can be used for long-term data storage and applications, which require higher computation power [83]. Such applications are less time-sensitive compared to those running in EC. The cloud infrastructure can come from various providers. Deploying SDN-EC-IoT architecture brings several benefits and solutions to IoT problems:

- *Centralized management of resources*: A centralized SDN controller can oversee the IoT network resources in this architecture. It allows greater flexibility to control the network, change configuration settings, provision new resources, and increase network capacity, all from a centralized network view [31].
- *Improve performance of real-time applications*: IoT services that require low latency and fast computation response are served by edge nodes in the EC layer. At the same time, CC will process less time-sensitive applications with high computation needs. This can reduce delay, response time and increase the QoS of the applications [84].
- *Reduce network traffic*: The EC can filter unnecessary data and gather critical data that must be transferred to the CC layer for further storage and processing. As a result, only selected data will be offloaded to CC, significantly reducing network traffic from EC to CC [85].
- *Simplify load balancing*: EC can distribute computation load among EC nodes or CC, depending on the nature of the IoT services. Distribution among EC nodes will speed up the computation process while not sacrificing the latency for real-time applications [86].

4. CHALLENGERS IN IOT: SOLVING WITH EC AND SDN

This section discusses the challenges faced in deploying IoT, including interoperability, scalability, low latency, big data handling, security and privacy, energy consumption, QoS, and task offloading. Several initiatives based on the EC and SDN paradigms have been proposed to solve the challenges.

4.1 INTEROPERABILITY

Interoperability in IoT is the feasibility of exchanging information among IoT devices and systems. This exchange of information does not rely on the deployed software and hardware. The interoperability issue arises due to the heterogeneous nature of different technologies and solutions used for IoT development [3]. The three interoperability perspectives are device, network, and platform interoperability.

- *Device interoperability*: IoT devices are classified into three classes based on capability and communication ability. The first-class devices are low-end IoT devices. Low-end IoT devices are resource-constrained regarding energy, processing power, and communication capabilities. Therefore, it is used mainly for basic sensing and actuating applications, for example, temperature and gas sensors. The second-class devices are known as middle-end devices. Middle-end IoT devices provide features with greater processing capabilities than low-end IoT devices due to the higher range of memory and processing units. These enable the devices, for example, Tessel and Arduino Yun [87], to run traditional operating systems (OS) with some low-level computer vision algorithm [88]. The third class is high-end IoT devices capable of running a standard OS with tentative computations, such as executing a Machine Learning (ML) algorithm. This is possible due to the higher resources available, such as the powerful processing unit and bigger RAM allocation. Due to these advantages, high-end IoT devices such as Raspberry Pi and Odroid-XU4 [89] are often used as gateways connecting

multiple middle- and low-end devices. However, without a one-size-fits-all communication standard, the devices that want to exchange information may use different communication technologies. Therefore, device interoperability concerns integrating new devices into the existing IoT, with devices able to exchange information between heterogeneous devices and heterogeneous communication protocols.

- *Network interoperability*: IoT devices in heterogeneous environments rely heavily on various short-range wireless communication and networking technologies, making them susceptible to intermittent and reliability issues. Interoperability at the network level refers to mechanisms that enable seamless message exchange between systems via a different medium, such as wired and wireless, for end-to-end communication [90]. Each system should be able to exchange messages with other systems via various networks for systems to be interoperable. Because the network environment in the IoT is dynamic and heterogeneous, the network interoperability level should handle issues such as addressing, routing, resource optimization, security, QoS, and mobility support.
- *Platform interoperability:* IoT consists of multi-vertical-oriented platforms, each with its own OS, architecture, programming language, data structure, and data access method [40]. Each platform adopts a specific set of standards to enable connectivity between components and platforms. The range of differences includes variances in OS such as Contiki NG, RIOT, and Zephyr; communication technologies such as WiFi, Bluetooth, NFC, and ZigBee; application layer protocols such as REST, MQTT, and CoAP [16]; and data formats such as XML, JSON, RDF, and CSV [91]. This dissimilarity and inconsistency pose significant challenges for developers seeking to create cross-platform and cross-domain IoT applications.

4.2 SCALABILITY

Scalability in IoT has become a crucial concern due to the rapid growth in the number of connected things. Typically, all related software, hardware, and networks in the IoT ecosystems need to manage the ever-increasing load of work or the possibility of expanding them to deal with that load successfully. The main goal of a scalable solution is to maintain or improve the QoS to guarantee a specific degree of performance under heavy workloads [92]. There are three types of scalability: vertical, horizontal, and functional.

- *Vertical scalability*: Increase the capacity of existing hardware by adding more computing resources, typically on a single node in a system. The benefits of this are that implementation is simplified, software expenses are reduced, and application compatibility is maintained [23].
- *Horizontal scalability*: Increase the capacity by connecting multiple hardware or software components to function as a unified entity. The primary objective is to distribute workloads across numerous components, reducing individual loads, minimizing response times, and improving concurrency. This approach offers several advantages, including workload distribution across multiple nodes and the prevention of a single point of failure in case one of the nodes experiences hardware issues [93].
- *Functional scalability*: Unlike vertical and horizontal scalability, functional scalability allows IoT ecosystems to expand to accommodate any number of services without affecting global system properties like performance, maintenance, evolution, and monitoring of underlying hardware and software. Achieving functional scalability entails establishing metrics to gauge the level of satisfaction in accommodating new services. This involves factors such as explicit control flow, distributed workflows, location transparency, decentralized data flows, separation of control data, and variability in computation and workflows. [23].

4.3 LOW LATENCY

Latency is the time it takes between when data is sent from a connected device and when it returns to the same device. To avoid interrupted services, having a high reliability and low latency (HRLL) condition [94]. The HRLL requirement varies for different IoT services and applications. Challenges faced in HRLL include optimising available, limited network resources without sacrificing the IoT application. Therefore, applications must be prioritised based on their features and the requirement for assigning network resources. ITU has defined three categories to facilitate the IoT HRLL application requirement [24], as shown in Table 3.

4.4 HANDLING OF BIG DATA

Data collection and sensing involve using sensors to track the performance of devices connected to the IoT environment. The sensors track the status of the IoT network by collecting and transmitting real-time data that can be stored and retrieved at any moment. High-quality data is crucial to ensure the accuracy of the later stages of data processing and analytic operations [95]. Data collection and sensing challenges include data duplication and redundancy, which will create unnecessary storage loads.

Data processing within an IoT environment follows a three-stage process: input, processing, and output. In the input stage, data collected from sensors is initially converted into a machine-readable format to facilitate computer processing [5]. Subsequently, during the processing stage, this machine-readable data undergoes various manipulation

techniques such as classification, sorting, and calculation. The specific techniques employed depend on the requirements of the IoT applications. Finally, in the output stage, the processed data is converted back into a human-readable format and presented as valuable information to end-users. Additionally, processed data can be stored for future reference. Thus, the challenges associated with data processing include the efficient handling of substantial data volumes using dependable data manipulation methods [96].

Data storage involves storing data collected from IoT sensors. Depending on the data usage for IoT applications, data for short-term applications is typically stored at the gateway or edge. In contrast, data is stored in a cloud environment for long-term applications. This ensures that time-critical applications can obtain the necessary data faster than less time-critical applications [97]. Challenges faced in data storage include security and data integrity when transferring and storing large amounts of data across the cloud.

Table 3. – HRLL application category

Category	Features	Applications
Enhanced Mobile Broadband (eMBB)	• provide extreme throughput. • enhanced spectral efficiency. • extended coverage.	• cloud office/gaming. • virtual/ augmented reality (VR/AR). • three-dimension/ultra-high-definition (3D/UHD) video.
Ultra-reliable Low-Latency Communications (URLLC).	• mission-critical area. • requires HRLL, high availability and location precision. • high mobility	• industrial automation. • autonomous driving. • mission-critical applications. • remote medical assistance.
Massive Machine-Type Communications (mMTC)	• connectivity to many devices with low reliability • long-range communication with energy efficiency • asynchronous access	• low power devices in a massive quantity

4.5 SECURITY AND PRIVACY

The IoT network suffers from higher security threats due to the nature of the heterogeneous IoT network [98]. Therefore, deploying security solutions is challenging when compared to the traditional network. The main concern is securing the network infrastructure where multiple users, devices, and vendors participate in a single platform [26]. This can be addressed by looking at the security hierarchy structure, where different security technologies can be adopted in different IoT layers. The perception layer is the most vulnerable to security attacks due to the lack of security mechanisms deployed on resource-limited sensors and actuators [99]. Therefore, attack detection and intrusion response techniques are commonly used as defence mechanisms. At the network layer, the aim is to secure the network environment through wireless network security, secure routing, and a firewall. This is important to ensure safe network connections. Lastly, the application layer guarantees the security of the application system through user authentication and access control. The critical issues of IoT security are authentication, confidentiality, authorization, integrity, availability, and privacy.

- *Authentication:* Authentication is the process of identifying a device, whereas authorization grants permissions. These processes are used by IoT devices to provide role-based access control and ensure that devices only have access and permission to do what they need to accomplish [100]. For example, only approved devices can communicate with other devices, applications, cloud accounts, and gateways. Three main IoT authentication security protocols are commonly deployed: distributed one-way authentication, distributed two-way authentication, and centralized three-way authentication [101]. Depending on the IoT security framework, the authentication process varies. To achieve further improvements, it's beneficial to incorporate various security measures. These enhancements encompass the integration of cryptography standards, key exchange mechanisms, certificate-based signatures, as well as symmetric and public/private key encryptions. These security mechanisms are typically implemented through Transport Layer Security (TLS).
- *Confidentiality:* IoT devices generate a large amount of private data that must be processed, sent securely, and stored. Data confidentiality ensures that these data are free from tampering and access by an authorised user [102]. For IoT confidentiality, the challenge is dealing with high scalability requirements, the heterogeneity of the nature of IoT, and the scarcity of resources in the embedded devices, such as energy and computational limitations.
- *Authorization*: In IoT networks, access control is typically managed through policy-based authorization. This model stores data in the cloud for easy access, management, and auditing [103]. It also enables quick revocation

of access to cloud-stored data and connected IoT devices. Gateways play a key role by enforcing a set of access control rules. These rules determine which IoT devices or smart applications are allowed access. Only those explicitly specified in the rules can access the associated devices. If needed, access credentials and permissions can be revoked instantly through system-level authorization.

- *Integrity:* Integrity is a crucial aspect in IoT due to the exchange of data among diverse devices. It is essential to guarantee the accuracy of data, verify its source, and prevent any tampering during transmission, whether intentional or accidental. To enforce data integrity, end-to-end security measures can be implemented in IoT communication [104]. However, data flow is regulated using firewalls and protocols, but this does not guarantee endpoint security due to the low computational capability of IoT nodes. In addition, the data leakage concerns at the intermediate nodes during the data transfer pose higher privacy issues.
- *Privacy*: As devices in the IoT environment transmit data autonomously, privacy is critically required [105]. The transmitted data is typically very rich and often includes meta-data, such as location, time, and context, thus making it possible to easily interpret it if it falls into the hands of unauthorised users.

4.6 ENERGY CONSUMPTION

Energetically autonomous devices are the fundamentals of the IoT. IoT devices, especially at the perception layers, consist of sensors and actuators powered by batteries [106]. They require low-power wireless technologies such as Zigbee, NFC, Bluetooth, and RFID for communications. They are expected to be able to operate without user intervention for months or years. Before the battery runs out of power, they will alert users in advance that the battery needs to be replaced. However, this device's challenges are managing and optimising energy consumption without compromising its services.

4.7 QOS

QoS in IoT measures the IoT network's overall performance, guaranteeing its capability to run high-priority applications [107]. It controls and monitors how data moves over the network to reduce jitter and packet loss. QoS manages network resources in setting multiple data transmission needs by prioritising the communication of specific data types. IoT applications have various QoS requirements, categorised as best effort, differentiated, and guaranteed services [108]. The challenge of classifying each IoT service is ensuring that each application falls under the correct QoS requirement. To maintain an acceptable QoS for safety critical IoT applications, it is imperative to implement QoS measures at every layer of the IoT architecture. Any delay occurring within the layers, spanning from the sensors to the end-user, can potentially pose issues for various critical applications across domains. These applications include areas like autonomous vehicles and healthcare, where timely and reliable data is paramount to safety and effectiveness.

4.8 TASK OFFLOADING

Task offloading involves transferring both the current workload and the substantial volume of data generated by IoT devices to either the EC or CC layers for further processing. CC serves as the backend layer for data storage and analysis, offering extensive storage and processing capabilities, especially suitable for large data sets beyond the reach of IoT devices. However, offloading tasks to the cloud layer can be inefficient, leading to network bandwidth overhead [109]. Moreover, it introduces delays in data analysis due to relatively high network latency [110]. On the other hand, offloading IoT tasks to EC rather than the cloud layer mitigates network congestion and reduces data analysis response times, capitalizing on the advantages of lower network latency. The challenges faced by task offloading are IoT devices need continuous offloading services, where the sequence of committing the task to the edge and re-establishing the connection must be handled seamlessly to address the service level requirements.

4.9 EC AND SDN ROLE

Several studies have shown promising results on the effectiveness of integrating EC and SDN to address the IoT challenges mentioned earlier. Deploying EC in IoT primarily focuses on processing, analysing, and managing IoT data. There are several ways in which EC addresses IoT challenges:

- *Low latency*: Enable real-time analysis and decision-making by devices and systems at the network's edge, rather than relying on a CC to process the data, to reduce time and bandwidth [111]–[113].
- *Handling of big data*: Analyse, filter, and decide on relevant data from the IoT network for further storage in CC [114], [115].
- *Energy consumption*: By processing data closer to the devices, EC helps reduce overall energy consumption. Moreover, tasks are shared among available EC whenever to mitigate devices with energy constraints [9], [116].
- *QoS*: Sensitive IoT applications are computed at EC rather than CC [8], [10].

On the other hand, deploying SDN in IoT allows for more flexible and efficient management of network resources. There are several ways in which SDN addresses IoT challenges:

- *Interoperability*: By providing a centralized control plane that can be used to orchestrate and manage the flow of data between different IoT devices and networks [117]–[119].
- *Scalability*: SDN allows network administrators to centralize control over the network, making it easier to manage large numbers of IoT devices and the traffic they generate [14], [92], [120].
- *Security and privacy*: By centralising control of the network and enabling the use of policies and rules to govern the flow of data, it is possible to better protect against security threats and privacy concerns [121], [122].
- *Tasks offloading*: Centralized control helps SDN identify available resources and distribute workloads more evenly across the network to optimise resource use and improve the system's overall performance [123].

5. EC AND SDN FOR IOT APPLICATION

EC and SDN provide an efficient platform for smart IoT applications, including smart cities, healthcare, homes, transportation, and grids. In this section, the review covers recent works on IoT applications based on EC-IoT, SDIoT, and SDN-EC-IoT architectures to address the IoT challenges. A brief comparison between the studied articles is presented in Table 4.

5.1 SMART CITIES

A smart city [16] is a modern urban area that uses different devices, especially IoT, to collect specific data. The data is used to efficiently manage assets, resources, and services to improve city operations. Taking advantage of the growth of IoT, it is transforming cities in various ways, from improving public safety to creating smarter cities. By providing real-time data on urban infrastructure, smart city planners can use this information to manage energy consumption, reduce traffic congestion, ensure better sanitation services, manage lighting, manage smart parking, and much more.

To manage the massive amount of data, smart cities require fast, efficient, and intelligent in data analysis [22]. This process requires compute-intensive devices near the end user to make fast decisions, especially for latency-aware applications such as traffic control and public safety. Furthermore, integrating ML algorithms [124] into the smart city platform can achieve fast decisions without human interaction. Another significant aspect of smart cities is managing the overall network resources. With the massive scale of an urban area, a centralized control system that can oversee the whole network is desirable [14]. For example, traffic lights and other traffic control systems can be connected to a central network, which manages and optimizes traffic flow and reduces congestion.

EC and SDN provide an ideal platform for smart cities [19]. EC provides data processing at the network edge for faster and more reliable communications than CC. On the other hand, SDN provides the ability to have an overall network view for more efficient network resource management. Both are essential elements for building reliable smart cities that are characterized by the following:

- *Low latency* – Most smart city applications are real-time, requiring low latency. For example, smart traffic light changers change from stop to go based on current traffic conditions [117]. The EC is a promising paradigm where data processing can be done at the network edge, decreasing the latency compared with the CC with a longer response time.
- *Energy consumption* – Reducing energy consumption in smart cities is essential, given the number of devices and sensors involved. One of the solutions is to reduce data travelling to CC for computation. By processing data closer to the devices, EC helps to reduce overall energy consumption. Moreover, tasks can be shared among available EC devices to mitigate any energy-constrained devices [17].
- *Security and privacy* – Smart cities involve the large-scale deployment of devices exposed to potential security cyber-attacks when connected to the internet [124]. Furthermore, as more devices are connected to the internet and generate data, there is a risk that this data could be used to track individuals or invade their privacy.

The SDN can help mitigate security and privacy issues by providing a centralized network control point. This centralization view can be used to implement security policies and controls that can be easily managed and updated.

Table 4. – Recent works addressing IoT challenges with EC-IoT, SDIoT and SDN-EC-IoT paradigm

Ref	Year	IoT Challengers								IoT Applications	Architecture		
		Interoperability	Scalability	Low latency	Handling of big data	Security and privacy	Energy consumption	QoS	Task Offloading		EC-IoT	SDIoT	SDN-EC-IoT
[22]	2022				✓		✓			Smart Cities		✓	
[16]	2022		✓				✓			Smart Cities			✓
[17]	2022			✓			✓			Smart Cities			✓
[8]	2022						✓	✓	✓	Smart Healthcare	✓		
[125]	2022				✓	✓				Smart Healthcare	✓		
[120]	2022		✓	✓						Smart Healthcare		✓	
[123]	2022			✓			✓			Smart Transportation	✓		
[68]	2022	✓		✓	✓	✓				Smart Transportation	✓		
[118]	2022								✓	Smart Transportation		✓	
[126]	2022		✓	✓						Smart Transportation		✓	
[127]	2022				✓			✓		Smart Home	✓		
[9]	2022		✓				✓		✓	Smart Home	✓		
[128]	2022			✓		✓				Smart Home	✓		
[129]	2022			✓			✓			Smart Home		✓	
[130]	2022					✓				Smart Home		✓	
[131]	2022						✓		✓	Smart Home			✓
[132]	2022	✓						✓	✓	Smart Grid		✓	
[58]	2021		✓	✓						Smart Cities		✓	
[14]	2021		✓				✓			Smart Cities		✓	
[19]	2021			✓			✓			Smart Cities			✓
[133]	2021			✓	✓		✓			Smart Cities		✓	
[10]	2021		✓		✓			✓		Smart Cities	✓		
[113]	2021		✓	✓						Smart Healthcare	✓		
[134]	2021			✓			✓			Smart Healthcare			✓
[135]	2021			✓			✓			Smart Healthcare	✓		
[109]	2021						✓	✓	✓	Smart Healthcare	✓		
[122]	2021					✓	✓			Smart Healthcare		✓	
[136]	2021		✓	✓					✓	Smart Healthcare			✓
[111]	2021		✓	✓				✓		Smart Healthcare			✓

[92]	2021			✓					✓			Smart Healthcare		✓	
[112]	2021				✓				✓			Smart Transportation	✓		
[137]	2021			✓		✓						Smart Transportation	✓		
[138]	2021				✓				✓			Smart Transportation		✓	
[139]	2021	✓								✓		Smart Transportation	✓		
[140]	2021	✓								✓		Smart Transportation			✓
[141]	2021						✓		✓			Smart Home	✓		
[142]	2021				✓		✓		✓			Smart Home	✓		
[143]	2021						✓					Smart Home		✓	
[144]	2021					✓	✓					Smart Home		✓	
[116]	2021						✓		✓			Smart Grid	✓		
[145]	2021			✓			✓					Smart Grid		✓	
[81]	2021			✓			✓					Smart Grid		✓	
[119]	2021	✓		✓							✓	Smart Grid		✓	
[146]	2021			✓	✓							Smart Grid	✓		
[124]	2020						✓					Smart Cities			✓
[117]	2020	✓							✓			Smart Cities		✓	
[71]	2020						✓		✓		✓	Smart Cities	✓		
[114]	2020			✓		✓						Smart Cities	✓		
[147]	2020			✓			✓					Smart Cities	✓		
[148]	2020						✓		✓			Smart Cities		✓	
[18]	2020			✓			✓					Smart Healthcare			✓
[149]	2020	✓					✓					Smart Healthcare	✓		
[150]	2020				✓				✓			Smart Healthcare		✓	
[151]	2020			✓							✓	Smart Transportation		✓	
[152]	2020			✓	✓							Smart Transportation			✓
[153]	2020			✓							✓	Smart Transportation			✓
[154]	2020			✓						✓		Smart Grid		✓	
[155]	2019				✓				✓			Smart Cities	✓		
[156]	2019	✓							✓			Smart Cities		✓	

5.2 SMART HEALTHCARE

Smart healthcare involves using connected devices and sensors to gather, transmit, and analyze health-related data to improve healthcare outcomes [134]. This can include simple devices such as wearable fitness trackers, smartwatches, and home monitoring systems that collect data on heart rate, activity levels, and sleep patterns. Examples of specialized devices for healthcare are medical devices like glucose monitors for diabetics and smart inhalers for asthma patients. The data collected by these devices can provide patients and healthcare providers with real-time insights into a patient's health, allowing for more personalized and proactive care. Moreover, it allows remote patient monitoring between patients, doctors, caregivers, and families, regardless of location [113]. One of the main concerns in smart healthcare is data management. It contains important and private information that must be analyzed and processed efficiently and securely.

EC and SDN guarantee the overall security of smart healthcare [122]. It provides security measures by authenticating and authorizing users. Furthermore, encryption and blockchain-based security methods ensure the data stored is safe. SDN can be used to conduct load balancing, network optimization, and optimal resource use among connected EC in the healthcare system [135].

5.3 SMART TRANSPORTATION

Smart transportation systems focus on improving various aspects of transportation, such as traffic management, vehicle tracking, public transportation, and parking management. It uses sensors, connectivity, and data analytics to provide real-time information, automate processes, and optimize the use of resources [68]. An example of a smart transportation system is the intelligent traffic system (ITS). ITS uses cameras and sensors to detect and report traffic congestion, accidents, and other incidents in real-time, allowing traffic to be rerouted and reducing wait times for drivers [126]. Smart transportation also involves the concept of the Internet of Vehicles (IoV).

By integrating internet and communications technologies, each vehicle can communicate with each other and the infrastructure around them. This allows vehicles to share real-time information such as location, speed, and traffic conditions, enabling new features and services such as improved traffic management, enhanced navigation, and increased safety [139]. Different applications in smart transportation benefit from EC and SDN, including:

- *Smart parking systems*: By using sensors to detect the presence of cars in parking spots and providing this information to a centralized system, drivers can locate and reserve available spots in real-time[112].
- *Traveller information*: Each travelling user is provided with real-time information such as travel time, travel speed, delay, accidents on roads, changes in route, diversions, work zone conditions, and so on [118].
- *Public transportation systems*: Connected bus or train systems allow riders to track the location of their vehicle in real-time, providing more accurate information about arrival times and helping to reduce wait times [138].
- *Safety applications*: Dedicated short-range communication enables vehicles to communicate with each other and the transportation infrastructure, allowing for features such as collision avoidance and lane change assistance [152].

5.4 SMART HOME

The smart home focuses on allowing intelligent control of smart devices inside the home connected to the internet, such as TVs, air conditioners, thermostats, lighting, security cameras, and smart speakers [9]. Different applications in the smart home benefit from EC and SDN, including:

- *Home automation*: the ability to remotely control various devices and appliances in the home using a smartphone or computer [157]. This can include adjusting the thermostat, turning lights on and off, and controlling appliances such as washing machines and dryers.
- *Home security*: smart home devices such as security cameras, door locks, and motion sensors can be used to enhance home security [144]. These devices can be linked to a home automation system, allowing homeowners to monitor their homes remotely and receive alerts for suspicious activity.
- *Energy management*: smart home devices such as thermostats and lighting can be used to manage energy consumption in the home [129]. For example, smart thermostats can learn a homeowner's schedule and adjust the temperature to save energy.

Smart home devices generate a large amount of data for smart control and decision-making inside homes. Data processing, analysis, and storage require huge resources to guarantee QoS and the services' continuity. The EC offers a scalable and distributed architecture to support smart homes by processing the data at the network edge. Doing so provides fast and efficient data processing with low energy consumption. On the other hand, the SDN provides centralized management control of the home's network. Centralized control can be used to monitor and control network access and implement security policies specific to the smart home environment.

5.5 SMART GRID

A Smart grid enables two-way communication between the power grid and the electricity consumers, allowing for real-time monitoring and control of the power grid [145]. On the consumer side, homes and businesses are equipped with smart meters to provide real-time data on energy consumption. This allows for more accurate billing and enables consumers to make more informed decisions about energy use. Power grid providers or utility companies analyze the real-time data using a centralized Supervisory Control and Data Acquisition (SCADA) server [81]. It allows remote adjustment to improve efficiency and reliability. Additionally, SCADA systems can automate equipment control, such as turning on or off power generators, and quickly respond to issues such as power outages.

With the EC paradigm, the SCADA model can be decentralized and turned into a hierarchical architecture to improve reliability and reduce latency [119]. The edge layer controls the microgrid and exchanges information with the neighbouring edge and the higher tiers. SCADA at the top tier is responsible for data analytics. On the other hand, SDN provides centralized management control among SCADA to improve the grid's security [132]. In addition, SDN can be used to optimize the communication between SCADA systems and other devices in the smart grid, such as power generators, transformers, and substations, which can improve the efficiency of the grid.

6. SDN-EC-IOT APPROACH

In this section, the review covers existing work based on the SDN-EC-IoT architecture to address underlying IoT challenges. Table 5 outlines the main features of such proposed applications. To address the interoperability in the IoT, Fawwaz et al. [16] proposed a distributed Message Queuing Telemetry Transport (MQTT) broker-optimized architecture. MQTT is a lightweight IoT-enabled exchange protocol that uses a publish and subscribe structure via a centralized broker to share data. By deploying a distributed MQTT broker for edge resources, the goal is to reduce network traffic and data delivery latency by only managing consumed topics in the network. An integer non-linear programming approach was formulated to optimize container placement and prevent the waste of EC resources. It uses the Distributed Broker and Edge (DBE) Manager and SDN controller to track all network container resource usage and each edge device’s available resources. The proposed solution improved performance by lowering the deployment failure ratio, power consumption, network usage, and synchronization overhead. However, this work does not consider low latency, required memory, or storage concerns.

Lin et al. [152] introduced a distributed mobile fog computing scheme to optimize the scheduling of delay-sensitive applications in vehicular networks, addressing scalability concerns in the IoT. By leveraging SDN, they partition vehicular networks into network, fog, and control layers. Their approach offers two solutions for mitigating delay-sensitive data scheduling challenges and the Multiple Time-constrained Vehicular applications Scheduling (MTVS) issue. The first solution involves processing delay-sensitive applications at the data level within the network layer and distributing data across fog computing units via a mobile distributed fog computing scheme. The second solution employs an efficient hybrid scheduling algorithm to tackle the MTVS problem. Simulation results demonstrate the scheme's effective utilization of network resources, surpassing recent efforts, particularly in achieving a higher success rate in resolving the MTVS issue. However, the applicability of this approach in real-world environments remains an open question.

To address the low latency requirement in the IoT, Ren et al. [136] proposed a centralized, secure, and fast task offloading strategy. This approach empowers resource-constrained edge devices for efficient task offloading. On the other hand, SDN optimizes global decisions, selecting the best fog node for secure and low-latency task processing. To enhance network security comprehensively, a two-layer blockchain-based security method is deployed. It secures critical decision outcomes and SDN inter-domain signaling at the control layer while safeguarding task-specific data and intra-domain signaling at the fog layer. Simulations reveal significant system performance improvements in efficiency, reliability, and security. However, energy consumption remains unaddressed in this approach.

Besides that, another work by Bardalai et al. [111] proposed OpenHealthQ, an OpenFlow-based traffic shaping model tailored for healthcare data management. OpenHealthQ offers secure, on-demand, and cost-effective access to healthcare-centric computing infrastructure within a distributed cloud architecture featuring SDN-based fog nodes at the network's edge. This model categorizes data from diverse devices into priority classes based on criteria such as priority level, throughput, and privacy requirements within the application. Experimental results highlight that OpenHealthQ significantly reduces end host response times and substantially enhances network throughput compared to the conventional Best-Effort (BE) approach. However, it's important to note that this approach primarily focused on limited traffic types.

To mitigate the security and privacy issues in IoT, Dantas Silva et al. [124], proposed a SDN-EC-IoT framework aimed at addressing security and privacy concerns, particularly regarding DDoS attacks. Their algorithm employs the cosine similarity method to compare the arrival rate of packets at the controller with a predefined limit. This approach identifies devices that exceed the network's handling capacity by sending excessive packets. Simulation results demonstrate the algorithm's effectiveness in mitigating DDoS attacks triggered by high traffic rates. Nevertheless, it's important to note that this approach does not consider energy consumption as a factor.

Table 5. – Summary of IoT applications using SDN-EC-IoT

Ref	Purpose	Contribution	Limitation	IoT Applications
[16]	A distributed architecture using EC with SDN to track available resources.	Interoperability	Require high memory and storage.	Smart Cities
[152]	Fog-based architecture for delay-sensitive applications. SDN controls the network layer.	Scalability	Require security mechanism.	Smart Transportation
[136]	EC and SDN optimized decisions for task offloading.	Low latency	Require some mechanism for energy consumption.	Smart Healthcare
[111]	A distributed EC and SDN architecture based on OpenFlow model, OpenHealthQ.	Low latency	Limited traffic type	Smart Healthcare
[124]	A SDN-EC-IoT based framework to mitigate DDoS attack.	Security and privacy	Require some mechanism for energy consumption.	Smart Cities
[18]	A secure framework for SDN-based EC in IoT	Security and privacy	Require real environment implementation.	Smart Healthcare
[19]	SDN based framework for urban monitoring with distributed processing by EC.	Energy consumption	Huge computational resources.	Smart Cities
[17]	SDN-EC based task scheduling.	Energy consumption	Require task offloading mechanism.	Smart Cities
[134]	SDN-EC based decision making and task allocation scheme.	Energy consumption	Require real environment implementation.	Smart Healthcare
[139]	SDN-Fog based architecture efficient resources usage.	QoS	Huge computational resources.	Smart Transportation
[140]	SDN-EC based architecture for resource placement.	QoS	Require some mechanism for energy consumption.	Smart Transportation
[153]	SDN-EC framework for task offloading scheme based on data collected.	Task offloading	Require real environment implementation.	Smart Transportation
[131]	SDN-EC framework for task offloading scheme based on data preprocessing.	Task offloading	Require real environment implementation.	Smart Homes

On the other hand, Li et al. [18] proposed a secure framework for SDN-based EC within IoT-enabled healthcare systems. In this framework, IoT devices undergo lightweight authentication by edge servers. Once authenticated, these devices gather patient data, which is then transmitted to edge servers for storage, processing, and analysis. The edge servers are connected to an SDN controller, responsible for load balancing, network optimization, and efficient resource utilization within the healthcare system. Simulation results indicate improved network performance, including reduced average response times, enhanced packet delivery ratios, lower latency, increased throughput, and reduced network control overhead. However, it's essential to note that this framework does not account for privacy considerations or address real-world implementation challenges.

To address the energy consumption in the IoT, Khazael et al. [19] proposed an innovative architecture for urban monitoring in response to IoT energy consumption concerns. This architecture integrates the publish-subscribe pattern with SDN, fostering distributed processing and enhancing complex event detection within monitoring applications. The architecture also accommodates the TESLA complex event definition language, allowing application developers to specify QoS requirements. Simulation results highlight substantial reductions in energy consumption and data packet traffic when compared to three baseline methods: Event data exchange, Distributed coordination protocol for event data exchange, and MusaNet. Nevertheless, applying this approach to a dense network presents complex challenges.

On the other hand, Sellami et al. [17] proposed an algorithm focusing on energy-efficient and low-latency-oriented task scheduling. Their approach formulates the online task assignment and scheduling problem as an energy-constrained Deep Q-Learning process. Deep Q-Learning is utilized to minimize network latency while ensuring energy

efficiency, particularly in conserving battery power within application-specific constraints. In response to the dynamic task arrival process, they introduce a deep reinforcement learning (DRL) approach for task scheduling and assignment in SDN-EC-IoT networks. Simulation results indicate that this approach outperforms three other deep learning algorithms (deterministic, random, and A3C agents) to reduce communication latency and enhance energy efficiency. However, it's worth noting that this algorithm does not account for task offloading to alternative edge nodes.

Besides that, another work by Saha et al. [134] proposed an edge-based decision-making and task allocation scheme (EDT) for SDN healthcare. This scheme employs machine learning (ML) to predict flow criticality and mobile device locations. The SDN controller deploys the required EDT module to each edge node based on these predictions. ML-based trajectory predictions enable the anticipation of mobile device network locations, facilitating the dynamic assignment of compute tasks to edge nodes. Simulation results reveal superior performance in terms of latency, energy consumption, and packet delivery when compared to existing ML algorithms like decision trees, random forests, K-nearest neighbours (KNN), Naïve Bayes (NB), and deep neural networks (DNN). However, the practical application of this approach in real-world settings remains an open question.

To mitigate the QoS requirement in IoT, Cao et al. [139] proposed a 5G Internet of Vehicles (IoV) architecture, integrating fog computing and SDN to enhance QoS in IoT settings. Their approach efficiently harnesses heterogeneous computing resources to ensure QoS. They devised an improved optimization algorithm, building upon the Two_Arch2 algorithm, by incorporating hierarchical clustering. This optimization model factors in service delay, task execution stability, energy consumption, and load balancing within the IoV context. Experimental results demonstrate the superiority of this improved algorithm in achieving resource allocation compared to similar approaches. Nevertheless, the practical applicability of this proposed approach in real-world scenarios remains an open question.

On the other hand, Li et al. [140] proposed a three-layer hierarchical control framework for SDN-IoV, integrating MEC. The aim was to address the SDN controller placement challenge, focusing on minimizing the delay between switches and controllers. Their framework introduces a controller placement policy based on the Louvain algorithm to determine optimal controller locations while considering load balance and buffer size constraints. Simulation results demonstrate that this algorithm outperforms two baseline approaches in terms of delay and load balance. However, it's important to note that this approach did not consider energy consumption.

To address the task offloading requirement in IoT, Guo et al. [153] proposed an SDN-enhanced vehicular EC network to address task offloading requirements in the IoT. This network facilitates centralized data and information management across the entire system to minimise processing delays associated with task offloading. Their approach employs an intelligent task offloading scheme based on Deep Q Learning, designed to adapt to the dynamically changing environment. Simulation results indicate that this scheme yields reduced processing delays compared to alternative task offloading algorithms. However, it's worth noting that further validation is necessary, as the simulation considered only a single unidirectional road scenario.

On the other hand, Li et al. [131] proposed an EC network task offloading model for a smart city, emphasizing data preprocessing. The model encompasses three key elements: the SDN controller, offload decision-making, and resource allocation. The primary goal is to enhance energy efficiency in passive house designs to reduce energy consumption in urban buildings during their operation. Simulation results indicate the feasibility of this model and its ability to achieve substantial energy-saving efficiency gains. However, the practical implementation of this approach in real-world environments remains an open question.

7. FUTURE RESEARCH DIRECTION

Despite observing the advantages of current solutions for integrating EC and SDN in IoT, multiple issues and challenges need to be investigated for integrating SDN-EC-IoT on top of IoT applications. This section aims to bring attention to the challenges that currently exist, and to spark conversation about the new ideas and future directions that the research community must address. Table 6 summarizes the challenges and future directions.

7.1 INTEROPERABILITY PLATFORM

Interoperability is indispensable for any network architecture in which the system must manage the exchange of information among heterogeneous IoT devices, platforms, and systems. Recently, the number of IoT solution providers has increased rapidly, with each platform having its preferred operating system, architecture, programming language, and data structure. This dissimilarity is a challenge to creating cross-platform and cross-domain IoT applications. For this reason, the SDN-EC-IoT should provide interoperability at the platform level by providing some cross-platform functions. This cross-platform function will act as a bridge to integrate all IoT platforms available in the network.

7.2 SCALABLE ARCHITECTURE

Scalability is a critical consideration for network architectures, especially in IoT ecosystems, where the system must accommodate the increasing workloads stemming from software, hardware, and networks. This scalability imperative is essential while simultaneously upholding each service's Quality of Service (QoS). It is known that the number of connected things is increasing, which may interrupt existing services and cause a network bottleneck due to

the huge quantity of data generated. For this reason, SDN-EC-IoT should guarantee scalability by applying load balancing and prioritizing types of service.

Table 6. – SDN-EC-IoT challenges and future direction

Topic	Challengers	Future Direction
Interoperability platform	• Network interoperability • Platform interoperability	• Cross-platform function
Scalable architecture	• High number of data • High number of services	• Load balance • Prioritize service
Low latency guarantee	• Limited network resources • High number of services	• Optimization • Prioritize service
Efficiency in handling big data	• Data processing • Data storage	• Data filtering • Data fusion
Security and privacy enhancement	• Network security • Data privacy • Authentication	• Layered security modal • Efficient encryption
Energy consumption optimization	• Network energy consumption • Device energy consumption	• Optimization • Artificial Intelligence
QoS guarantee	• Limited network resources • High number of services	• Traffic classification • Artificial Intelligence
Resource aware task offloading	• Limited resources • Offloading decision	Task classification
Machine Learning in SDN-EC-IoT	• Processing resource • Energy consumption • Storage limitation	• Data augmentation • Transfer learning • Model compression

7.3 LOW LATENCY GUARANTEE

Each IoT service has its requirements for low latency and high reliability. In IoT ecosystems, limited network resources must be managed efficiently and assigned to multiple IoT services. The system must guarantee all IoT services are given the required network resources to function without interruption. For this reason, it is necessary to propose algorithms focusing on optimizing the usage of network resources and prioritizing IoT services to ensure each service can function without interruption.

7.4 EFFICIENCY IN HANDLING BIG DATA

In an IoT environment, the amount of data collected from real-time sensors is huge. This huge amount of data must be managed efficiently in processing and storage. Obtaining sufficient and high-quality data is crucial to ensuring the accuracy of the later stages of data processing and analytic operations. For this reason, future work must address resolving unnecessary data, such as duplication and redundancy, for efficient storage usage in SDN-EC-IoT.

7.5 SECURITY AND PRIVACY ENHANCEMENT

Security is a paramount concern in the complex architecture of SDN-EC-IoT networks, where the involvement of multiple users, IoT devices, and data introduces the potential for security breaches. To address this, a hierarchical approach incorporating various mechanisms like cryptography, hash functions, and blockchain is essential to ensure the security of communications.

Other than that, data privacy is also another critical issue within SDN-EC-IoT, as vast amounts of data flow between communication nodes containing sensitive user information are susceptible to exploitation by unauthorized entities. Future efforts must prioritize data privacy by implementing encryption techniques like Public Key Infrastructure (PKI) and Elliptic Curve Cryptography (ECC), especially for resource constrained IoT devices at the perception layer. It is also worth noting that IoT devices, especially at the perception layer often lack the computational resources (CPU and memory) required for cryptographic operations used in authentication. In such cases, IoT devices can delegate the authentication process to the nearest edge controller to execute the authentication procedure efficiently.

7.6 ENERGY CONSUMPTION OPTIMISATION

The SDN-EC-IoT architecture consists of multi-layer, different distributed systems. The energy consumption is expected to be significant, increasing the overall cost. Therefore, future work must address this problem by developing a new or optimizing an existing energy-efficient protocol without compromising the IoT services.

7.7 QOS GUARANTEE

QoS is a vital network aspect that guarantees its capability to run high-priority IoT applications. By prioritizing the transmission of different data types, QoS regulates network resources to meet the needs of numerous data transmissions. To maintain an acceptable level of QoS for high-priority IoT applications, QoS approaches such as traffic classification and traffic shaping at each layer within the SDN-EC-IoT architecture.

7.8 RESOURCE AWARE TASK OFFLOADING

Task offloading is essential in transferring computational tasks from an IoT device to SDN-EC-IoT infrastructure to improve overall system performance and conserve local resources. It requires careful consideration of each task's network connectivity, available resources, and time constraints. For this, future work can look into utilizing the SDN function to identify available resources as one factor determining EC's task-offloading process.

7.9 MACHINE LEARNING IN SDN-EC-IOT

ML involves building algorithms that can learn from data and improve performance over time without being explicitly programmed. Recently, ML has been used widely in IoT applications to improve functionality and efficiency in anomaly detection, predictive maintenance, traffic analysis, resource optimization, intelligent decision-making, and personalization. However, there are still important challenges and future directions to integrate the SDN-EC-IoT with ML efficiently, as listed below:

Power and processing resource utilization - ML models are divided into two sections: training and inferencing. Both parts can be executed at the edge layer and SDN control plane, increasing the demand for processing power, especially for the training part. This increases energy consumption overall, which will cause long-term problems. Therefore, the challenge for future implementation is finding a balance between power requirements and processing capacity to support ML implementation in SDN-EC-IoT.

Storage requirement – In general, larger datasets can help improve the accuracy and generalization of ML models. It provides more diverse and representative examples for the model to learn from. However, a larger dataset requires large storage space, creating a critical issue for SDN-EC-IoT. For this, the research community needs to focus on proposing efficient solutions that balance the amount of training data with the complexity of the model and the available computational resources based on specific IoT applications.

8. CONCLUSION

The growth of IoT devices in the future is expected to be exponential, with an estimated 41.6 billion IoT devices in use by 2025 [158]. To support this growth, there is a need for reliable and scalable networks to connect these devices. It requires significant bandwidth, processing power, storage capacity, and the ability to handle large volumes of data in real time. SDN and EC can help to address these challenges by providing a flexible, scalable, and secure network infrastructure that can support the increasing demand for IoT devices. SDN enables centralized network management and automation, while EC can bring processing power closer to IoT devices, reducing latency and improving data processing and analysis. Together, SDN-EC-IoT can provide the necessary infrastructure to support the growth of IoT devices and enable new applications and services.

This paper compiles 127 articles published between 2019 and 2023 and analyses the technical and implementation aspects of 74 of them. The main findings highlight the effective integration of SDN and EC technologies, which can significantly support the expansion of IoT devices and enable the development of novel applications and services. SDN plays a crucial role in centralizing network management and automation, while EC brings processing capabilities closer to IoT devices, thus reducing latency and enhancing data processing efficiency. This synergy between SDN and EC provides a promising foundation for IoT advancement.

This paper highlights valuable insights into future research directions within SDN-EC-IoT. It advocates for the development of interoperability platforms, scalable architectures, guarantees for low latency and QoS, improved handling of big data, strengthened security and privacy features, optimized energy consumption, resource-aware task offloading, and the incorporation of machine learning. These research avenues are essential for ensuring the continued success and growth of IoT applications.

This paper discusses practical case studies that exemplify the integration of SDN and EC in various IoT applications, such as smart cities, healthcare, transportation, homes, and grids. However, this paper acknowledges that some of these proposals require further validation in real-world scenarios, emphasizing the need for practical testing and implementation. Lastly, future research must find an optimal balance based on specific IoT applications. It is

important to note that this paper emphasizes the need for future research to explore these points further and improve SDN-EC-IoT integration, considering the expected exponential growth in IoT applications.

Funding

None

ACKNOWLEDGEMENT

This work was supported by Universiti Malaysia Pahang Al-Sultan Abdullah under the Postgraduate Research Grant Scheme (PGRS23039).

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

[1] M. Chui, M. Collins, and M. Patel, "The Internet of Things: Catching up to an accelerating opportunity," no. November, 2021.

[2] T. Cisco and A. Internet, "Cisco: 2020 CISO Benchmark Report," *Comput. Fraud Secur.*, vol. 2020, no. 3, pp. 4–4, 2020, doi: 10.1016/s1361-3723(20)30026-9.

[3] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and Open Challenges," *Mob. Networks Appl.*, vol. 24, no. 3, pp. 796–809, 2019, doi: 10.1007/s11036-018-1089-9.

[4] A. Cimmino, M. Poveda-Villalón, and R. García-Castro, "EWOT: A semantic interoperability approach for heterogeneous IoT ecosystems based on the web of things," *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030822.

[5] 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.*, vol. 76, no. 7, pp. 5175–5191, 2020, doi: 10.1007/s11227-019-02773-1.

[6] G. Nebbione and M. C. Calzarossa, "Security of IoT application layer protocols: Challenges and findings," *Futur. Internet*, vol. 12, no. 3, pp. 1–20, 2020, doi: 10.3390/fi12030055.

[7] S. Shukla, M. F. Hassan, D. C. Tran, R. Akbar, I. V. Paputungan, and M. K. Khan, "Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR)," *Cluster Comput.*, vol. 3, 2021, doi: 10.1007/s10586-021-03279-3.

[8] M. Alrazgan, "Internet of Medical Things and Edge Computing for Improving Healthcare in Smart Cities," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/5776954.

[9] Z. Sharif, L. T. Jung, M. Ayaz, M. Yahya, and D. Khan, "Smart Home Automation by Internet-of-Things Edge Computing Platform," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 474–484, 2022, doi: 10.14569/IJACSA.2022.0130455.

[10] Z. Lv, D. Chen, R. Lou, and Q. Wang, "Intelligent edge computing based on machine learning for smart city," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 90–99, 2021, doi: 10.1016/j.future.2020.08.037.

[11] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, no. 6, 2019, doi: 10.1145/3362031.

[12] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, "Complementing IoT Services through Software Defined Networking and Edge Computing: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1761–1804, 2020, doi: 10.1109/COMST.2020.2997475.

[13] A. M. Farooqi, M. A. Alam, S. I. Hassan, and S. M. Idrees, "A Fog Computing Model for VANET to Reduce Latency and Delay Using 5G Network in Smart City Transportation," *Appl. Sci.*, vol. 12, no. 4, 2022, doi: 10.3390/app12042083.

[14] H. Babbar, S. Rani, D. Gupta, H. M. Aljahdali, A. Singh, and F. Al-Turjman, "Load balancing algorithm on the immense scale of internet of things in sdn for smart cities," *Sustain.*, vol. 13, no. 17, pp. 1–16, 2021, doi: 10.3390/su13179587.

[15] V. Balasubramanian, M. Aloqaily, M. Reisslein, and A. Scaglione, "Intelligent Resource Management at the Edge for Ubiquitous IoT: An SDN-Based Federated Learning Approach," *IEEE Netw.*, vol. 35, no. 5, pp. 114–121, 2021, doi: 10.1109/MNET.011.2100121.

[16] D. Z. Fawwaz, S. H. Chung, C. W. Ahn, and W. S. Kim, "Optimal Distributed MQTT Broker and Services Placement for SDN-Edge Based Smart City Architecture," *Sensors*, vol. 22, no. 9, 2022, doi: 10.3390/s22093431.

[17] B. Sellami, A. Hakiri, S. Ben Yahia, and P. Berthou, "Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network," *Comput. Networks*, vol. 210, no. April, p. 108957, 2022, doi: 10.1016/j.comnet.2022.108957.

[18] J. Li *et al.*, "A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System," *IEEE*

Access, vol. 8, pp. 135479–135490, 2020, doi: 10.1109/ACCESS.2020.3011503.
[19] B. Khazael, H. T. Malazi, and S. Clarke, "Complex Event Processing in Smart City Monitoring Applications," *IEEE Access*, vol. 9, pp. 143150–143165, 2021, doi: 10.1109/ACCESS.2021.3119975.
[20] V. K. Rathi *et al.*, "An edge AI-enabled IoT healthcare monitoring system for smart cities," *Comput. Electr. Eng.*, vol. 96, no. PB, p. 107524, 2021, doi: 10.1016/j.compeleceng.2021.107524.
[21] A. Das, S. Chakraborty, and S. Chakraborty, "Where do all my smart home data go? Context-aware data generation and forwarding for edge-based microservices over shared IoT infrastructure," *Futur. Gener. Comput. Syst.*, vol. 134, pp. 204–218, 2022, doi: 10.1016/j.future.2022.03.027.
[22] F. Alassery, "High Performance Priority Packets Scheduling Mechanism for Big Data in Smart Cities," *Comput. Mater. Contin.*, vol. 72, no. 1, pp. 535–559, 2022, doi: 10.32604/cmc.2022.023558.
[23] D. Arellanes and K. K. Lau, "Evaluating IoT service composition mechanisms for the scalability of IoT systems," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 827–848, 2020, doi: 10.1016/j.future.2020.02.073.
[24] Siddiqi, Yu, J. Joung, M. A. Siddiqi, H. Yu, and J. Joung, "2019 5G Ultra-Reliable Low-Latency Communication.pdf," *Electronics*, vol. 8, no. 9. p. 981, 2019, [Online]. Available: https://www.mdpi.com/2079-9292/8/9/981.
[25] I. F. Siddiqui, N. M. F. Qureshi, B. S. Chowdhry, and M. A. Uqaili, "Edge-node-aware adaptive data processing frameworfor smart grid," *Wirel. Pers. Commun.*, vol. 106, no. 1, pp. 179–189, 2019, doi: 10.1007/s11277-019-06264-7.
[26] M. binti Mohamad Noor and W. H. Hassan, "Current research on Internet of Things (IoT) security: A survey," *Comput. Networks*, vol. 148, pp. 283–294, 2019, doi: 10.1016/j.comnet.2018.11.025.
[27] M. Laroui, B. Nour, H. Moungla, M. A. Cherif, H. Afifi, and M. Guizani, "Edge and fog computing for IoT: A survey on current research activities & future directions," *Comput. Commun.*, vol. 180, no. September, pp. 210–231, 2021, doi: 10.1016/j.comcom.2021.09.003.
[28] Imran, Z. Ghaffar, A. Alshahrani, M. Fayaz, A. M. Alghamdi, and J. Gwak, "A topical review on machine learning, software defined networking, internet of things applications: Research limitations and challenges," *Electron.*, vol. 10, no. 8, 2021, doi: 10.3390/electronics10080880.
[29] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1–52, 2020, doi: 10.3390/s20226441.
[30] M. Al Ja'afreh, H. Adhami, A. E. Alchalabi, M. Hoda, and A. El Saddik, "Toward integrating software defined networks with the Internet of Things: a review," *Cluster Comput.*, vol. 4, 2021, doi: 10.1007/s10586-021-03402-4.
[31] P. P. Ray and N. Kumar, "SDN/NFV architectures for edge-cloud oriented IoT: A systematic review," *Comput. Commun.*, vol. 169, no. January, pp. 129–153, 2021, doi: 10.1016/j.comcom.2021.01.018.
[32] Y. Li *et al.*, "Enhancing the internet of things with knowledge-driven software-defined networking technology: Future perspectives," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–20, 2020, doi: 10.3390/s20123459.
[33] W. Rafique *et al.*, "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–20, 2020, doi: 10.1016/j.comcom.2021.09.003.
[34] S. S. Jazaeri, S. Jabbehdari, P. Asghari, and H. Haj Seyyed Javadi, "Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions," *Cluster Comput.*, vol. 24, no. 4, pp. 3187–3228, 2021, doi: 10.1007/s10586-021-03311-6.
[35] M. Al Ja'afreh, H. Adhami, A. E. Alchalabi, M. Hoda, and A. El Saddik, "Toward integrating software defined networks with the Internet of Things: a review," *Cluster Comput.*, vol. 25, no. 3, pp. 1619–1636, 2022, doi: 10.1007/s10586-021-03402-4.
[36] M. Laroui, B. Nour, H. Moungla, M. A. Cherif, H. Afifi, and M. Guizani, "Edge and fog computing for IoT: A survey on current research activities & future directions," *Comput. Commun.*, vol. 180, no. September, pp. 210–231, 2021, doi: 10.1016/j.comcom.2021.09.003.
[37] W. Rafique *et al.*, "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV," *Sensors (Switzerland)*, vol. 20, no. 12, pp. 1–20, 2020, doi: 10.1016/j.comcom.2021.09.003.
[38] International Telecommunication Union, "ITU-T Y.2060 Overview of the Internet of Things," 2012. https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2060-201206-I!!PDF-E&type=items.
[39] R. Morabito and J. Jimenez, "IETF Protocol Suite for the Internet of Things: Overview and Recent Advancements," *IEEE Commun. Stand. Mag.*, vol. 4, no. 2, pp. 41–49, 2020, doi: 10.1109/MCOMSTD.001.1900014.
[40] E. Lee, Y. D. Seo, S. R. Oh, and Y. G. Kim, "A Survey on Standards for Interoperability and Security in the Internet of Things," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 2, pp. 1020–1047, 2021, doi: 10.1109/COMST.2021.3067354.
[41] T. Domínguez-Bolaño, O. Campos, V. Barral, C. J. Escudero, and J. A. García-Naya, "An overview of IoT architectures, technologies, and existing open-source projects," *Internet of Things (Netherlands)*, vol. 20, p. 100626, 2022, doi: 10.1016/j.iot.2022.100626.
[42] A. Sadeghi-niaraki, "Internet of Thing (IoT) review of review : Bibliometric overview since its foundation,"

Futur. Gener. Comput. Syst., vol. 143, pp. 361–377, 2023, doi: 10.1016/j.future.2023.01.016.
[43] P. Zhang, C. Jiang, X. Pang, and Y. Qian, "STEC-IoT: A Security Tactic by Virtualizing Edge Computing on IoT," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2459–2467, 2021, doi: 10.1109/JIOT.2020.3017742.
[44] M. Babar, M. S. Khan, F. Ali, M. Imran, and M. Shoaib, "Cloudlet Computing: Recent Advances, Taxonomy, and Challenges," *IEEE Access*, vol. 9, pp. 29609–29622, 2021, doi: 10.1109/ACCESS.2021.3059072.
[45] M. Muneeb, K. M. Ko, and Y. H. Park, "A fog computing architecture with multi-layer for computing-intensive iot applications," *Appl. Sci.*, vol. 11, no. 24, 2021, doi: 10.3390/app112411585.
[46] Y. Zhang, Y. Zhang, X. Lan, J. Ren, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1227–1240, 2020, doi: 10.1109/TNET.2020.2979807.
[47] J. C. Correa Chica, J. C. Imbachi, and J. F. Botero Vega, "Security in SDN: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 159, no. March, p. 102595, 2020, doi: 10.1016/j.jnca.2020.102595.
[48] E. Haleplidis, S. Denazis, J. H. Salim, O. Koufopavlou, D. Meyer, and K. Pentikousis, "SDN Layers and Architecture Terminology," vol. RFC7426, pp. 1–35, 2015.
[49] L. Yang, B. Ng, W. K. G. Seah, L. Groves, and D. Singh, "A survey on network forwarding in Software-Defined Networking," *J. Netw. Comput. Appl.*, vol. 176, no. October 2020, p. 102947, 2021, doi: 10.1016/j.jnca.2020.102947.
[50] R. Yang, X. Chang, J. Mišić, and V. B. Mišić, "Performance Modeling of Linux Network System with Open vSwitch," *Peer-to-Peer Netw. Appl.*, vol. 13, no. 1, pp. 151–162, 2020, doi: 10.1007/s12083-019-00723-5.
[51] B. Oconnor *et al.*, "Using P4 on fixed-pipeline and programmable stratum switches," *2019 ACM/IEEE Symp. Archit. Netw. Commun. Syst. ANCS 2019*, pp. 7–8, 2019, doi: 10.1109/ANCS.2019.8901885.
[52] Stordis, "Open Networking." https://stordis.com/open-networking/.
[53] Cisco, "Software-Defined Networking." https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html.
[54] HP, "Software Defined Networking." https://techlibrary.hpe.com/ie/en/networking/solutions/technology/sdn/index.aspx#.Y05G7nZBwuU.
[55] G. Lakhani and A. Kothari, *Fault Administration by Load Balancing in Distributed SDN Controller: A Review*, vol. 114, no. 4. Springer US, 2020.
[56] H. Mokhtar, X. Di, Y. Zhou, A. Hassan, Z. Ma, and S. Musa, "Multiple-level threshold load balancing in distributed SDN controllers," *Comput. Networks*, vol. 198, no. July, p. 108369, 2021, doi: 10.1016/j.comnet.2021.108369.
[57] E. Amiri, E. Alizadeh, and M. H. Rezvani, "Controller selection in software defined networks using best-worst multi-criteria decision-making," *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1506–1517, 2020, doi: 10.11591/eei.v9i4.2393.
[58] S. K. Keshari, V. Kansal, and S. Kumar, "A Systematic Review of Quality of Services (QoS) in Software Defined Networking (SDN)," *Wirel. Pers. Commun.*, vol. 116, no. 3, pp. 2593–2614, 2021, doi: 10.1007/s11277-020-07812-2.
[59] M. T. Islam, N. Islam, and M. Al Refat, "Node to Node Performance Evaluation through RYU SDN Controller," *Wirel. Pers. Commun.*, vol. 112, no. 1, pp. 555–570, 2020, doi: 10.1007/s11277-020-07060-4.
[60] D. Lunagariya and B. Goswami, "A comparative performance analysis of stellar SDN controllers using emulators," *Proc. 2021 1st Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2021*, 2021, doi: 10.1109/ICAECT49130.2021.9392391.
[61] B. Almadani, A. Beg, and A. Mahmoud, "DSF: A Distributed SDN Control Plane Framework for the East/West Interface," *IEEE Access*, vol. 9, pp. 26735–26754, 2021, doi: 10.1109/ACCESS.2021.3057690.
[62] Z. Latif, K. Sharif, F. Li, M. M. Karim, S. Biswas, and Y. Wang, "A comprehensive survey of interface protocols for software defined networks," *J. Netw. Comput. Appl.*, vol. 156, no. January, p. 102563, 2020, doi: 10.1016/j.jnca.2020.102563.
[63] A. Shirmarz and A. Ghaffari, *Performance issues and solutions in SDN-based data center: a survey*, vol. 76, no. 10. Springer US, 2020.
[64] K. Nisar *et al.*, "A survey on the architecture, application, and security of software defined networking: Challenges and open issues," *Internet of Things (Netherlands)*, vol. 12, p. 100289, 2020, doi: 10.1016/j.iot.2020.100289.
[65] S. Chen *et al.*, "Internet of Things Based Smart Grids Supported by Intelligent Edge Computing," *IEEE Access*, vol. 7, pp. 74089–74102, 2019, doi: 10.1109/ACCESS.2019.2920488.
[66] M. Aazam, S. Zeadally, and E. F. Flushing, "Task offloading in edge computing for machine learning-based smart healthcare," *Comput. Networks*, vol. 191, no. March, p. 108019, 2021, doi: 10.1016/j.comnet.2021.108019.
[67] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, 2020, doi: 10.1109/JIOT.2020.2987070.

[68] N. Chen and Y. Chen, "Anomalous Vehicle Recognition in Smart Urban Traffic Monitoring as an Edge Service†," *Futur. Internet*, vol. 14, no. 2, 2022, doi: 10.3390/fi14020054.
[69] J. Hwang, L. Nkenyereye, N. Sung, J. Kim, and J. Song, "IoT Service Slicing and Task Offloading for Edge Computing," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11526–11547, 2021, doi: 10.1109/JIOT.2021.3052498.
[70] M. Bukhsh, S. Abdullah, and I. S. Bajwa, "A Decentralized Edge Computing Latency-Aware Task Management Method with High Availability for IoT Applications," *IEEE Access*, vol. 9, pp. 138994–139008, 2021, doi: 10.1109/ACCESS.2021.3116717.
[71] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent Offloading for Collaborative Smart City Services in Edge Computing," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7919–7927, 2020, doi: 10.1109/JIOT.2020.3000871.
[72] H. Mokhtar, X. Di, Y. Zhou, A. Hassan, Z. Ma, and S. Musa, "Multiple-level threshold load balancing in distributed SDN controllers," *Comput. Networks*, vol. 198, no. July, p. 108369, 2021, doi: 10.1016/j.comnet.2021.108369.
[73] S. Sawalha and G. Al-Naymat, "Towards an efficient big data management schema for IoT," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 9, pp. 7803–7818, 2022, doi: 10.1016/j.jksuci.2021.09.013.
[74] R. Chai, X. Yang, C. Du, and Q. Chen, "Network cost optimization-based capacitated controller deployment for SDN," *Comput. Networks*, vol. 197, p. 108326, 2021, doi: 10.1016/j.comnet.2021.108326.
[75] M. Rahouti, K. Xiong, and Y. Xin, "Secure Software-Defined Networking Communication Systems for Smart Cities: Current Status, Challenges, and Trends," *IEEE Access*, vol. 9, pp. 12083–12113, 2021, doi: 10.1109/ACCESS.2020.3047996.
[76] S. Hameed *et al.*, "A Scalable Key and Trust Management Solution for IoT Sensors Using SDN and Blockchain Technology," *IEEE Sens. J.*, vol. 21, no. 6, pp. 8716–8733, 2021, doi: 10.1109/JSEN.2021.3052009.
[77] K. S. Sahoo *et al.*, "ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5852–5860, 2020, doi: 10.1109/JIOT.2019.2952527.
[78] S. Chattopadhyay, S. Chatterjee, S. Nandi, and S. Chakraborty, "Aloe: Fault-Tolerant Network Management and Orchestration Framework for IoT Applications," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 4, pp. 2396–2409, 2020, doi: 10.1109/TNSM.2020.3008426.
[79] R. Chai, X. Yang, C. Du, and Q. Chen, "Network cost optimization-based capacitated controller deployment for SDN," *Comput. Networks*, vol. 197, p. 108326, 2021, doi: 10.1016/j.comnet.2021.108326.
[80] D. Espinel Sarmiento, A. Lebre, L. Nussbaum, and A. Chari, "Decentralized SDN Control Plane for a Distributed Cloud-Edge Infrastructure: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 23, no. 1, pp. 256–281, 2021, doi: 10.1109/COMST.2021.3050297.
[81] A. Xiong *et al.*, "A Distributed Security SDN Cluster Architecture for Smart Grid Based on Blockchain Technology," *Secur. Commun. Networks*, vol. 2021, 2021, doi: 10.1155/2021/9495093.
[82] H. Ning, Y. Li, F. Shi, and L. T. Yang, "Heterogeneous edge computing open platforms and tools for internet of things," *Futur. Gener. Comput. Syst.*, vol. 106, pp. 67–76, 2020, doi: 10.1016/j.future.2019.12.036.
[83] A. Montazerolghaem, "Software-defined load-balanced data center: design, implementation and performance analysis," *Cluster Comput.*, vol. 24, no. 2, pp. 591–610, 2021, doi: 10.1007/s10586-020-03134-x.
[84] A. Abouaomar, S. Cherkaoui, Z. Mlika, and A. Kobbane, "Resource Provisioning in Edge Computing for Latency-Sensitive Applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11088–11099, 2021, doi: 10.1109/JIOT.2021.3052082.
[85] A. Naouri, H. Wu, N. A. Nouri, S. Dhelim, and H. Ning, "A Novel Framework for Mobile-Edge Computing by Optimizing Task Offloading," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 13065–13076, 2021, doi: 10.1109/JIOT.2021.3064225.
[86] A. Montazerolghaem, "Software-defined load-balanced data center: design, implementation and performance analysis," *Cluster Comput.*, vol. 24, no. 2, pp. 591–610, 2021, doi: 10.1007/s10586-020-03134-x.
[87] M. Nasir, K. Muhammad, A. Ullah, J. Ahmad, S. Wook Baik, and M. Sajjad, "Enabling automation and edge intelligence over resource constraint IoT devices for smart home," *Neurocomputing*, vol. 491, pp. 494–506, 2022, doi: 10.1016/j.neucom.2021.04.138.
[88] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, "Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 51, no. 1, pp. 223–236, 2021, doi: 10.1109/TSMC.2020.3042898.
[89] D. Abdelqawy, A. El-Korany, A. Kamel, and S. Makady, "Hub-OS: An interoperable IoT computing platform for resources utilization with real-time support," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1498–1510, 2022, doi: 10.1016/j.jksuci.2022.02.011.
[90] C. Campolo, G. Genovese, G. Singh, and A. Molinaro, "Scalable and interoperable edge-based federated learning in IoT contexts," *Comput. Networks*, vol. 223, no. October 2022, p. 109576, 2023, doi: 10.1016/j.comnet.2023.109576.

[91] A. Amjad, F. Azam, M. W. Anwar, and W. H. Butt, "A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT," *IEEE Access*, vol. 9, pp. 96528–96545, 2021, doi: 10.1109/ACCESS.2021.3094763.

[92] D. P. Isravel, S. Silas, and E. B. Rajsingh, "Sdn-based traffic management for personalized ambient assisted living healthcare system," in *Advances in Intelligent Systems and Computing*, 2021, vol. 1167, pp. 379–388, doi: 10.1007/978-981-15-5285-4_38.

[93] A. Rizwan, D. A. Karras, J. Kumar, M. Sánchez-Chero, M. M. Mogollón Taboada, and G. C. Altamirano, "An Internet of Things (IoT) Based Block Chain Technology to Enhance the Quality of Supply Chain Management (SCM)," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/9679050.

[94] Y. Hu, Y. Li, M. C. Gursoy, S. Velipasalar, and A. Schmeink, "Throughput Analysis of Low-Latency IoT Systems with QoS Constraints and Finite Blocklength Codes," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3093–3104, 2020, doi: 10.1109/TVT.2020.2968463.

[95] D. Wu *et al.*, "Software-Defined Edge Computing: A New Architecture Paradigm to Support IoT Data Analysis," pp. 1–7, 2021, [Online]. Available: http://arxiv.org/abs/2104.11645.

[96] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, and A. Hussain, "Big data and IoT-based applications in smart environments: A systematic review," *Comput. Sci. Rev.*, vol. 39, p. 100318, 2021, doi: 10.1016/j.cosrev.2020.100318.

[97] X. Xu *et al.*, "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Futur. Gener. Comput. Syst.*, vol. 95, pp. 522–533, 2019, doi: 10.1016/j.future.2018.12.055.

[98] A. Cheema, M. Tariq, A. Hafiz, M. M. Khan, F. Ahmad, and M. Anwar, "Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review," *Secur. Commun. Networks*, vol. 2022, 2022, doi: 10.1155/2022/8379532.

[99] H. Mrabet, S. Belguith, A. Alhomoud, and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors (Switzerland)*, vol. 20, no. 13, pp. 1–20, 2020, doi: 10.3390/s20133625.

[100] M. Mamdouh, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, "Authentication and Identity Management of IoHT Devices: Achievements, Challenges, and Future Directions," *Comput. Secur.*, vol. 111, p. 102491, 2021, doi: 10.1016/j.cose.2021.102491.

[101] S. Szymoniak and S. Kesar, "Key Agreement and Authentication Protocols in the Internet of Things: A Survey," *Appl. Sci.*, vol. 13, no. 1, 2023, doi: 10.3390/app13010404.

[102] A. Ali, A. Mateen, A. Hanan, and F. Amin, "Advanced Security Framework for Internet of Things (IoT)," *Technologies*, vol. 10, no. 3, p. 60, 2022, doi: 10.3390/technologies10030060.

[103] A. Khan, A. Ahmad, M. Ahmed, J. Sessa, and M. Anisetti, "Authorization schemes for internet of things: requirements, weaknesses, future challenges and trends," *Complex Intell. Syst.*, vol. 8, no. 5, pp. 3919–3941, 2022, doi: 10.1007/s40747-022-00765-y.

[104] H. Wang, D. He, J. Yu, N. N. Xiong, and B. Wu, "RDIC: A blockchain-based remote data integrity checking scheme for IoT in 5G networks," *J. Parallel Distrib. Comput.*, vol. 152, pp. 1–10, 2021, doi: 10.1016/j.jpdc.2021.02.012.

[105] C. Perera, M. Barhamgi, A. K. Bandara, M. Ajmal, B. Price, and B. Nuseibeh, "Designing privacy-aware internet of things applications," *Inf. Sci. (Ny).*, vol. 512, pp. 238–257, 2020, doi: 10.1016/j.ins.2019.09.061.

[106] F. Michelinakis, A. S. Al-Selwi, M. Capuzzo, A. Zanella, K. Mahmood, and A. Elmokashfi, "Dissecting Energy Consumption of NB-IoT Devices Empirically," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1224–1242, 2021, doi: 10.1109/JIOT.2020.3013949.

[107] A. N. Abosaif and H. S. Hamza, "Quality of service-aware service selection algorithms for the internet of things environment: A review paper," *Array*, vol. 8, no. April, p. 100041, 2020, doi: 10.1016/j.array.2020.100041.

[108] H. Gao, W. Huang, and Y. Duan, "The Cloud-edge-based Dynamic Reconfiguration to Service Workflow for Mobile Ecommerce Environments," *ACM Trans. Internet Technol.*, vol. 21, no. 1, pp. 1–23, Feb. 2021, doi: 10.1145/3391198.

[109] M. Aazam, S. Zeadally, and E. F. Flushing, "Task offloading in edge computing for machine learning-based smart healthcare," *Comput. Networks*, vol. 191, no. March, p. 108019, 2021, doi: 10.1016/j.comnet.2021.108019.

[110] M. K. Hussein and M. H. Mousa, "Efficient task offloading for IoT-Based applications in fog computing using ant colony optimization," *IEEE Access*, vol. 8, pp. 37191–37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[111] P. Bardalai, N. Medhi, B. Bargayary, and D. K. Saikia, "OpenHealthQ: OpenFlow based QoS management of Healthcare Data in a Software-Defined Fog environment," *IEEE Int. Conf. Commun.*, pp. 0–5, 2021, doi: 10.1109/ICC42927.2021.9500637.

[112] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A Smart, Efficient, and Reliable Parking Surveillance System with Edge Artificial Intelligence on IoT Devices," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4962–4974, 2021, doi: 10.1109/TITS.2020.2984197.

[113] V. K. Rathi *et al.*, "An edge AI-enabled IoT healthcare monitoring system for smart cities," *Comput. Electr.*

Eng., vol. 96, no. PB, p. 107524, 2021, doi: 10.1016/j.compeleceng.2021.107524.
[114] C. Zhang, "Design and application of fog computing and Internet of Things service platform for smart city," *Futur. Gener. Comput. Syst.*, vol. 112, pp. 630–640, 2020, doi: 10.1016/j.future.2020.06.016.
[115] A. I. Taloba *et al.*, "A blockchain-based hybrid platform for multimedia data processing in IoT-Healthcare," *Alexandria Eng. J.*, vol. 62, 2022, doi: 10.1016/j.aej.2022.09.031.
[116] N. Hudson, M. J. Hossain, M. Hosseinzadeh, H. Khamfroush, M. Rahnamay-Naeini, and N. Ghani, "A Framework for Edge Intelligent Smart Distribution Grids via Federated Learning," *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, vol. 2021-July, 2021, doi: 10.1109/ICCCN52240.2021.9522360.
[117] L. El-Garoui, S. Pierre, and S. Chamberland, "A new sdn-based routing protocol for improving delay in smart city environments," *Smart Cities*, vol. 3, no. 3, pp. 1004–1021, 2020, doi: 10.3390/smartcities3030050.
[118] N. Kumar, S. Mittal, V. Garg, and N. Kumar, "Deep Reinforcement Learning-Based Traffic Light Scheduling Framework for SDN-Enabled Smart Transportation System," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2411–2421, 2022, doi: 10.1109/TITS.2021.3095161.
[119] A. H. M. Jakaria, M. A. Rahman, and A. Gokhale, "Resiliency-Aware Deployment of SDN in Smart Grid SCADA: A Formal Synthesis Model," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1430–1444, 2021, doi: 10.1109/TNSM.2021.3050148.
[120] B. Preveze, A. Alkhayyat, F. Abedi, A. M. Jawad, and A. S. Abosinnee, "SDN-Driven Internet of Health Things: A Novel Adaptive Switching Technique for Hospital Healthcare Monitoring System," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/3150756.
[121] M. Y. Mehmood *et al.*, "Demand-response management using a fleet of electric vehicles: An opportunistic-SDN-based edge-cloud framework for smart grids," *Comput. Electr. Eng.*, vol. 2021, no. 4, pp. 77637–77648, 2021, doi: 10.1016/j.compeleceng.2020.106634.
[122] E. Barka, S. Dahmane, C. A. Kerrache, M. Khayat, and F. Sallabi, "Sthm: A secured and trusted healthcare monitoring architecture using sdn and blockchain," *Electron.*, vol. 10, no. 15, pp. 1–15, 2021, doi: 10.3390/electronics10151787.
[123] C. Lin *et al.*, "Mobile-edge computing-based delay minimization controller placement in SDN-IoV," *Comput. Networks*, vol. 27, no. 4, pp. 3832–3840, 2021, doi: 10.1109/TGCN.2022.3162237.
[124] F. S. Dantas Silva, E. Silva, E. P. Neto, M. Lemos, A. J. Venancio Neto, and F. Esposito, "A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios," *Sensors (Switzerland)*, vol. 20, no. 11, pp. 1–28, 2020, doi: 10.3390/s20113078.
[125] A. I. Taloba *et al.*, "A blockchain-based hybrid platform for multimedia data processing in IoT-Healthcare," *Alexandria Eng. J.*, vol. 62, 2022, doi: 10.1016/j.aej.2022.09.031.
[126] H. Babbar, S. Rani, A. K. Bashir, and R. Nawaz, "LBSMT: Load Balancing Switch Migration Algorithm for Cooperative Communication Intelligent Transportation Systems," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 3, pp. 1386–1395, 2022, doi: 10.1109/TGCN.2022.3162237.
[127] A. Das, S. Chakraborty, and S. Chakraborty, "Where do all my smart home data go? Context-aware data generation and forwarding for edge-based microservices over shared IoT infrastructure," *Futur. Gener. Comput. Syst.*, vol. 134, pp. 204–218, 2022, doi: 10.1016/j.future.2022.03.027.
[128] Y. Guo, Z. Zhang, and Y. Guo, "SecFHome: Secure remote authentication in fog-enabled smart home environment," *Comput. Networks*, vol. 207, no. September 2021, p. 108818, 2022, doi: 10.1016/j.comnet.2022.108818.
[129] A. Nazari, F. Tavassolian, M. Abbasi, R. Mohammadi, and P. Yaryab, "An Intelligent SDN-Based Clustering Approach for Optimizing IoT Power Consumption in Smart Homes," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/8783380.
[130] S. S. S. Homes *et al.*, "Enabling automation and edge intelligence over resource constraint IoT devices for smart home," *Comput. Networks*, vol. 2022, no. September 2021, pp. 1–23, 2022, doi: 10.1016/j.comnet.2022.108818.
[131] Y. Li, X. He, and Y. Bian, "Task Offloading of Edge Computing Network and Energy Saving of Passive House for Smart City," *Mob. Inf. Syst.*, vol. 2022, 2022, doi: 10.1155/2022/4832240.
[132] Y. Su, P. Jiang, H. Chen, and X. Deng, "A QoS-Guaranteed and Congestion-Controlled SDN Routing Strategy for Smart Grid," *Appl. Sci.*, vol. 12, no. 15, 2022, doi: 10.3390/app12157629.
[133] Z. Eghbali and M. Z. Lighvan, "A hierarchical approach for accelerating IoT data management process based on SDN principles," *J. Netw. Comput. Appl.*, vol. 181, no. October 2020, p. 103027, 2021, doi: 10.1016/j.jnca.2021.103027.
[134] R. Saha, N. Ahmed, and S. Misra, "SDN-Controller Triggered Dynamic Decision Control Mechanism for Healthcare IoT," *2021 IEEE Glob. Commun. Conf. GLOBECOM 2021 - Proc.*, 2021, doi: 10.1109/GLOBECOM46510.2021.9685911.
[135] J. Islam, T. Kumar, I. Kovacevic, and E. Harjula, "Resource-aware Dynamic Service Deployment for Local IoT Edge Computing: Healthcare Use Case," *IEEE Access*, vol. 9, pp. 115868–115884, 2021, doi: 10.1109/ACCESS.2021.3102867.

[136] J. Ren, J. Li, H. Liu, and T. Qin, "Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT," *Tsinghua Sci. Technol.*, vol. 27, no. 4, pp. 760–776, 2022, doi: 10.26599/TST.2021.9010046.

[137] M. Peyman, P. J. Copado, R. D. Tordecilla, L. D. C. Martins, F. Xhafa, and A. A. Juan, "Edge computing and iot analytics for agile optimization in intelligent transportation systems," *Energies*, vol. 14, no. 19, pp. 1–26, 2021, doi: 10.3390/en14196309.

[138] Y. Yang, "A SDN-based traffic estimation approach in the internet of vehicles," *Wirel. Networks*, vol. 1, 2021, doi: 10.1007/s11276-021-02668-1.

[139] B. Cao, Z. Sun, J. Zhang, and Y. Gu, "Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3832–3840, 2021, doi: 10.1109/TITS.2020.3048844.

[140] B. Li, X. Deng, and Y. Deng, "Mobile-edge computing-based delay minimization controller placement in SDN-IoV," *Comput. Networks*, vol. 193, no. January, 2021, doi: 10.1016/j.comnet.2021.108049.

[141] M. Nasir, K. Muhammad, A. Ullah, J. Ahmad, S. Wook Baik, and M. Sajjad, "Enabling automation and edge intelligence over resource constraint IoT devices for smart home," *Neurocomputing*, vol. 491, pp. 494–506, 2022, doi: 10.1016/j.neucom.2021.04.138.

[142] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati, "Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm," *Sensors*, vol. 21, no. 14, p. 4932, Jul. 2021, doi: 10.3390/s21144932.

[143] W. Iqbal, H. Abbas, P. Deng, and J. Wan, "ALAM : Anonymous Lightweight Authentication," vol. 8, no. 12, pp. 9622–9633, 2021.

[144] H. Gordon, C. Batula, B. Tushir, B. Dezfouli, and Y. Liu, "Securing smart homes via software-defined networking and low-cost traffic classification," *Proc. - 2021 IEEE 45th Annu. Comput. Software, Appl. Conf. COMPSAC 2021*, pp. 1049–1057, 2021, doi: 10.1109/COMPSAC51774.2021.00143.

[145] P. R. Grammatikis *et al.*, "SDN-Based Resilient Smart Grid: The SDN-microSENSE Architecture," *Digital*, vol. 1, no. 4, pp. 173–187, 2021, doi: 10.3390/digital1040013.

[146] L. Xi, Y. Wang, Y. Wang, Z. Wang, X. Wang, and Y. Chen, "Deep Reinforcement Learning-Based Service-Oriented Resource Allocation in Smart Grids," *IEEE Access*, vol. 9, pp. 77637–77648, 2021, doi: 10.1109/ACCESS.2021.3082259.

[147] M. Gheisari, G. Wang, and S. Chen, "An Edge Computing-enhanced Internet of Things Framework for Privacy-preserving in Smart City," *Comput. Electr. Eng.*, vol. 81, p. 106504, 2020, doi: 10.1016/j.compeleceng.2019.106504.

[148] A. M. Alsmadi *et al.*, "An Efficient Counter-Based DDoS Attack Detection Framework Leveraging Software Defined IoT (SD-IoT)," *Futur. Gener. Comput. Syst.*, vol. 33, no. 2, pp. 243–253, 2020, doi: 10.1016/j.future.2018.05.054.

[149] G. Tripathi, M. A. Ahad, and S. Paiva, "Sms: A secure healthcare model for smart cities," *Electron.*, vol. 9, no. 7, pp. 1–18, 2020, doi: 10.3390/electronics9071135.

[150] S. Misra, R. Saha, and N. Ahmed, "Health-Flow: Criticality-Aware Flow Control for SDN-Based Healthcare IoT," *2020 IEEE Glob. Commun. Conf. GLOBECOM 2020 - Proc.*, vol. 2020-Janua, 2020, doi: 10.1109/GLOBECOM42002.2020.9348058.

[151] C. Lin, G. Han, J. Du, T. Xu, L. Shu, and Z. Lv, "Spatiotemporal Congestion-Aware Path Planning Toward Intelligent Transportation Systems in Software-Defined Smart City IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8012–8024, 2020, doi: 10.1109/JIOT.2020.2994963.

[152] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A Distributed Mobile Fog Computing Scheme for Mobile Delay-Sensitive Applications in SDN-Enabled Vehicular Networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5481–5493, 2020, doi: 10.1109/TVT.2020.2980934.

[153] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent Task Offloading in Vehicular Edge Computing Networks," *IEEE Wirel. Commun.*, vol. 27, no. 4, pp. 126–132, 2020, doi: 10.1109/MWC.001.1900489.

[154] K. N. Qureshi, R. Hussain, and G. Jeon, "A distributed software defined networking model to improve the scalability and quality of services for flexible green energy internet for smart grid systems," *Comput. Electr. Eng.*, vol. 84, p. 106634, 2020, doi: 10.1016/j.compeleceng.2020.106634.

[155] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent Edge Computing for IoT-Based Energy Management in Smart Cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, 2019, doi: 10.1109/MNET.2019.1800254.

[156] A. Rego, L. Garcia, S. Sendra, and J. Lloret, "Software Defined Network-based control system for an efficient traffic management for emergency situations in smart cities," *Futur. Gener. Comput. Syst.*, vol. 88, pp. 243–253, 2018, doi: 10.1016/j.future.2018.05.054.

[157] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati, "Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm," *Sensors*, vol. 21, no. 14, p. 4932, Jul. 2021, doi: 10.3390/s21144932.

[158] IDC, "The Growth in Connected IoT Devices Is Expected to Generate 79.4ZB of Data in 2025, According to a New IDC Forecast," *Idc*, 2019. https://www.idc.com/getdoc.jsp?containerId=prUS45213219.