Edge-computing-driven Internet of Things: A Survey

LINGHE KONG, JINLIN TAN, JUNQIN HUANG, GUIHAI CHEN, and SHUAITIAN WANG, Shanghai Jiao Tong University, China
XI JIN and PENG ZENG, Shenyang Institute of Automation, Chinese Academy of Sciences, China
MUHAMMAD KHAN, King Saud University, Kingdom of Saudi Arabia
SAJAL K. DAS, Missouri University of Science and Technology, USA

The Internet of Things (IoT) is impacting the world's connectivity landscape. More and more IoT devices are connected, bringing many benefits to our daily lives. However, the influx of IoT devices poses non-trivial challenges for the existing cloud-based computing paradigm. In the cloud-based architecture, a large amount of IoT data is transferred to the cloud for data management, analysis, and decision making. It could not only cause a heavy workload on the cloud but also result in unacceptable network latency, ultimately undermining the benefits of cloud-based computing. To address these challenges, researchers are looking for new computing models for the IoT. Edge computing, a new decentralized computing model, is valued by more and more researchers in academia and industry. The main idea of edge computing is placing data processing in near-edge devices instead of remote cloud servers. It is promising to build more scalable, low-latency IoT systems. Many studies have been proposed on edge computing and IoT, but a comprehensive survey of this crossover area is still lacking.

In this survey, we first introduce the impact of edge computing on the development of IoT and point out why edge computing is more suitable for IoT than other computing paradigms. Then, we analyze the necessity of systematical investigation on the edge-computing-driven IoT (ECDriven-IoT) and summarize new challenges occurring in ECDriven-IoT. We categorize recent advances from bottom to top, covering six aspects of ECDriven-IoT. Finally, we conclude lessons learned and propose some challenging

CCS Concepts: • **Computer systems organization** → **Distributed architectures**; *Real-time operating systems*; • **Networks** → *Network protocols*; • **Security and privacy**; • **Computing methodologies** → *Distributed computing methodologies*;

Additional Key Words and Phrases: Edge computing, Internet of Things

This work was supported in part by NSFC Grants No. 62141220, No. 61972253, No. U1908212, No. 72061127001, No. 62172276, and No. 61972254, the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and Open Research Projects of Zhejiang Lab No. 2022NL0AB01.
Authors' addresses: L. Kong (corresponding author), J. Tan, J. Huang, G. Chen, and S. Wang, Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, 200240; emails: {linghe.kong, jinlintan, junqin.huang}@sjtu.edu.cn, gchen@cs.sjtu.edu.cn, wang-st@sjtu.edu.cn; X. Jin and P. Zeng, Shenyang Institute of Automation, Chinese Academy of Sciences, No. 135, Chuangxin Road, Hunnan District, Shenyang City, Liaoning Province, 110169; emails: {jinxi, zp}@sia.cn; M. Khan, King Saud University, Kingdom of Saudi Arabia, Po Box 92144, Riyadh 11653; email: mkhurram@ksu.edu.sa; S. K. Das, Missouri University of Science and Technology, Rolla, MO 65409, USA; email: sdas@mst.edu.

0360-0300/2022/12-ART174 $15.00
https://doi.org/10.1145/3555308

ACM Reference format:
Linghe Kong, Jinlin Tan, Junqin Huang, Guihai Chen, Shuaitian Wang, Xi Jin, Peng Zeng, Muhammad Khan, and Sajal K. Das. 2022. Edge-computing-driven Internet of Things: A Survey. *ACM Comput. Surv.* 55, 8, Article 174 (December 2022), 41 pages.
https://doi.org/10.1145/3555308

1 INTRODUCTION

The **Internet of Things (IoT)** is a revolutionary approach that interlinks uniquely addressable physical and virtual devices through different communication protocols. According to the statistics, the number of wireless-connected IoT devices will reach 50 billion by 2025 [51]. Potential devices include smartphones, bio-nano things, body sensors, smart tags, wearable devices, embedded objects, and traditional electronic gadgets [6]. These devices usually have a variety of sensors inside for collecting environmental data, which are fundamental elements of data-driven intelligence. Thus, massive deployed devices lead to an explosive data increase in the meantime. The collected data need to be processed and analyzed before providing useful results for users. But the computation ability of lightweight IoT devices is quite limited. One solution to this problem is cloud computing. In the cloud-based paradigm, IoT data is first transferred to the cloud server for processing, and then computing results will be sent back to devices. However, data transmission rate and network bandwidth could become bottlenecks to the further development of massive IoT [162]. Moreover, as most IoT devices will generate personal and sensitive data, it is inappropriate to send all IoT data to remote cloud servers, which will cause security and privacy concerns.

Edge computing is a new computing paradigm that directs computational data, applications, and services away from cloud servers to the network edge. Content providers and application developers can use edge computing to offer users services closer to them in geography, which can accelerate the response speed of services. Edge computing is characterized as high bandwidth, ultra-low latency, and real-time access to network information [86, 150]. And IoT applications usually require real-time response, privacy preservation, and massive data transmission. Compared with cloud computing, edge computing has the potential to match large-scale IoT applications' requirements.

The common goal of IoT and edge computing is to perform seamless computing anytime and anywhere, but they act in different roles in the system. IoT focuses on endpoint sensing, while edge computing focuses on near-field computation. Thus, it is promising for **edge-computing-driven IoT (ECDriven-IoT)** systems to make these two technologies complement each other. Nowadays, IoT has been widely used in many complex scenarios, such as smart homes, smart cities, smart grids, **virtual reality (VR)**, augmented reality (AR), and autonomous driving. ECDriven-IoT can benefit these IoT applications from three aspects: **(1) Real-time response and high quality of services (QoS).** Edge computing can provide shorter network latency than cloud computing, as edge servers lie closer to IoT devices in geography. This superiority can support high-demand real-time IoT applications better. Owing to a majority of data processed in edge servers, the amount of data offloaded to the cloud can be largely reduced. Thus, ECDriven-IoT can bring higher QoS for those real-time IoT applications. **(2) Low energy consumption.** Most IoT nodes are power-limited devices, but synchronizing large amounts of sensing data to the remote cloud wastes much power. With edge computing, IoT nodes only need to send data to local edge servers, so the energy consumption of IoT nodes can be decreased to a lower level. Thus, ECDriven-IoT can extend the lifetime of IoT nodes and reduce the maintenance overhead. **(3) High scalability.** One unavoidable challenge in cloud-based IoT systems is the large-scale access requirements. The cloud server could be the system bottleneck due to large amounts of concurrent connections from IoT nodes.

In ECDriven-IoT, edge servers (e.g., base stations) provide moderate computing resources in a distributed manner, so ECDriven-IoT can provide good scalability that satisfies the requirements of large-scale IoT applications like smart cities or autonomous driving. Therefore, we believe edge computing is indispensable for future IoT, and the study combining IoT and edge computing has academic prospects.

Many surveys [57, 157] pointed out that recent advances related to IoT and edge computing have made many efforts to satisfy these requirements. However, when combining edge computing and IoT, there are still several new challenges regarding how to efficiently integrate these two technologies and bridge the difference between them. We summarize three new challenges in ECDriven-IoT systems:

- **Heterogeneity of edge computing and IoT.** IoT devices are working everywhere and vary across different scenarios. Thus, there are various hardware devices and communication protocols in IoT systems. For edge computing, the deployment architecture of edge nodes also requires different solutions for different scenarios. Thus, combining edge computing with IoT is faced with the challenge on how to unify the diversity of IoT and edge computing and make them complement each other. To efficiently apply edge computing in heterogeneous IoT systems, the cooperation architecture of ECDriven-IoT, hardware devices, and communication protocols need to be explored and form industry standards.
- **Coordination between communication and computing.** When combining edge computing with IoT, the system is more complex than only IoT or edge computing-based ones. The communication between edge servers and IoT devices will bring extra consumption to the whole system. Besides, edge servers and IoT devices are limited in power and computing capacity. For example, if IoT devices offload all workload to edge servers, then it must pose a greater demand on communication cost and computing capacity of edge nodes. So, we should explore how to allocate workload between edge servers and IoT devices to balance the cost of communication and computation.
- **Complicated security and privacy issues.** How to guarantee systems' security and privacy is always a significant challenge in IoT and edge computing. However, these issues become more tricky due to the heterogeneity and limited computing capability of ECDriven-IoT. IoT devices and edge servers are vulnerable to various attacks. Once any of these points are compromised, the system will be in great danger. So, a qualified ECDriven-IoT system should fully consider possible security threats and countermeasures in different application scenarios.

There have been numerous studies from IoT to edge computing, covering every aspect of the system. It is necessary to reveal what research has been done in this area and explore what the future research direction is. Many excellent surveys have focused on either IoT or edge computing. In the IoT aspect, a plethora of surveys have referred to architecture [57, 157], communication [106, 124], IoT application [61, 159] as well as security and privacy [10, 110]. As for edge computing combined with IoT, there have also been several surveys from different perspectives as shown in Table 1. However, these surveys:

- covered a limited number of research areas, and the system architecture of ECDriven-IoT has not been discussed.
- revealed many challenges in edge computing or IoT, but those new challenges arising from combining edge computing and IoT have not been explored.

ECDriven-IoT is a promising solution taking advantage of edge computing to build scalable and efficient IoT systems. Both academia and industry need a survey to explain what happens

Table 1. Comparison of Previous Surveys in IoT and Edge Computing

Survey	IoT	Edge Computing	IoT-Edge Architecture	Architecture	Operating System	Communication Protocol	Computing	Application	Challenge
Rafique et al. [142]	✓	✓	✓			✓	✓	✓	✓
J. Pan et al. [130]	✓	✓	✓			✓			✓
Abbas et al. [2]		✓					✓	✓	✓
Porambage et al. [138]	✓	✓	✓				✓	✓	
Javed et al. [80]	✓				✓	✓			✓
Mouradian et al. [118]		✓					✓	✓	✓
Salman et al. [152]	✓	✓	✓			✓	✓		✓
Roman et al. [146]		✓					✓		✓
Baktir et al. [17]		✓					✓	✓	✓
Y. Mao et al. [108]		✓				✓	✓		✓
W. Yu et al. [194]	✓	✓	✓				✓	✓	✓
Elazhary [49]	✓	✓				✓			✓
Y. Ai et al. [7]	✓	✓	✓				✓	✓	✓
Alwarafy et al. [12]	✓	✓	✓		✓	✓			
Jararweh et al. [78]	✓	✓	✓				✓		✓
Our Survey	✓	✓	✓	✓	✓	✓	✓	✓	✓

when edge computing encounters IoT, what benefits it brings in, and what new challenges it faces. Taking the requirements of a comprehensive survey into account, we illustrate the architecture of ECDriven-IoT from the different levels in detail and summarize recent research advances. The main contributions of this survey are summarized as follows:

- We reveal three new challenges in ECDriven-IoT, including the heterogeneity of IoT and edge computing, coordination between communication and computation, and more tricky security and privacy issues.
- We categorize existing related studies from six aspects, i.e., system architecture, operating system, communication protocol, computing paradigm, application, and security and privacy. It gives a whole view of the current advance of ECDriven-IoT and describes possible solutions for addressing these challenges in each aspect.
- Finally, we conclude key lessons learned after reviewing existing related work and give several insights into future research challenges and directions in ECDriven-IoT.

The rest of this survey is organized as follows. Section 2 introduces the background of IoT and edge computing and compares cloud computing with edge computing. The taxonomy of ECDriven-IoT is also proposed in this section. According to the taxonomy, various hardware architectures of ECDriven-IoT are explored in Section 3. In Section 4, we present operating systems adopted in IoT and edge computing, which play a significant role in ECDriven-IoT. Communication protocols and computing technologies are discussed in Sections 5 and 6, respectively. Section 7 discusses security and privacy concerns when deploying ECDriven-IoT systems in practice. ECDriven-IoT applications are introduced in Section 8. Section 9 provides lessons learned, open challenges along with future research directions. Finally, we make a conclusion in Section 10.

2 BACKGROUND

In this section, we introduce the background of ECDriven-IoT, which has two main components: IoT and edge computing. We also compare the difference between edge computing and cloud computing to explain why edge computing is more suitable for IoT than cloud computing. And then, we give a taxonomy of ECDriven-IoT.

2.1 Internet of Things

In the IoT world, all objects that exist in reality can connect to the Internet and be accessed by users. Through a specific addressing scheme, IoT devices can cooperate to complete the designated work [15]. The main advantage of IoT is its great impact on people's daily life and potential user behavior [134]. On the one hand, for individual users, the benefit is reflected in areas such as

electronic health, smart home, and life support. On the other hand, for industry, IoT also plays an active role in automation, logistics, and intelligent transportation.

IoT has attracted immense attention from the industry and academia [6, 185]. More and more IoT applications are focusing on achieving real-time responses, such as **Virtual Reality (VR)** [52], **Augmented Reality (AR)** [52], and automatic driving. The very short latency is non-negotiable for these applications. In cloud computing, due to geographical distance and network fluctuation, the latency is too high to satisfy real-time requirements. Besides, massive data deteriorate transmission performance. So, how to effectively allocate network bandwidth and computing resources is a challenge [189]. In the IoT community, there are different data formats and communication protocols, making IoT a vertically fragmented network system [190], which poses another challenge to accomplish the desired low-latency performance. Furthermore, most IoT devices are power-limited, and it is necessary to balance the power consumption to extend the lifetime of IoT devices.

Before IoT further deepens its impact on the world, many challenges are still worthy of attention and research. One of the critical issues is how to achieve good interconnection and interoperability among IoT devices, guarantee security demands, and provide a high level of intelligence. In addition, IoT devices usually lack of computing power and energy capacity. Therefore, a new computational paradigm should target the resource efficiency in addition to scalability issues. Edge computing as a new computing paradigm could provide such help for IoT.

2.2 Edge Computing

Edge computing essentially migrates partial computing jobs from remote cloud servers to local edge servers. It performs data preprocessing and analysis near the data sources. Since edge servers are closer to data-generated devices, they can have a quicker response than cloud servers. On the contrary, the advantage of cloud computing is providing global scheduling capability and powerful computing resources. Similar to edge computing, **fog computing (FC)** is a highly virtualized platform that offers computing resources, storage, and control between end-users and cloud servers, proposed by Cisco in 2012 [24]. In this survey, we refer to edge computing and FC collectively as edge computing.

2.2.1 Cloud Computing. Cloud computing is another significant change after large computer to client-server transformation. Users can share software and hardware resources in cloud computing [70]. The complex hardware structure in cloud systems is transparent to users. So, users do not need the expertise or direct control of cloud servers. There have been many studies on the cloud and IoT, namely, the CloudIoT paradigm [26]. They have thoroughly investigated the main attributes, characteristics, basic concepts, and open issues of the CloudIoT paradigm. Table 2 shows the connection and difference between cloud computing and edge computing. Edge computing is essentially an edge optimization of cloud computing. Both of them are designed for handling big data. However, the main difference is that data can be distributed and processed on the closer edge servers in edge computing. Figure 1 also shows the difference in geographic distribution between cloud computing and edge computing. Thus, edge computing is more suitable for real-time data processing and secure intelligent analysis.

Many studies attempt to optimize cloud computing to suit IoT scenarios [27, 44]. For example, Zhou et al. proposed an architecture named CloudThings, which is an approach to combine cloud computing and IoT. This architecture is a cloud-oriented IoT approach, helping IaaS, PaaS, and SaaS in developing and managing IoT applications [199]. Pacheco et al. proposed a privacy-protected architecture for integrating cloud computing and IoT. This architecture presents a scheme for protecting data generated by IoT devices without a secure transport layer protocol [34]. However, the requirement of real-time response, massive data throughput, and low power still constrain the

Table 2. Comparison of Edge Computing and Cloud Computing

	Edge Computing	Cloud Computing
Computing Location	Edge devices, Distributed core network	Centralized big data centers
Key Feature	Close to data source, Edge and core network	Centralization
Network Components	Terminal device, edge device and IoT gateways, Core network hardwares	All basic network components, Data centers
Flexibility and Scalability	High	Low
Size	Massive small nodes or according to demands	Large
Deployment	Temporary deployment or deployment with minimal planning	Complex deployment
Bandwidth Requirements	A little and well balanced	Long-haul network bandwidth requirement

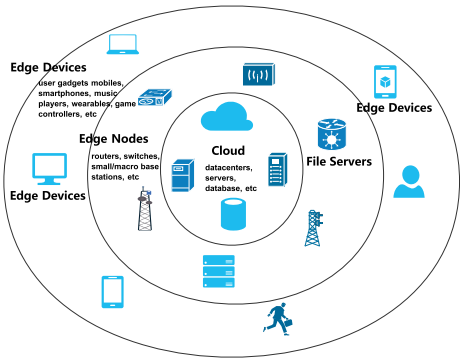


Fig. 1. Cloud, edge nodes, and edge devices.

application of cloud computing in IoT [144]. Thus, another solution is to propose a new computing paradigm to solve these problems thoroughly.

2.2.2 Edge Computing. Edge computing conforms to the computing characteristics of mobile devices in IoT. The core of this architecture is **mobile edge computing (MEC)**. MEC is a new concept that integrates IT and telecommunications, which adds functions such as computation, storage, and processing to the wireless network side. It enables more and more mobile devices to quickly and easily access IoT, such as wearable smart devices. Yaser et al. proposed a layered model consisting of a MEC server and a Cloudlets infrastructure [79]. This architecture aims to increase the coverage of mobile user signals. And it allows users to complete the services they request with minimal cost in terms of power and response latency. The main goal of the MEC solution is to export some cloud functions to the mobile network edge, increasing available bandwidth and reducing latency. Unlike the general architectural model, mobile hardware architecture is used more in communications, using multiple **software-defined network (SDN)** controllers and virtualization to solve data processing in communications [153].

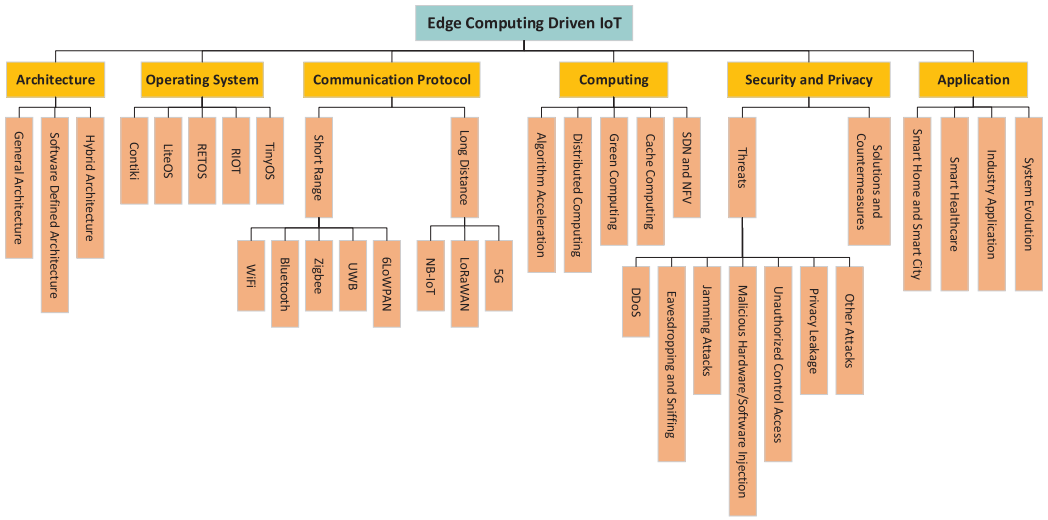


Fig. 2. Taxonomy of edge-computing-driven IoT.

2.3 Taxonomy of ECDriven-IoT

The taxonomy of the ECDriven-IoT is shown in Figure 2. We categorize relevant literature into six parts: hardware architecture, operating system, communication protocol, computing layer, security and privacy mechanism, and application. Existing research works are reviewed and grouped into the above six parts according to their research focuses. These parts are discussed from bottom to up according to the layer in ECDriven-IoT systems. Although these six parts belong to different research areas, they work together to form a complete ECDriven-IoT system.

The first layer is the lowest-level hardware architecture layer, which focuses on IoT and edge computing hardware. Research work includes general hardware architecture that typically contains terminal things and edge network devices, mobile architecture suitable for mobile IoT scenarios, and scalable hierarchical architecture. These three different architectures are categorized according to different deployment scenarios. The second layer, the operating system layer, mainly concludes several well-known IoT operating systems, such as Huawei LiteOS and mbedOS. Those IoT operating systems are widely used in practical applications. And they are generally lightweight, making them suitable for application to edge devices. The third layer is the study of communication protocols, including both short-range ones and long-distance ones. An important area of research in IoT is communication interaction between various devices. Especially in edge computing environments, more communication takes place among devices. These devices often use different communication protocols and require cross-protocol communication. Short-range communication protocols include famous Wi-Fi, Bluetooth, ZigBee, and 6LoPWAN [162]. Long distance communication protocols include NB-IoT, LoRaWAN, and 5G. The fourth layer is the computing layer, including computation offloading, IoT distributed computing, caching, **software-defined network (SDN)**, and **network function virtualization (NFV)**. These new computational studies are currently not given sufficient attention, but they all have large development prospects and can optimize edge computing in the IoT. The fifth layer is the security and privacy layer. It is necessary to consider security and privacy factors when designing secure ECDriven-IoT systems. We analyze possible threats and existing countermeasures in ECDriven-IoT. The final layer is the application layer. We introduce several popular ECDriven-IoT applications, including urban smart living, industrial applications, and optimization of the entire system.

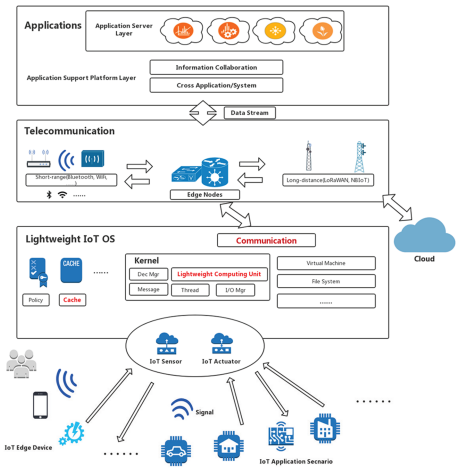


Fig. 3. Edge-computing-driven IoT model architecture.

3 ARCHITECTURE OF EDGE-COMPUTING-DRIVEN IOT

As a complement to cloud computing, the IoT system will become more complicated when meeting edge computing. Since the number of edge nodes is large and edge nodes distribute everywhere in geography, one research point is how to manage and maintain the ECDriven-IoT architecture. Figure 3 shows a typical ECDriven-IoT model architecture. The architecture of the ECDriven-IoT needs to: (1) Manage devices. There is a variety of IoT devices, and the number of IoT devices is relatively large. Thus, they should be managed efficiently to satisfy network bandwidth and power consumption requirements. (2) Allocate resources. Edge computing nodes lie near IoT devices, which can significantly reduce communication latency. However, the processing capabilities of edge nodes are limited. The latency will be very high if application tasks are waiting for the node. Thus, the architecture should efficiently allocate computing resources to IoT devices. (3) Discover services. For edge nodes, they need to discover services and allocate computing and storage resources. Thus, how to efficiently discover services at a low cost is another problem that needs to consider in architecture. (4) Schedule power. IoT nodes are often power-limited, so the architecture should be energy efficient to reduce power consumption.

3.1 General Hardware Architecture

Unlike cloud computing, edge computing complements and extends cloud computing to edges and endpoints. Edge computing benefits from edge devices' proximity to sensors while leveraging the on-demand scalability of cloud resources [42]. The distributed infrastructure of the ECDriven-IoT contains heterogeneous resources and manages the architecture in a distributed manner. There are various participants in this distributed architecture, including data centers, network cores, network edges, and endpoints. Thus, the architecture should be properly designed. Figure 4 shows a typical three-layer architecture of the ECDriven-IoT.

IoT brings not only new entry points for big data analytics but also distributed data sources at the network edge. Bonomi et al. introduced a general hardware architecture that meets the needs of most IoT scenarios [23]. The general hardware architecture is characterized by a low-cost configuration that is easy to maintain and meets the needs of IoT architecture in general, resulting in significant benefits. Dautov et al. introduced a distributed hierarchical data fusion architecture for IoT networks, consisting of edge devices, a network, communications units, and cloud platforms

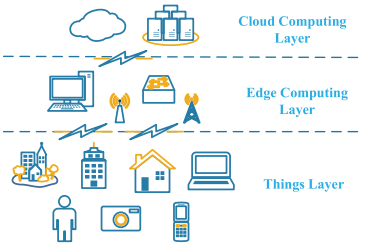


Fig. 4. The three-layers architecture of ECDriven-IoT.

together [43]. Different data sources are combined at each level of the IoT hierarchy to produce timely and accurate results by utilizing the computational capabilities of intermediate nodes.

EdgeX Foundry is another typical ECDriven-IoT architecture model [47], which creates an open-source framework for IoT edge computing. The framework is completely independent of hardware and operating systems, supports a plug-and-play component ecosystem, unifies the market, and accelerates the deployment of IoT solutions. Also, it addresses critical interoperability challenges for edge nodes and data normalization in a distributed IoT edge architecture [56].

Edge computing nodes deployed in various environments are heterogeneous, while general architecture can implement cross-platform management of heterogeneous resources. In the architecture, how to efficiently manage fog/edge computing infrastructure, allocate available resources to IoT devices, and schedule fog/edge computing resources is of significant importance [100]. Li et al. proposed an architecture called ECIoT and studied the management of radio resources and computing resources in ECIoT [98]. ECIoT focuses on resource allocation and performance control. To improve the performance of ECIoT, they use the Lyapunov stochastic optimization method to maximize system efficiency. Kitagami et al. proposed a multi-agent-based flexible IoT edge computing architecture to balance global optimization by a cloud and local optimization by edges for optimizing the role of cloud servers and edge servers dynamically [88]. In Reference [99], a multi-layer resource allocation scheme was proposed, and it can adapt to the characteristics of resource-constrained nodes at edges.

3.2 Software Defined Hardware Architecture

With the increasing demands of users and broader network access, IoT applications, network developers, service providers, and network carriers have to provide up-to-date services to users. In the same way, communication networks are expanding exponentially, leading to the whole system consisting of many subnets. These subnets have different communication and routing protocols. Integrating these heterogeneous subnets into a unified communications platform is a critical technical challenge, especially in a dynamic environment. For the practical envisioning of edge computing in IoT, there is a need for a simplified architecture that hides all the complexities of communication and provides a simple implementation.

According to the definition, SDN refers to a network architecture where the forwarding state in the data plane is managed by a remotely controlled plane decoupled from the former [89]. On the one hand, SDN allows a clear separation of concerns between service in the control plane and data plane, thus making the network architecture more easily handled. On the other hand, SDN mechanisms aim to balance the degree of centralized control/coordination through an explicit SDN controller and decentralized operations through flow-based routing and rescheduling within network components. It is necessary to reduce collection overhead and guarantee data effectiveness in the ECDriven-IoT system, which makes SDN very suitable for edge computing and IoT.

Salman et al. introduced a hardware architecture that integrates new technologies such as an SDN and **virtual network functions (VNF)** [153]. This architecture is used to implement and flexibly manage distributed edge networks, improve network scalability, and reduce costs. For example, for a typical factory, services and workloads are more IT-centric (e.g., factory data centers), and as they move down, they become OT-centric (e.g., factory machines). Software-defined resource allocation and management is gaining momentum in the edge computing paradigm as it can enable plant operators to better adapt to future needs. From a network perspective, this translates into an SDN implementing VNF throughout the plant.

Yaser et al. proposed a comprehensive framework model based on a software definition to simplify IoT management process [77]. It abstracts all control and management operations from underlying devices and places them in the middleware layer to hide the complexity of traditional system architectures. It is a model for forwarding, storing, and protecting generated data from IoT objects through integrated software, and is ideal for use in edge computing and edge network environments. Qin et al. designed a software-defined architecture by extending the **Multi-Network Information Architecture (MINA)** [140]. MINA is middleware with a multi-layer IoT SDN controller. The IoT SDN controller they developed supports a variety of scheduling commands. At the same time, this architecture can optimize the IoT network environment by using genetic algorithms. The architecture provides differentiated service quality for different IoT tasks across heterogeneous wireless networks.

3.3 Hybrid Hardware Architecture

In addition to general and mobile architecture, hybrid architecture has attracted the attention of many researchers. Sun et al. introduced a more flexible IoT architecture called edgeIoT, which uses fog computing to collect data at network edge [170]. Specifically, each fog node provides computing power and connects to **base stations (BS)**. The SDN-based cellular is used for packet forwarding between fog nodes and hierarchical calculations at each fog node.

Chang et al. also proposed a hybrid cloud architecture model, called Edge Cloud, designed to provide low-latency, high-bandwidth efficiency utilization [33]. As the name suggests, Edge Cloud combines edge networks with cloud data centers for data processing and vulnerable storage. Cloud data centers host regular computing and database components. This architecture takes advantage of edge and cloud computing to reduce latency and save bandwidth resources.

Munir et al. also proposed a similar edge computing architecture and designed a reconfigurable layered fog node architecture that can be suitable for fog computing applications [120]. Different from edgeIoT, the bottom-up abstraction of Munir's architecture includes the application layer, analysis layer, virtualization layer, and hardware layer. The hierarchical architecture facilitates the abstraction and implementation of edge computing paradigms that are distributed in nature and involve multiple vendors. This architecture analyzes the characteristics of applications and reconfigures the fabric resources to maximize the mobile workloads of the service for satisfying peak workload demands.

4 OPERATING SYSTEM

The IoT nodes usually connect to the Internet through communication protocols. Due to the heterogeneous nature of IoT, many different communication protocols are adopted in the system. Moreover, there are many IoT devices, including mobile phones, sensors, and other hardware platforms, such as Aurdriono [67], Raspberry [114], Intel, and Zolertia Z1 [53]. The operating system can bridge all the differences between these devices and provide a unified application programming interface. Considering the limited memory and power, traditional operating systems, such as the Linux and **Berkeley software distribution (BSD)**, are not suitable for IoT

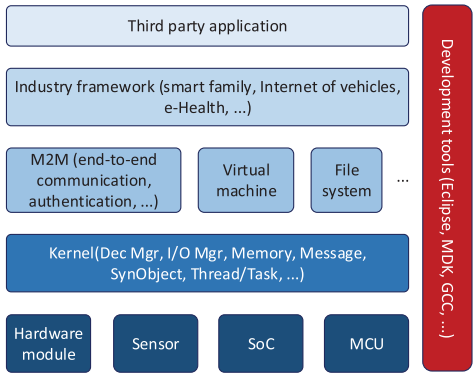


Fig. 5. A typical IoT OS infrastructure.

devices. The general operating system in IoT is ascendant. Many companies and research institutes invest many resources in researching IoT operating systems. Current popular IoT operating systems include LiteOS, Contiki, Win10IoT, FreeRTOS, and mbedOS [13]. Figure 5 shows a typical IoT operating system infrastructure. In general, IoT operating system is supported by the kernel, end-to-end communication, peripheral components (e.g., the file system, Java virtual machine, XML file parser), industry framework, and integrated development environment.

Although IoT operating systems have been developed for several years, applying edge computing in IoT brings some new demands:

- **Realize scalable kernel size.** The core of the operating system should be able to adapt to various configuration environments, from low-end embedded applications with small to tens of kilobytes of memory to complex applications with up to tens of Minionbytes of memory.
- **Satisfy real-time, high-reliability, and energy-saving requirements.** Kernel should also have some features of the general embedded operating system, such as predictable external event response time, predictable interruption response time, control, and management mechanisms for various external hardware.
- **Shield the characteristics of IoT fragmentation and provide a unified programming interface.** Fragmentation refers to various hardware device configurations, and different application areas vary widely. The "fragmentation" feature has constrained the development and growth of IoT.
- **Reduce the cost and time of application development.** The IoT operating system is a public business development platform with rich and complete IoT basic functional components and application development environments, which can reduce the development time and cost of applications.

There have been many surveys focusing on IoT operating systems [58, 80, 121]. However, all of them neglect to discuss the applicability of IoT OS in edge computing. Thus, they cannot illustrate the new features and requirements for operating systems in the ECDriven-IoT area. In this section, we discuss challenges for the operating system area, in terms of architecture, real-time support, networking technologies, and energy efficiency. Also, considering many IoT operating systems have been proposed in the community, we only illustrate some of the most used IoT operating systems in detail. We summarize the features of five popular IoT operating systems in Table 3.

4.1 Architecture of IoT Operating System

The operating system architecture can largely influence the kernel size of the system. Current mainstream OS architectures can be categorized as monolithic, micro-kernel, virtual machine,

Table 3. Comparison of Operating Systems

Feature	Contiki	TinyOS	LiteOS	RETOS	RIOT
Architecture	Modular	Monolithic	Modular	Modular	Modular
Real-time Support	No	No	No	Yes	Yes
Communication Support	FileSystem, Network, 6LoPWAN, Command, Line Interface	6LoWPAN, IPv6, Multi-hop Protocol	LoRaWAN, FileSystem, Network, 6LoWPAN	Static/Dynamic, Network	LoRaWAN, FileSystem, Network, 6LoWPAN, GraphicalUI
Power Consumption	No power management	Most efficient	Low	Timer ticks	Deep sleep mode

or layered ones. The monolithic architecture embeds necessary OS components and applications within its kernel, which could increase the kernel size and the difficulty of adding new features or deleting old ones. The micro-kernel architecture provides minimum functions in the kernel. Thus, applications and OSs are considered decoupled modules to make them easy to be added or be removed. So, the extension of such architecture will be more flexible. Also, a small kernel size makes the micro-kernel architecture more suitable for the ECDriven-IoT. Another type of OS architecture is the virtual architecture, in which a virtual machine mimicking hardware is exported to user programs. As an improvement to early monolithic systems, this system architecture has modules as a layer-based architecture. Each layer has different functionalities. However, a few IoT devices adopt the virtual and layered architecture, so we mainly focus on the first two architectures.

4.1.1 Architecture of Contiki. With a modular architecture, Contiki can efficiently reduce the size of the system, and multiple embedded OSs choose modular architecture due to its small size. Contiki is an event-driven OS with multi-threading supports, thus providing optional threading facilities for every process. Due to its customization, ease of extension, and better reliability, this OS can serve as a memory, file, and time server [80]. In the ECDriven-IoT, it can meet the requirements of heterogeneity.

4.1.2 Architecture of LiteOS. As a UNIX-like OS, LiteOS provides an abstraction of IoT devices. LiteOS is also a modular architecture OS. To minimize the programming learning complexities, it provides an efficient way and operating features, thus allowing user-friendly operations. The main feature of LiteOS is that it provides a shell and a hierarchical file system. Moreover, LiteOS has a much smaller code footprint, thus making it suitable for other platforms. The kernel of LiteOS is a subsystem of the whole system architecture. Dynamical loading and multi-threading are implemented in the kernel, thus providing concurrency supports [28]. However, LiteOS slows down the program execution under the limitation of hardware and consumption power. Multiple approaches have been proposed to solve this problem.

4.1.3 Architecture of RETOS. RETOS can solve various problems in IoT applications. RETOS was developed with the aims of reconfiguration, vigorous activity, and efficiency of resources, so it can efficiently deal with difficulties faced by IoT sensor nodes. As a modular system, RETOS ensures efficiency and reliability through a dual model, operation, and code checking. Moreover, RETOS can prevent hardware manipulation, memory access, kernel reading, and other dangerous operations [29]. The RETOS performs well in the field of wireless sensor networks. However, when applied in ECDriven-IoT, RETOS cannot satisfy other requirements, such as real-time response and a unified programming interface.

4.1.4 Architecture of RIOT. The ECDriven-IoT consists of billions of IoT devices and edge nodes. These devices usually have small memory, low power consumption, and limited communication bandwidth. Considering the requirements of real-time systems, the ECDriven-IoT needs a broader vision to embed intelligence to smartphones and portables, to achieve the connection of everything [179]. To save memory, RIOT adopts a micro-kernel architecture, so the size of its kernel is minimized. Adopting multi-threading aims to be effective with energy, memory, modulator, and APIs. Also, RIOT is a highly reliable OS, which is important in the ECDriven-IoT system.

4.2 Scheduling Algorithm and Real-time Support

In the ECDriven-IoT system, computing tasks are executed locally or offloaded to edge nodes. For real-time applications, computing tasks need to be completed in a short time. Especially in ECDriven-IoT, execution time becomes a critical metric because of transmission latency demands. The scheduling algorithm determines the execution orders, how tasks are executed, and when tasks are completed. A scheduler targets high throughput, high energy efficiency, fairness, and good resource utilization.

Scheduling is of great importance for deciding the time interval of task execution. Real-time scheduling algorithms aim to maximize throughput and complete tasks within the given time constraint [9]. In edge computing scenes, computing tasks can be divided into periodic and aperiodic tasks. So, these tasks are scheduled with periodic and aperiodic schedulers, respectively [160]. The operating system in the ECDriven-IoT is promising to handle real-time tasks efficiently. In the following, we will explore the scheduling schemes of the typical IoT operating systems.

4.2.1 Scheduling Algorithm of Contiki. Contiki has a hybrid programming model. It is primarily an event-driven OS but also supports multi-threads. As Contiki is event-driven, the processes will run to completion. Contiki provides support for multi-threading, which is implemented as a library on the top of the kernel. As for the real-time response, Contiki is mainly event-driven and does not implement any scheduling algorithm. Applications are handled according to their priority [32]. Thus, Contiki is not suitable for ECDriven-IoT, as it does not support real-time capabilities [58].

4.2.2 Scheduling Algorithm of TinyOS. TinyOS has multiple scheduling techniques and algorithms. In the event-driven model, a hardware interruption is handled by the event handler, and it can cause preemption. The single task queue adopts the **first-in-first-out (FIFO)** strategy and has no interruption for the FIFO algorithm, which is a non-preemptive algorithm. The new version of TinyOS has a new feature, priority scheduling. Thus, tasks with a higher priority can interrupt low-priority ones to meet their deadline demands. Moreover, TinyOS implements cooperative algorithms, such as **earliest-deadline-first (EDF)** and **adaptive double-ring scheduling (ADRS)**. The system enables preemption to ensure that tasks with a higher priority are completed before other tasks. However, preemption involves context saving and switching, which makes the scheduler more complicated and consumes more power. Thus, the preemption happens only in particular conditions. Owing to its efficient scheduling algorithm, TinyOS is considered one of the best OSs for IoT platforms. However, the real-time requirements are more strict when applied to edge computing. TinyOS still has many shortcomings in terms of real-time applications.

4.2.3 Scheduling Algorithm of RETOS. RETOS provides high concurrency with preemption functions. As a multi-thread OS, RETOS implements the boosting thread scheduler and introduces event-aware thread scheduling, which boosts the priority of threads [80]. To support real-time applications, RETOS enables developers to assign task priority explicitly and provides kernel dynamic priority management. Thus, RETOS can satisfy the latency requirement in the ECDriven-IoT system.

4.2.4 Scheduling Algorithm of RIOT. Using a scheduler through fixed priority and preemption, RIOT allows for soft real-time capabilities [71]. RIOT can handle low-priority tasks to deal with high-priority applications. RIOT applies a simple principle to achieve real-time scheduling: When a high-priority thread arrives, threads with low priority will be preempted, and the high-priority task runs right now until finished. What is more, RIOT can minimize response latency and power consumption by mimicking the parallel execution of events with the same priority. At the same time, it brings no context switches fee.

4.3 Networking Technologies in Operating System

In the ECDriven-IoT system, IoT devices offload collected data to edge nodes or cloud data centers. Thus, the connectivity between IoT devices and edge nodes is a fundamental guarantee to transport data. The essential elements of these devices are the device, local network, and the Internet. Communication technologies in the IoT community vary from device to device, and we will discuss them in the next section in detail. For operating systems, the network stack will hugely influence the performance of applications. Thus, operating systems should consider heterogeneous communication protocols.

4.3.1 Networking Stack for Contiki. Contiki supports not only a full TCP/IP stack but also a lightweight stack for low-power radio communication. Contiki implements μIP, the first standalone stack [11]. μIP supports IPv4 and IPv6 with a limited memory, which is suitable for the ECDriven-IoT. It can communicate with both the lightweight stack and full-stack, and its peers do not need to have a complete protocol stack. Also, Contiki applies *Rime* [46], so Contiki supports low radio communication and various communication modes. The module of Rime employs simple functions, making the stack lightweight and suitable for IoT. However, Contiki does not support as many communication protocols as the system requires when it refers to edge computing.

4.3.2 Networking Stack for TinyOS. In the ECDriven-IoT, both IoT devices and edge nodes are of limited energy and memory, and these nodes are connected to the Internet and communicate with each other. Therefore, we need an operating system that can provide stable communication links between devices. TinyOS adopts a protocol that can be used for the transport layer, networking layer, and medium access control layer. Thus it can consume as fewer layers as possible and make TinyOS reliable and robust [91]. Moreover, TinyOS can be suitable for various applications, so it is a good choice in the ECDriven-IoT.

4.3.3 Networking Stack for LiteOS. In LiteOS, MAC and communication protocols are taken as threads or files. Hence, it provides flexibility for different communication protocols. These protocols can be loaded dynamically as applications. During the communication, data packets will be sent to the port where the protocol is listening [28]. This feature makes it very suitable for IoT applications that vary in the communication protocol, but it has difficulty satisfying the latency requirements for treating the protocol as an application.

4.3.4 Networking Stack for RETOS. RETOS divides the kernel into static and dynamic parts for adapting to resource-constrained hardware environments. This design enables an easy programming interface for application developers. The static kernel part is optimized at the device driver level and guarantees the kernel performance in transmitting data packets and maintaining network connectivity. The dynamic kernel part is similar to LiteOS and implemented as loaded modules. So, different routing and communication protocols can be managed as the dynamic part and applied to different applications [31].

4.4 Power Consumption

The energy efficiency of the operating system in ECDriven-IoT is an essential requirement, as most IoT nodes in the system are power-limited. However, all nodes should communicate with other nodes, which is a process that consumes energy. Hence, a device should consume as little energy as possible [41, 175]. The heavy research area is based on energy-efficient protocols [30].

4.4.1 Power Consumption of Contiki. Since Contiki's kernel does not embed any power management algorithm, the power management strategies are customized by application developers. Contiki provides an interface to applications and allows them to manage the power system. In the ECDriven-IoT, the IoT nodes need power management schemes to meet the lower-power limits. As an event-driven OS, Contiki wakes up to respond to an interruption, and the poll handlers handle these events. In this manner, power management schemes must be designed to reduce the overall power consumption. When using Contiki, programmers must pay attention to power management to achieve the high energy efficiency of applications.

4.4.2 Power Consumption of TinyOS. Different techniques have been incorporated into TinyOS to achieve minimum power utilization. TinyOS with software thread integration is a method in which energy is conserved in TinyOS. By integrating software threads, TinyOS makes full use of idle time during transmission, processing, and sensing of data [139]. In TinyOS, which supports **high-power listening (HPL)**, TinyOS estimates the overall load of the sensing nodes and then dynamically allocates the required energy to the sensing nodes [93]. This method can only be possible with an accurate estimation of energy consumption in sensing nodes. Sensing nodes consume energy in a variety of ways [3]. TinyOS, in this case, is the most efficient OS, because it estimates the energy consumption by the sensing nodes, TinyOS itself, and its components. TinyOS supports various methods for estimating the energy consumption of different applications.

4.4.3 Power Consumption of LiteOS. LiteOS is a multi-threaded operating system, and while it does not introduce any overhead, it consumes more energy than TinyOS. However, LiteOS has a small memory footprint, so it can reduce energy consumption to a minimum, making it suitable for the ECDriven-IoT to some degree.

4.4.4 Power Consumption of RETOS. RETOS is a multi-threading operating system, so it adopts many scheduling operations. Threads scheduling involves significant context switching. All these operations are energy exhaustive. To address these issues, RETOS adopts a variable time tick, and timer requests are scheduled according to the remaining tasks, which can effectively minimize energy consumption.

5 COMMUNICATION PROTOCOL

The IoT sensing layer collects sensing data (e.g., sound, light, electricity) through sensors. Based on the terminal module of the network layer, base stations are connected to the network layer to realize data transmission after data acquisition. The network layer is responsible for transmitting data collected by the sensing layer. It should use different communication technologies based on specific scenario characteristics. The application layer can be viewed as the data and business platform of the IoT. As the collection point of all IoT terminal data, the data platform is responsible for unified data storage and analysis.

The communication protocol at the network layer is a group of competitors. It is also the focus of this section. In the ECDriven-IoT system, each terminal device and edge network device can be regarded as an independent individual. The communication between such independent components has the characteristics of hardware heterogeneity, low power, and short communication

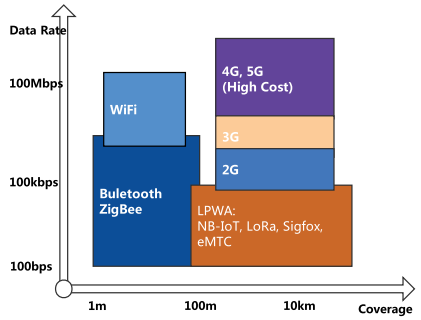


Fig. 6. Various wireless network communication protocols.

time. These features pose a great challenge in selecting and designing communication protocols. IoT network layer communication protocols can be divided into short-range and long-distance communication protocols. Short-range communication protocols include Wi-Fi [72], Bluetooth technology [66], ZigBee [50], and UWB [8]. Long-distance communication protocols include NB-IoT [5], LoRaWAN [165], and 5G [14]. Figure 6 shows the rate and coverage of various wireless network communication protocols.

5.1 Short-range Communication Protocols

We consider placing all computing processes in edge devices and networks as much as possible in an IoT environment. Therefore, the inter-communication process between edge devices and edge networks in computing is particularly critical. The short-range communication protocols have several advantages and disadvantages, each of which applies to different IoT environments. Therefore, according to specific communication environments and requirements, it is worthwhile to study and improve the communication protocols so that the calculation process can be more efficient.

The most popular Wi-Fi technology has a fast transmission speed. However, with the speed increase, the power consumption also increases sharply, and then the transmission distance becomes a bottleneck. Long-distance transmission requires an **access point (AP)** to bridge the data link as a middleman, which will largely increase the cost. Therefore, Wi-Fi technology is more suitable for indoor wireless Internet access scenarios and terminal applications such as PCs and PDAs. Both Bluetooth and Wi-Fi work on the 2.4 GHz band, so there are some interference problems in the same frequency band. Bluetooth consumes slightly less power than Wi-Fi, and the transmission speed is far lower than Wi-Fi. It is widely used in asset tracking, location tags, and medical sensors, such as smartwatches and Bluetooth positioning. ZigBee technology has relatively low power consumption and short communication distance. It is mainly used in wireless sensors and medical scenes. UWB technology has a relatively clean frequency band and no interference from other communication technologies. It is currently used in high-precision positioning scenarios. These popular short-range communication protocols can be well applied in the ECDriven-IoT as the communication basis between edge devices.

5.1.1 Wi-Fi. Vivek et al. adopted Wi-Fi and ZigBee to implement a home automation system [178]. With the help of light, temperature, and safety feedback loops, the system provides comfortable brightness, temperature regulation, and basic safety by utilizing popular Wi-Fi signals provided by IoT devices, such as smart air conditioners, smart lights, and thermostats. Shi et al. used Wi-Fi signals to capture the behavioral characteristics of daily human activities in their paper [161]. This method does not require hardware devices and only needs to recognize users' unique physiological and behavioral characteristics through the Wi-Fi signal, thereby realizing

some functions such as user authentication. Meanwhile, Acer et al. utilized Wi-Fi aware network search to analyze IoT data [4].

Wi-Fi technology's data rate is fast enough for ECDriven-IoT applications. However, as transmission speed increases, the power consumption of devices also increases dramatically. In edge devices, energy saving is a critical factor. So, Wi-Fi may not adapt to some scenarios well in ECDriven-IoT systems.

5.1.2 Bluetooth. In recent years, much of the work on wireless sensor networks targets to be efficient, low cost, scalable, and easy to deploy. Optimizing battery usage and power consumption reduces costs and extends sensor life. Bluetooth is an ideal communication protocol for the ECDriven-IoT. Its low power consumption makes edge devices run for a long time and reduces maintenance. Generally speaking, edge computing does not require high data transmission speed.

Nair et al. introduced an architecture that uses the **Bluetooth low energy (BLE)** communication standard and hybrid topologies to reduce the power consumption of communication systems [122]. The BLE is considered a low-power version of traditional Bluetooth. However, the extensive use of BLE in deployments can lead to high collision rates, especially in device-intensive IoT environments. To alleviate this contradiction, Harris et al. proposed opportunistic listening, an extension of the BLE active mode with tags and scanning devices [69].

For smart cars that use Bluetooth technology, users can connect their smartphones with their cars. In this case, they can replace the phone's speaker and microphone with the car's ones, and use the car's devices to make a call or message. At the same time, you can also use your mobile phone to read diagnostic data about your body everywhere [125].

5.1.3 ZigBee. The smart home is an IoT application closely related to human life. ZigBee is a widely-used communication protocol in the smart home. At the same time, ZigBee is also the ideal communication protocol for the ECDriven-IoT. Because of its low power consumption, ZigBee can be suitable for IoT environments that include massive wireless sensors.

Moravcevic et al. proposed a way to integrate the ZigBee protocol into smart homes [116]. This approach firstly defines a home device as a service that can add ZigBee devices from different manufacturers to the system. Various home devices on the market today can communicate using the ZigBee protocol. So, energy-efficient devices that support ZigBee can be added to the smart home system.

In addition to smart home applications, ZigBee can combine with other new technologies to control and communicate between IoT devices. Ferreira et al. proposed a model combining event capture and device control [55]. This model is implemented using basic general techniques such as RESTful API or UPnP. With ZigBee communication technology, this easy-to-capture body interaction allows developers to make fun and useful applications. Another important aspect of this technology is the data exchange between various types of endpoints by using standardized communication protocols. It allows a wide variety of programs to utilize data exchange to achieve specific user needs, even though these programs are independently developed by developers around the world.

Meanwhile, the privacy and security of IoT in the communication protocol layer have also received the attention of researchers. Ronen et al. discovered a new type of worm threat [147]. When there are too many IoT devices, and the density exceeds a certain amount, the worm will spread rapidly in a large area of the communication layer, and adjacent IoT devices will infect each other. They verified the infection with the help of the Philips Hue Smart Light platform. They use ZigBee only as their wireless communication technology and found that this worm threat can be transmitted directly between various types of adjacent light bulbs, which are light bulbs that can communicate with each other on the same platform. The contagious nature of this attack can cause city lights compromised on a large scale in a matter of minutes.

5.1.4 UWB. **Ultra-wideband (UWB)** is a communication technology that uses a non-sinusoidal narrow pulse of nanoseconds to microseconds to transmit data. UWB was used in early applications of short-distance high-speed data transmission. In recent years, many researchers have begun to use their sub-nanosecond ultra-narrow pulses for short-range accurate indoor positioning. The UWB architecture finds its place in surveillance systems, medical applications, and IoT applications. Antennas using UWB communication technology have compact hardware and low power consumption. This feature is critical for portable IoT devices. Bekasiewicz et al. described this UWB antenna structure for IoT [21]. This well-designed structure enables small-sized physical areas while maintaining electrical performance. What sets UWB apart from other short-range communication protocols is that its frequency band is relatively clean. It is usually used for precise positioning, but the application scenario is not rich enough.

5.1.5 6LoPWAN. When we first considered the sensor communication network, the first thing that came to mind was the use of **Internet Protocol (IP)**. IP is unsuitable for sensor or personal area networks, because it is too heavy for these applications. Recently, more and more research has started to work on low data rates, low power consumption, and small-size IP protocols. The **IPv6 low-power wireless personal area network (6LoWPAN)** is a low-speed wireless network standard. It supports the use of IP in IEEE 802.15.4 wireless networks [76]. The key point in the breakthrough of 6LoWPAN-related work is to achieve a very compact and efficient IP, eliminating the communication difficulties brought by the unique protocol standard. The 6LoWPAN protocol has the following features:

- **Popularity.** IP networks have been adopted widely. IPv6, the core technology of the next-generation Internet, is also accelerating its popularity. It is more acceptable to use IPv6 in low-speed wireless personal area networks.
- **Applicability.** The IP network protocol stack architecture is widely recognized, and the low-speed wireless personal area network can be developed simply and efficiently based on this architecture.
- **Adequate address space.** When IPv6 is applied to low-speed wireless personal area networks, the biggest highlight is the large address space, which is precisely needed for deploying large-scale, low-speed wireless personal area network equipment.
- **Stateless automatic address configuration.** When a node boots up in IPv6, it can automatically obtain a MAC address and configure an IPv6 address. This feature is attractive for sensor networks, because it is not feasible to configure the user interface for sensor nodes in most cases, and nodes must have automatic configuration capabilities.
- **Easy access.** Low-speed wireless personal area networks use IPv6 technology to make them easier to access other IP-based networks and next-generation Internet, enabling them to take full advantage of IP network technologies.

These features are ideal for ECDriven-IoT communication environments, especially for IoT applications that require large-scale deployment of low-power communication devices. Mulligan et al. introduced a simple 6LoWPAN protocol architecture and compared it with ZigBee [119]. Ma et al. introduced the advantages of 6LoWPAN and details of some key technologies [103].

The short-distance communication protocols in IoT mainly include Wi-Fi, Bluetooth, ZigBee, UWB, and 6LoPWAN. Table 4 shows the main features and differences between them.

5.1.6 Cross Technology Communication. Because of the hardware complexity and network heterogeneity of the ECDriven-IoT, **cross-technology communication (CTC)** is becoming a popular research direction. Considering the IoT environment of densely deployed devices, mainstream wireless technologies typically share radio spectrums. Wireless technologies shared spectrums

Table 4. Comparison of Wi-Fi, Bluetooth, ZigBee, UWB, and 6LoPWAN

	Wi-Fi	Bluetooth	ZigBee	UWB	6LoPWAN
Data Rate	1 Gbps or more	1 Mbps	100 Kbps	53–480 Mbps	250 Kbps/40 Kbps
Communication Distance	20–200 m	20–200 m	2–20 m	0.2–40 m	10–100 m
Frequency Band	2.4 GHz/5.8 GHz	2.4 GHz	2.4 GHz	3.1 GHz/10.6 GHz	2.4 GHz/915 MHz
Security	Low	High	Medium	High	High
Power Consumption	High	Medium	Low	High	Medium
Cost	High	Low	Medium	High	Low
Application	PC, PDA wireless Internet access	Mobile phone transmission, Medical health	Wireless sensing	Accurate locating	Smart home

Table 5. Comparison of NB-IoT, LoRaWAN, and 5G

	Spectrum Cost	Module Cost	Coverage	Battery Performance	Data Rate	Flexibility
NB-IoT	Authorized, high cost	≤20 dollars	18–21 km	Fast power consumption	200 Kbps	Limited by operator
LoRaWAN	Unauthorized, low cost	≤10 dollars	12–15 km	Long lasting electricity	0.2–37.5 Kbps	Self-built network
5G	Authorized, high cost	~200 dollars	10–100 m	Long Life Time	10 Gbit/s	Base-station provide

will inevitably interfere with each other. But every coin has two sides. This drawback also makes cross-protocol communication possible.

For example, Zhou et al. introduced a cross-technology communication protocol, ZiFi [200]. The system uses ZigBee radios to identify the presence of Wi-Fi networks through the unique interference characteristics generated by Wi-Fi beacons, which can significantly improve the standby energy efficiency of Wi-Fi devices. Kim et al. introduced FreeBee [87], which supports three popular wireless technologies (Wi-Fi, ZigBee, and Bluetooth) across technology broadcasts. FreeBee's core idea is to modulate symbolic messages by changing the timing of three standard beacon frames without additional frames and traffic.

5.2 Long-distance Communication Protocols

In a long-distance scenario, if terminal devices cannot solve the power supply problem, a technology with lower power consumption and broader coverage is needed to meet the requirements of IoT communication. Thus, driven by business and technology, some researchers and enterprises have developed a new type of communication technology, LPWAN, a low-power WAN technology [165]. Long-distance and low-power communications have a broader application prospect in future IoT environments, because not all IoT and edge devices are in close proximity. Thus, LPWAN is more suitable for **machine-to-machine (M2M)** communication in edge computing. Table 5 shows the main features and differences between these three long-distance communication technologies.

LPWAN is a long-range wireless network communication technology that has been widely used to optimize M2M communications in IoT applications. The main technical advantages of

LPWAN are ultra-low power consumption, long-distance, low throughput, and broad coverage. Some typical applications include urban coverage, remote meter reading, manhole cover testing, and offshore fishing vessel testing. Long-distance communication protocols are also often used for IoT communication for specific remote edge devices.

5.2.1 NB-IoT. NB-IoT is called narrowband IoT and can be deployed directly on LTE networks. Good compatibility reduces the cost of deployment. It has lower power consumption. Theoretically, the terminal module carrying NB-IoT uses a battery and has a standby time of up to 10 years. The reduction in module costs has also led more companies in the market to use this technology. In 3GPP, an LTE-based narrowband system has been introduced to support the IoT [22].

Mangalvedhe et al. [107] introduced the NB-IoT system design, some potential problems, and solutions for the actual deployment system. Adhikary et al. focus on the coverage of NB-IoT in IoT environments [5]. They believe that NB-IoT provides broader coverage than traditional LTE systems. Petrov et al. proved the possibility of applying NB-IoT to IoT cars [136]. They conducted a comprehensive system-level assessment revealing the impact of in-vehicle NB-IoT communication on critical metrics, such as reliability, transmission delay, and energy efficiency. The results show that the development potential of NB-IoT may meet the future performance requirements of IoT vehicles.

The NB-IoT has four features: *(1) Wide coverage area.* NB-IoT provides better indoor coverage. *(2) Strong connectivity ability.* NB-IoT supports more than 100,000 connections in a single workspace. *(3) Low power consumption.* Usually, the standby time of the NB-IoT terminal device can last for several years. *(4) Low cost.* The NB-IoT license band can be deployed in-band, guard band or independent carrier mode to coexist with existing networks. Therefore, NB-IoT can be widely used in various related industries, such as intelligent remote meter reading, asset tracking, intelligent parking, and intelligent mechanized agriculture.

5.2.2 LoRaWAN. LoRaWAN is a long-distance communication protocol different from NB-IoT. It is an ultra-long-range wireless transmission technology based on chirp spread spectrum technology promoted and adopted by Semtech. At the most basic level, wireless protocols like LoRaWAN are relatively simple. LoRaWAN is a star topology [132]. This type of structure is generally better than a mesh network, because it has advantages in maintaining battery power and increasing communication range.

Many researchers have studied the performance and metrics of systems using this protocol. Petric et al. used the LoRa FABIAN protocol stack to generate and then observe the traffic between IoT nodes and LoRa stations to perform the test [135]. In addition to long working life and low production costs, coverage is a key performance indicator for long-distance communication protocols. Petajajarvi et al. studied the coverage of LoRaWAN technology through actual measurement work [137]. Bor et al. developed a platform for LoRa performance evaluation and described a protocol that leverages LoRa's unique features on top of LoRa's physical layer [25]. This protocol enables energy-efficient wide-area multi-hop data collection.

Because of the similar name, many people confuse LoRaWAN with LoRa. However, LoRaWAN refers to the networking protocol of the MAC layer, and LoRa is just a protocol for the physical layer. From the perspective of network layering, LoRaWAN can use any physical layer protocol, and LoRa can also be used as the physical layer of other networking technologies. Several technologies that compete with LoRaWAN also use LoRa at the physical layer. LoRa is one of the LPWAN communication technologies and a long-distance communication solution based on chirp spread spectrum technology. This solution changes the previous trade-offs between transmission distance and power consumption to provide users with a simple system that can achieve long-distance, long battery life, and large capacity, thereby expanding the network.

5.2.3 5G. With many emerging scenes, such as autonomous driving, and smart cities, requiring higher data rates, the **fifth-generation (5G)** cellular network has arisen with a high data rate and broad communication areas. In ECDriven-IoT, the low latency and high energy efficiency requirements lead to smaller transmission time intervals. Moreover, small cells can achieve high area capacity in densification. All of these have led to new radio access technologies and a new core network [158]. With the help of higher frequencies, large-scale antennae can be deployed at base stations. Thus array gains can overcome the shortage of higher path loss and can gain spatial multiplexing [94].

With a resilient cloud-native core network and end-to-end support for network slicing, 5G is distinguished by high flexibility and scalable network technology. Based on three major user case domains, 5G can support deterministic and isochronous communication with high reliability and availability. 5G can be applied in the ECDriven-IoT with hard guarantees for latency bounds, packet loss, and reliability, as well as synchronization down to the nanosecond level [59]. Moreover, the seamless change of the application server can be supported by 5G with low latency. 5G application enablers will be studied for interactions between users, application servers, and the network in a complementary manner [82].

6 COMPUTING

We are not only concerned with the underlying hardware and communication protocols but also computational processes in IoT and edge networks. These research areas include algorithmic acceleration for different scenarios, distributed computing in IoT [48], green computing [92], and caching [183], as well as SDN and NFV. These new computational studies lack sufficient focus, but they all have huge development prospects and can optimize edge computing in IoT.

6.1 Computation Offloading

The ECDriven-IoT has many hardware and protocol problems, so the computing capability of edge nodes is limited and how to compute efficiently is still a tricky challenge in the ECDriven-IoT [90, 104]. Some innovative algorithms have been proposed to overcome the limitations of the ECDriven-IoT.

Data can be processed and pruned in edge nodes before being transmitted to the cloud through intelligent gateways. However, considering edge nodes have limited computing capabilities and energy power, only a part of the data can be processed locally. In computation offloading, it refers to the offloading decision [39], server selection, Wireless resource allocation, transmission power setting, computation resource allocation, and the slot partition. Aazam et al. expanded the integration of IoT and cloud computing. They analyzed the network architecture and performance of this concept [1].

Chen et al. proposed a game-theoretic approach to achieve efficient computation offloading for edge computing and formulate the distributed computation offloading decision-making problem among end devices as a multi-user computation offloading game [37, 38]. Mao et al. investigated a green edge computing system with energy harvesting devices and developed a computation offloading strategy that jointly decides the offloading decision, the CPU-cycle frequencies for mobile execution, and the transmit power for computation offloading [109]. Moreover, research has been done on resource allocation for a multi-user system based on **time-division multiple access (TDMA)** and **orthogonal frequency-division multiple access (OFDMA)**. The optimal resource allocation is formulated as a convex optimization problem for minimizing the weighted-sum mobile energy consumption under the constraint of computation latency [193].

6.2 Distributed Computing

As IoT and edge computing both have the feature of being distributed everywhere, how to organize system resources and combine them is another difficulty in computing in the ECDriven-IoT. In this context, distributed computing provides an opportunity to solve this problem efficiently.

Distributed computing is the process of aggregating the power of several computing entities that are logically distributed and may even be distributed in geography, to collaboratively run a single computational task transparently and coherently, so that they appear as a single, centralized system.

Chien et al. introduced the idea of distributed computing in IoT [40]. They proposed a distributed smart camera architecture used in video sensor networks to accelerate computer vision algorithms for smart cameras in the IoT. Similarly, edge computing is also a distributed computing method in IoT. Hesham et al. realized this vision. They used edge computing in a distributed computing environment to move workloads from a centralized cloud to the network edge while verifying edge efficiency and resourcefulness [48]. Distributed computing is a cheap and efficient alternative computing method. It can compute in any location, so it can efficiently make use of the computing capability of edge nodes and reduce the transmission bandwidth requirements, which can push the development of the ECDriven-IoT.

6.3 Caching

In computation offloading and distributed computing, massive data are generated and transmitted among edge nodes, which influences communication latency and IoT devices' power. Caching is an effective method to increase computing speed and save computing bandwidth. In the ECDriven-IoT scenes, some applications require few computing resources and storage in IoT devices and edge nodes. Thus, these remaining caches can be utilized more efficiently, reducing latency and improving system efficiency. The idea of caching is transplanted to edge computing to reduce the data transmission cost and system delay. At the same time, it will make design and development more difficult.

Combining caching and edge computing with IoT is a promising means of alleviating traffic in a backhaul. With network stability taken into account, Du et al. formulated a stochastic optimization problem to jointly optimize the offloading decision and cache decision making [45]. Xia et al. [186] investigated a cache-aided mobile edge computing network, where the source offloads the computation task to multiple destinations having computation capacity with the help of a cache-aided relay. However, their work does not explore the cache-aid with IoT. So when applied to IoT, how to efficiently solve the complexity of IoT and edge computing is still a challenge.

Distributed caching is widely used in the caching deployment of base stations. However, the caching capacity of a single BS is generally particularly limited, which will degrade the performance of the wireless mobile network [154]. Li et al. designed a collaborative cache scheme in the heterogeneous mobile edge computing network, in which the edge caching of macro base stations and small base stations are utilized to bring storage resources closer to users [96]. For most IoT devices, the smaller the cache size, the more complex the topology. Therefore, more research should explore the combination of edge computing and IoT in caching to make it a more efficient way to reduce latency and energy consumption.

Another important research direction for cache computing is cached content placement [128]. In general, cached content placement system information is updated continuously to improve the cache hit ratio. In an edge computing environment, caching typically occurs on user devices. The cache at the user device may allow the user to download the requested content in a more efficient manner using **device-to-device (D2D)** communication [184]. As the density of edge devices increases, the cache advantages of user devices will be reflected in their low deployment costs.

6.4 Software Defined Network and Network Function Virtualization

With the development of the ECDriven-IoT, the number of users has grown exponentially. Network developers, service providers, and network carriers provide up-to-date services to the users. Thus, the network is expanding exponentially in the same way and how to efficiently operate these networks is complex. Thus, SDN has emerged as efficient network deployment and management solution [16]. SDN provides a separation between the control plane and the data plane, which equips network developers with the ability to efficiently expand the network and the convenience to manage network resources. Furthermore, network operators can configure, upgrade, and maintain network resources dynamically. Since SDN is logically defined, the controller can access these network resources more efficiently [195]. In edge computing, SDN can provide flexibility and manageability. For the ECDriven-IoT, the data generated and collected in IoT devices need to be routed to the edge or cloud. SDN can alleviate these complex communication requirements with service discovery, provisioning, and orchestration at the edge nodes.

Network function virtualization (NFV) deals with the hardware-oriented function transformation, such as firewalls or DNS for the software applications [111]. It can provide dynamic service orchestration. Thus, efficient services deployment can be realized without hardware support and achieve the **service function chain (SFC)** [86]. Due to the heterogeneous nature, ECDriven-IoT, SDN, and NFV can be integrated into a whole system and interact with each other. NFV can operate as a service orchestrator, and SDN can automate the service chaining by installing customized flow rules at the forwarding stations. This system can improve performance in real-time applications and reduce transmission delays, which are significant metrics in these application scenes [81].

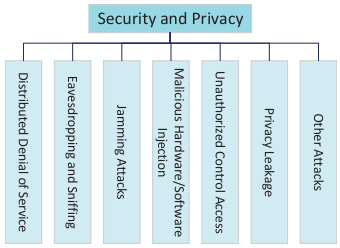
7 SECURITY AND PRIVACY ISSUES IN ECDRIVEN-IOT

When applying edge computing to IoT, new and unforeseen security and privacy problems will arise. Due to the high mobility and heterogeneous features of ECDriven-IoT, the system is more vulnerable to potentially malicious activities. In addition, many advanced security mechanisms cannot be transplanted to IoT devices and edge nodes owing to the limited computing capabilities and power. In the ECDriven-IoT system, the communication between IoT devices and edge nodes is relatively frequent, thus making the network more unstable. In terms of privacy, many users' privacy-sensitive information will be stored in IoT devices and edge nodes, or transported to the cloud server. In such a distributed architecture, security and privacy become crucial challenges. This architecture is more vulnerable to attacks and threats. In the communication, computation, and storage process, malicious attacks will be encountered [127]. Figure 7 shows possible security and privacy attacks and their solutions in ECDriven-IoT.

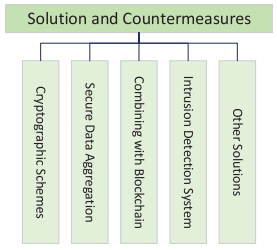
7.1 Security and Privacy Threats

In this section, we will illustrate potential attacks in the ECDriven-IoT system. Different kinds of threats of ECDriven-IoT networks will be introduced, as well as their sources at different levels. Owing to the features of IoT and edge computing, as well as the application scenes, the ECDriven-IoT faces many security and privacy threats, such as **distributed denial of service (DDoS)** attacks, physical attacks, eavesdropping or sniffing, and privacy leakage [101, 146].

7.1.1 DDoS Attacks. DDoS attacks toward edge-computing nodes consist of outage attacks, sleep deprivation, and battery draining. In outage attacks, edge nodes will be exposed to unauthorized users and unable to perform in the designed way [117]. As a much harder-to-detect attack, sleep deprivation adversaries overwhelm edge-computing nodes with an undesired set of legitimate requests. Battery-draining attackers will deplete the battery of the edge-computing nodes or



(a) Security and privacy threats.



(b) Solutions and countermeasures.

Fig. 7. Security and privacy threats and solutions in ECDriven-IoT.

IoT sensors/devices. DDoS attacks can also occur at the communication layer with continuous or intermittent jamming [12].

7.1.2 Eavesdropping or Sniffing. In eavesdropping, adversaries can listen over communication links to acquire private information, thus leading to privacy concerns. Through this attack, attackers can take much important information about the system, such as user names, personal information, or some commercial secrets [146].

7.1.3 Jamming Attacks. Jamming attacks are a kind of energy-consumption denial-of-service attack. They can be launched in the link or physical layer. These attacks often utilize the weakness of IoT systems and edge computing architectures. In jamming attacks, adversaries intentionally flood the network with forged messages to exhaust the systems' communication bandwidth, computing sources, and storage volumes, making the whole system unable to carry out tasks [169].

7.1.4 Malicious Hardware/Software Injection. Attackers can inject malicious inputs into the edge-computing node servers and perform hacking by adding unauthorized software or hardware components to the communication between IoT devices and edge-computing nodes. This attack can also make adversaries acquire many unauthorized data, thus raising privacy concerns [12].

7.1.5 Unauthorized Control Access. In the ECDriven-IoT paradigm, edge computing and IoT nodes communicate with each other to access or share their data. However, these devices and nodes can not use complicated methods to authorize permission access. Attackers can access one of the unsecured edge nodes and possibly control the whole system, which is of great danger.

7.1.6 Privacy Leakage. As for privacy, the ECDriven-IoT can be applied to many personal scenes, such as healthcare and smart homes. Thus, such personal information will be collected by IoT devices and then transmitted to edge nodes to be processed and stored. However, considering the limited self-protecting ability of edge computing nodes and IoT devices, privacy leakage potential exists in the ECDriven-IoT system. Furthermore, the ECDriven-IoT system will acquire

position information to serve the users, so attackers can obtain users' physical position or other sensitive information if they compromise the devices. Thus, how to guarantee data privacy in the process from collection to storage is a crucial problem [12].

7.1.7 Other Attacks. The ECDriven-IoT is an emerging paradigm that combines heterogeneous resources and devices. Thus the system is vulnerable to many attacks from different levels. Beyond the attacks mentioned above, there still are non-network side-channel attacks [117], routing information attacks [182], forgery attacks [163], replay/freshness attacks [54], and inessential log attacks [105]. Considering the importance of system security, the community has proposed many attack countermeasures to protect this paradigm.

7.2 Solutions and Countermeasures for Security and Privacy

In this subsection, mainstream solutions against security and privacy attacks are discussed in detail. And we analyze the advantages and disadvantages of existing countermeasures when applied to the ECDriven-IoT.

7.2.1 Cryptographic Schemes. Cryptographic schemes are widely used and serve as efficient strategies to protect communication protocols against various attacks [83]. Encryption/decryption solutions are inapplicable for wired networks owing to the limited resources in IoT nodes. The standard encryption/decryption methods are memory- and computing-exhaustive. However, edge-computing nodes are typically tiny sensors with limited resources, such as battery power, computing capabilities, and storage memory [100]. These techniques have been investigated and improved to suit the ECDriven-IoT paradigm. Chen et al. proposed a new security access method without cryptographic schemes for the ECDriven-IoT paradigm. This solution benefits from the difference in the hardware of heterogeneous wireless accesses instead of password authentication [36]. Alababy et al. constructed a valid network security model to protect data and suggested a solution to protect the system from several attacks[10]. Mollah et al. proposed a secure data-sharing scheme and a secure searching strategy. This sharing scheme uses public and private key encryption to ensure its security. Thus, applications can perform secure data search and sharing [113].

7.2.2 Secure Data Aggregation, Deduplication, and Analysis. When considering how to strengthen the system security and privacy paradigm, we naturally take data security and privacy as the most crucial component. For data security, the system must trace data from aggregation, transmission to analysis, throughout the data lifetime. **Secure data aggregation (SDA)** is a highly secure, privacy-preserving, and efficient data compression strategy [181]. Individual devices send their data to edge computing nodes. And then, edge nodes aggregate these data by computing the multiplication of individual data in SDA. To provide users fair incentives, Okay et al. employed signature techniques, the Boneh-Goh-Nissim cryptosystem, and secret sharing. By adding noises into the data for differential privacy, oblivious data security and fault tolerance can be achieved [112]. In Reference [151], to protect against false data injection attacks, they proposed to filter out the inserted data. To achieve better privacy, the Paillier cryptosystem was modified to achieve better privacy protection and is used to encrypt consumption data from users [129]. In the ECDriven-IoT paradigm, there is an increasing demand for systems that can provide cost-efficient secure data storage. For example, Storer et al. proposed **secure data deduplication (SDD)** to achieve secure data storage. And many deduplication methods have been proposed since then [97, 141]. As an effective way to achieve data security and space efficiency, SDD can be applied to single-server and distributed storage systems [168]. Furthermore, **artificial intelligence (AI)** functionalities have shifted from cloud servers to edge devices, which can potentially improve security and privacy in the ECDriven-IoT [12].

7.2.3 Combining the ECDriven-IoT With Blockchain Technologies. Blockchain is viewed as a distributed tamper-resistant database that can be maintained, shared, replicated, and synchronized by multiple participants in the **peer-to-peer (P2P)** network [191]. Considering the security and privacy problems in edge computing, blockchain can be a potential technology to establish a secure, trusted, and decentralized intelligent system in ECDriven-IoT [75, 84, 187]. When applying blockchain in edge computing and IoT, it can ensure a reliable tracking of ECDriven-IoT data transmission and eliminate the requirement for a central trusted intermediary between the communicating IoT edge devices [126]. Aiming to improve authentication efficiency, Guo et al. [63] combined edge computing with blockchain to build a distributed and trusted authentication system. This system can guarantee trusted authentication and reliable traceability in edge-computing nodes. It consists of both a physical network layer, a blockchain network layer, and a blockchain edge layer to support edge computing. As for the use of blockchain, the system can prevent network connections from being attacked. Zhao et al. proposed a flexible and configurable blockchain architecture that provides a mutual authentication protocol and secure consensus, making it suitable for the ECDriven-IoT. In this architecture, user-defined sensitive data will be encrypted before storage. Besides, the smart contract is adopted to achieve conditional access, which can protect blockchain data and transactions [198]. In the ECDriven-IoT, amounts of data is shared among edge nodes. But owing to the lack of trust, data sharing is hard to complete, and it is difficult to overcome the computation limitations at the edge. Xu et al. developed a blockchain-based big-data-sharing framework to support various applications across resource-limited edge nodes with a low-computation-complexity consensus mechanism. This framework can be applied to edge devices with low computation and provide security and privacy protection [187]. Furthermore, Kang et al. proposed to utilize consortium blockchain to establish a secure and distributed vehicular blockchain system for data management and storage by deploying smart contracts [84].

7.2.4 Trusted Execution Environment. With the emergence of the ECDriven-IoT, edge devices can process large data streams. However, this process exposes the data to a sophisticated vulnerable attack environment at the edge. The **trusted execution environment (TEE)** can isolate data and their computations to shield them from edge attacks. Guan et al. [62] proposed a system shielding legacy applications from untrusted operating systems by constructing a trusted execution environment for security-critical applications. Thus, edge applications can execute in these environments to prevent data from attacks. To optimize data plane performance when achieving the TEE, Heejin et al. advocated a stream analytics engine called StreamBox-TZ to offer strong data security, verifiable results, and good performance, thus making efficient data analytics in the edge possible [131].

7.2.5 Intrusion Detection System. The **intrusion detection system (IDS)** mainly focuses on detecting attacks [164, 166]. However, except for monitoring the network's operations and links, the IDS can mitigate security threats and report suspicious activities to make the system more stable and secure. Furthermore, the IDS can detect routing attacks and Black Hole attacks [173]. Hosse [74] presented a new distributed and lightweight IDS based on an **artificial immune system (AIS)**. This system consists of the cloud, fog, and edge layers, making it suitable for edge computing. Wang et al. proposed an IDS architecture for the ECDriven-IoT, which integrates a trust evaluation mechanism and service template with balanced dynamics [180]. This trust evaluation mechanism can strengthen the system's security.

7.2.6 Other Solutions. Table 6 shows the advantages and disadvantages of these solutions. As there are many kinds of attacks on security and privacy, these solutions are various to make ECDriven-IoT as secure as possible. Beside aforementioned solutions, policy-based mechanisms

Table 6. Solutions and Countermeasures of Security and Privacy Threats

Solutions	Layer	Advantages	Disadvantages
Cryptographic Schemes	Communication Layer	Highly secure	Battery power, computing capability storage memory
Secure Data Aggregation, Deduplication, Analysis	Data Layer	Protect data security and privacy	Consume power, render sensitive data to intruders network bandwidth
Combine with Blockchain	Architecture Layer	Trusted, reliable, and secure	More complicated system more computing capability
Intrusion Detection System	Communication Layer	Mitigate security threats	Resource consumption

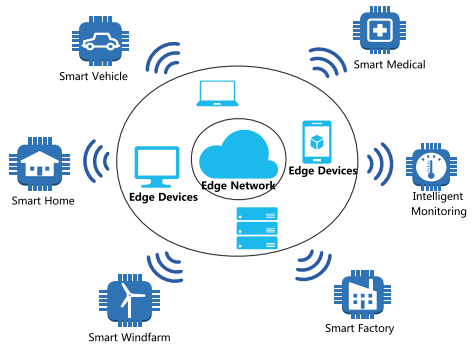


Fig. 8. Application scenarios of edge-computing-driven IoT.

[117], secure firmware update [117], and reliable routing protocols [102] also play a very important role in making the system secure. As for the security of data, many solutions have been proposed in academia, such as de-patterning data transmissions [197], decentralization [197], and authorization [101]. In many applications, these solutions can be combined with other solutions to work together. Although there have been many solutions, security and privacy in ECDriven-IoT remain a big challenge in the present.

8 APPLICATIONS

The ECDriven-IoT is suitable for many applications. This section will explain how it works and how to satisfy the requirements in these application scenes. The ECDriven-IoT plays a major role in responsive and latency-sensitive IoT applications.

8.1 Smart Homes and Smart Cities

One of the pioneering applications of IoT technology is in home automation and consumer electronics [145]. Shi et al. introduced some of the challenges and application prospects in smart life [162]. Figure 8 shows some typical application scenarios of the ECDriven-IoT. More and more applications are benefiting from the advantages of edge computing, such as smart homes, smart vehicles, smart medical systems, and intelligent monitoring.

The smart home is a popular IoT application scenario, and some established market products are widely acclaimed. These products range from simple thermostat sensors to more sophisticated automation systems, like smart metering, smart heating and lighting, smart cleaning services, and smart home entertainment systems. However, the smart home is not simply adding IoT communication modules to traditional home products. In addition to smart devices communicating with each other, IoT data such as room environment data is also essential for smart homes. Therefore,

the deployment of a large number of inexpensive sensors and controllers is needed as part of a collaborative effort in-house. The large amount of data generated by these sensors will be transmitted and used by other IoT devices. Considering data transmission bandwidth pressure and privacy data protection requirements, edge computing can be an ideal choice for building smart homes [174]. Furthermore, edge computing will bring other features such as easy installation, relocation, privacy preservation, and flexibility [148].

Smart homes can be extended to smart communities and even smart cities, and are expected to become an indispensable part of human life. ECDriven-IoT systems can also serve as the ideal architecture for smart cities. According to the data growth trend of a city today, the data will grow exponentially in the future. These data are generated by public safety, health, utilities, and transportation. Processing the data at the network edge is more efficient than building a cloud data center. Next, considering the sudden events and public safety in the city, edge computing can save data transmission time and reduce response latency. This benefit is critical for applications that require predictability and low latency. In addition, edge computing can make decisions and diagnoses from the network edge, where events occur faster than in the cloud center. Finally, the natural advantage of edge computing is location awareness. Some geo-based applications like transportation can collect and analyze data to avoid the dilemma of transmission to the cloud [171].

8.2 Smart Healthcare

The ECDriven-IoT can make a big difference in smart healthcare, where IoT is widely adopted. Wearable low-power IoT medical sensors for monitoring health-related data and tracking records are now popular in public healthcare facilities [73]. Embedding sensors and actuators on patients are to help doctors monitor patients' health status and provide feedback to healthcare providers. However, performance without edge computing is not good enough in terms of latency and accuracy [172]. Remote patient monitoring is a typical use case in smart healthcare. It provides convenience for doctors and patients far away from medical facilities. The records of patients have to be processed immediately and securely. Thus, transmission latency can be the bottleneck that prevents smart healthcare from being applied widely. With the potential benefits, the role of edge computing in the health and social assistance industries becomes more evident. Much research on the employment of the ECDriven-IoT in health care has been investigated. In Reference [176], a remote patient health monitoring scheme was proposed in smart homes via the concept of edge computing at the gateways. This monitoring system adopts advanced techniques at the edge of the network. These techniques involve data mining, distributed storage, and notification services. Rahmani et al. introduced the smart gateway concept and explored its application in remote health monitoring [143]. The medical data generated in edge nodes will be collected and processed to update the monitoring system's parameters. Due to the geo-distributed nature of the network, the system can provide real-time notification for patients and privacy in data gathering.

Gia et al. introduced a medical application of the ECDriven-IoT [60]. The system utilizes edge computing on the intelligent gateway to enhance the health monitoring system. The specific measures are data mining at the network edge, and distributed storage to enhance notification services. Stanciu et al. used blockchain technology as a starting point and integrated blockchain technology into an edge computing platform to implement a distributed control system [167].

8.3 Mobile VR and AR

The progress of smartphones and smart glasses has increased the popularity of augmented reality applications [192]. With the development of **virtual reality (VR)** and **augmented reality (AR)**,

humans can interact more naturally with the virtual world through the data that are collected by IoT devices [156]. With IoT sensors, AR technologies can extend the real world to the virtual world [18]. In the initial step, cloud computing provides the demands of computational power, which can satisfy these requirements in latency and quality. However, VR and AR can be applied to more scenes, such as tourism, smart transportation networks, and robotic-assisted surgeries. Cloud computing is no longer used to satisfy the requirements in latency and throughout the network, as these scenes are strict in latency, which may decrease user satisfaction. For low-latency offloading services in VR and AR, edge computing can effectively reduce the latency in combining these processed data with physical reality.

Edge computing can also migrate computing tasks from mobile devices to edge nodes to increase the computational capacity of VR devices, save battery life, and reduce latency at the same time [35]. Additionally, edge computing can be connected with the cloud for stronger computing capabilities when needed [19].

Zao et al. proposed an architecture that combines edge nodes and cloud data centers to leverage the augmented brain-computer interface [196]. The main benefit of this architecture is the low latency and real-time interaction, which can provide a more comfortable playing experience in VR and AR application scenes.

8.4 Industry Application

The **Industry Internet of Things (IIoT)** is known as Industry 4.0, which means the new era in the industry area [95]. IIoT incorporates numerous advanced communication and automation technologies, AI, and big data analysis to improve intelligence and connectivity in industry [188]. Today, new intelligent technologies are applied to accelerate the innovation and transformation of the factory workforce. IoT can collect data in extreme scenes to protect workers from danger. Furthermore, these collected data can be stored and analyzed to make better decisions.

IIoT provides many benefits, such as improving operational efficiency, connectivity, and scalability, and saving the time cost for manufacturing processes [133]. Combined with many smart machines, IIoT aims for higher accuracy, greater efficiency, and more constant working capabilities than humans. As a complement to IoT, edge computing can play a very important role in IIoT. For instance, real-time edge analysis and enhanced edge security are the two main ECDriven-IoT application scenes. Additionally, edge computing can provide an opportunity to address shortcomings in the IIoT domain [123].

Edge computing can optimize the performance of traditional IIoT. Instead of transmitting the sensor data to the cloud directly, edge computing can process those data in edge nodes to reduce the data volume and bandwidth. Processing sensor data in the edge nodes can also reduce the latency and preserve the storage in the cloud, improving the service quality of many applications, including video streaming [85]. Harper et al. proposed a fog-computing-based communication architecture that will substantially minimize the energy consumption of the IoT nodes [68]. Edge computational capabilities are further used to predict future data measurements and reduce the throughput from IoT devices to control units.

8.5 System Evolution

Sarkar et al. showed the parameters and features of edge computing by mathematical calculations [155]. Their research analyzed and compared the power consumption, service delay, carbon dioxide emissions, and cost of edge computing and cloud computing. Villari et al. introduced a new concept, osmotic computing [177], an emerging calculation paradigm similar to edge computing. It also supports data processing at the network edge while providing IoT services. The author also

discussed some of its characteristics and future directions. Morabito et al. showed how to enhance edge computing with lightweight virtualization in the IoT [115].

Dastjerdi et al. presented an introduction to the concepts and characteristics of fog computing. They also analyzed what a complete edge computing software system looks like, including the system design patterns, API, and service management [42]. In their research, Gupta et al. started from a software perspective and first proposed several challenges to be solved in implementing the edge and IoT paradigms [65]. The most critical challenge is resource management technology. In other words, how to determine which application modules are deployed in the edge device to minimize latency and ensure adequate throughput. At the same time, network congestion and energy costs also need to be considered in the future.

9 LESSONS LEARNED, OPEN CHALLENGES, AND FUTURE DIRECTIONS

9.1 Key Lessons Learned

We have illustrated many challenges ECDriven-IoT has met when applied to reality. In contrast, we also can discover many opportunities and advantages that ECDriven-IoT will eventually bring. Edge computing and IoT, when they are deployed independently, both have many shortcomings, which have prevented them from being widely used and developed. But when combined, they can help each other in bandwidth, power consumption, latency, security, and so on. We draw some lessons from our extensive survey of related work, including ECDriven-IoT architecture and standards, efficient communication, application, and security.

9.1.1 Unified Architecture and Standard. Since edge and IoT devices are heterogeneous from bottom hardware to top system design, current related research is fragmented and lacks a unified measurement for proposed ECDriven-IoT solutions. For example, multiple embedded operating systems are designed to abstract heterogeneous IoT and edge devices, but ECDriven-IoT applications cannot migrate directly between these operating systems, because they have different exposed interfaces. Establishing standards is very important for the development of a field, and the same is true for ECDriven-IoT. ECDriven-IoT should have a unified architecture or interface standard to facilitate its deployment and usage. Otherwise, the benefits of ECDriven-IoT will be largely hindered by the heterogeneous nature of IoT and edge computing. Though some existing studies have provided architecture design schemes of ECDriven-IoT, few of them are committed to pushing forward standards establishment in this area.

9.1.2 Efficient Communication and Computation Coordination. The communication latency of ECDriven-IoT applications can be further reduced owing to the decentralized fabric of edge computing. However, when it comes to different IoT applications, different communication protocols, and network conditions, how to ensure that the communication between IoT devices and edge nodes is efficient is a critical concern. Specifically, how to coordinate edge nodes to complete the computing work of IoT devices is a key optimization problem. Besides, diverse communication protocols in the IoT area make the encoding-decoding in edge nodes more complex. One solution for edge nodes to handle various protocols is supporting several communication protocols, but this solution is relatively costly. Another solution is the newly emerged **cross-technology communication (CTC)** technologies, which enable two or more different communication protocols to communicate with each other. Although many CTC technologies have been explored to make communication more efficient, they are only limited to two or three technologies [64]. Thus, how to achieve efficient communication still need more exploration.

9.1.3 Practical Security and Privacy Solutions. Security and privacy issues of ECDriven-IoT are more complicated due to the heterogeneous and distributed architecture. Current solutions cover

multiple layers of ECDriven-IoT, i.e., architecture layer, communication layer, and data layer. But these solutions are far from satisfactory because of high computation complexity or specific hardware demands. For example, blockchain-based solutions for architecture security bring extra communication overheads and storage costs; TEE-based solutions for computation security rely on trusted hardware. ECDriven-IoT should focus on exploring lightweight and the common security and privacy solutions, which are practical and efficient even in resource-constrained devices.

9.1.4 Different Designs for Different Application Scenes. The design of ECDriven-IoT cannot be illustrated simply by a single model, and the system requirements vary for different application scenarios. The ECDriven-IoT system design should be flexibly adapted to make it more efficient and suitable for various ECDriven-IoT applications. For example, we may be more concerned with the computational and communication complexity of real-time applications (e.g., smart health, autonomous driving, VR) but more concerned with the power consumption of long-life protection applications (e.g., field environmental monitoring). It is a trade-off in the design of ECDriven-IoT systems. Therefore, when designing an ECDriven-IoT system, we should fully consider what metrics the application really cares about and give a suitable application-specific solution.

9.2 Challenges and Future Directions

There are many challenges to be solved in edge computing, especially related to IoT. In this section, we will discuss some of the open research challenges and potential future work in the ECDriven-IoT.

9.2.1 Heterogeneous Platforms in Edge Computing and IoT. In a traditional cloud computing data center, users do not need to know how the program works or care about the underlying hardware architecture. However, in the ECDriven-IoT, edge devices and networks need to take on computing tasks while considering heterogeneous hardware platforms. The heterogeneous nature of ECDriven-IoT leads to a significant increase in programming workload for developers. The future development of IoT relies on edge computing, and the application scenarios are rich and colorful. Effectively solving the difficulties brought by heterogeneous platforms will make more developers invest in such work.

How to discover resources and services in a distributed computing environment is an area to be explored. To make full use of the edge devices of the network, it is necessary to establish a discovery mechanism to find the appropriate nodes that can be deployed in a distributed manner. Because of the sheer number of devices available, these mechanisms cannot rely on manuals. In addition, various heterogeneous devices are needed to meet the latest computing needs, such as large-scale machine learning tasks. These mechanisms must seamlessly integrate computational workflows at different levels without increasing latency or compromising the user experience. The original cloud-based methods are no longer applicable in edge computing.

9.2.2 Task Allocation in the ECDriven-IoT. The biggest challenge for the ECDriven-IoT is how to deploy large-scale computing and storage capabilities dynamically [20]. The appropriate deployment will make device sides work together efficiently and seamlessly. Evolving distributed computing has spawned many technologies that are used to facilitate the task of partitioning in multiple geographies. However, on the edge side, partitioned computing not only poses the challenge of efficient partitioning but also encounters bottlenecks in automatic allocation without the capacity or location of edge nodes. Therefore, a new scheduling strategy is needed to assign tasks to edge nodes. It is a prominent issue that must be addressed for large-scale deployments of IoT edge devices and networks and will affect the scale of the development of the ECDriven-IoT.

9.2.3 Data Abstraction in Edge Computing and IoT. Although data-generation devices in the IoT do not need to send generated data to the data center frequently, the edge node needs some required data to perform analysis work. Data abstraction refers to these data pre-processing algorithms and solutions, including noise cancellation, data classification, and computing. For example, gateways need to process some events such as noise cancellation. However, IoT devices are rich and varied, and different devices use different data formats. It brings the first challenge to data abstraction. The second challenge is how to effectively determine the level of data abstraction. Considering data security issues, the application does not get all the raw data, but only abstracts the parts it is interested in. If too little of the raw data is filtered, then the application will not get the information it needs. However, keeping too much raw data can cause storage problems. In addition, the data generated by edge devices are often unreliable due to external interference from sensors. Therefore, extracting accurate information from unreliable raw data is another challenge.

The application needs to control objects to provide a specific service, such as reading and writing data. The data abstraction layer combines the presentation of data and corresponding operations and provides a unified interface. In addition, finding a universal way of data abstraction is not easy because of the diversity of devices, different ways of presenting data, and different corresponding operations.

9.2.4 Edge Nodes Security. The security of the ECDriven-IoT requires end-to-end protection. As the device is closer to IoT, the difficulties in network edge-side access control and threat protection will increase dramatically. Edge-side security mainly includes device security, network security, data security, and application security. In addition, the confidentiality of critical data and the protection of personal privacy data are vital areas of IoT security [149].

Several issues must be addressed before end devices (e.g., switches, base stations) are used as edge nodes for shared access. First, risks associated with users and owners of edge devices need to be defined. Second, when the device is used as an edge computing node, the original functionality of the device cannot be compromised. Third, multiple users on edge nodes need security as their primary concern. Fourth, the minimum service level needs to be guaranteed to the users of edge nodes. Finally, workloads, computing power, data locations and migration, maintenance costs, and energy consumption need to establish an appropriate pricing model.

9.2.5 Development Tools for Edge-computing-driven IoT. As the number of edge nodes supporting general-purpose computing continues to increase, the demand for development frameworks and toolkits will continue to grow. Edge analysis is different from existing work. Since edge analysis is implemented in user-driven applications, existing tools may not be suitable for expressing edge analysis workflows. The programming model needs to use edge nodes to support the parallelism of tasks and perform calculations on multiple levels of hardware. At the same time, programming languages need to consider the hardware heterogeneity in the workflow and the computing power of various resources. So, ECDriven-IoT is more complicated than existing cloud computing models.

10 CONCLUSION

In this article, we present a comprehensive survey of the ECDriven-IoT, including supporting technologies and research challenges in this field. We first categorize existing studies to help researchers find innovative research topics. We then propose some open issues worthy of study and contribute to the development of the industry. Currently, the research on the ECDriven-IoT topic is still highly fragmented, which is not conducive to the research and development of the field. Therefore, this survey helps review and summarize existing research work and promote cross-cooperation in related areas.

REFERENCES

[1] Mohammad Aazam and Eui-Nam Huh. 2014. Fog computing and smart gateway-based communication for cloud of things. In *Proceedings of the International Conference on Future Internet of Things and Cloud*. IEEE, 464–470.

[2] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. 2018. Mobile edge computing: A survey. *IEEE Internet Things J.* 5, 1 (2018), 450–465.

[3] Stefano Abbate, Marco Avvenuti, Daniel Cesarini, and Alessio Vecchio. 2012. Estimation of energy consumption for TinyOS 2.x-based applications. *Procedia Comput. Sci.* 10 (2012), 1166–1171.

[4] Utku Günay Acer, Aidan Boran, Claudio Forlivesi, Werner Liekens, Fernando Pérez-cruz, and Fahim Kawsar. 2015. Sensing WiFi network for personal IoT analytics. In *Proceedings of the International Conference on the Internet of Things (IOT'15)*. IEEE, 104–111.

[5] Ansuman Adhikary, Xingqin Lin, and Y.-P. Eric Wang. 2016. Performance evaluation of NB-IoT coverage. In *Proceedings of the IEEE Vehicular Technology Society (VTC'16)*. IEEE, 1–5.

[6] Bilal Afzal, Muhammad Umair, Ghalib Asadullah Shah, and Ejaz Ahmed. 2019. Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. *Future Gen. Comput. Syst.* 92 (2019), 718–731.

[7] Yuan Ai, Mugen Peng, and Kecheng Zhang. 2018. Edge computing technologies for internet of things: A primer. *Dig. Commun. Netw.* 4, 2 (2018), 77–86.

[8] G. R. Aiello and G. D. Rogerson. 2003. Ultra-wideband wireless systems. *IEEE Microwave Mag.* 4, 2 (2003), 36–47.

[9] K. Akkaya and M. Younis. 2003. An energy-aware QoS routing protocol for wireless sensor networks. In *Proceedings of the International Conference on Distributed Computing Systems Workshops*. IEEE, 710–715.

[10] Fadele Ayotunde Alaba, Mazliza Othman, Ibrahim Abaker Targio Hashem, and Faiz Alotaibi. 2017. Internet of things security: A survey. *J. Netw. Comput. Appl.* 88 (2017), 10–28.

[11] Thamer A. Alghamdi, Aboubaker Lasebae, and Mahdi Aiash. 2013. Security analysis of the constrained application protocol in the Internet of Things. In *Proceedings of the International Conference on Future Generation Communication Technologies*. IEEE, 163–168.

[12] Abdulmalik Alwarafy, Khaled A. Al-Thelaya, Mohamed Abdallah, Jens Schneider, and Mounir Hamdi. 2021. A survey on security and privacy issues in edge-computing-assisted internet of things. *IEEE Internet Things J.* 8, 6 (2021), 4004–4022.

[13] Mahdi Amiri-Kordestani and Hadj Bourdoucen. 2017. A survey on embedded open source system software for the internet of things. In *Proceedings of the Free and Open Source Software Conference*. 6.

[14] Jeffrey G. Andrews, Stefano Buzzi, Wan Choi, Stephen V. Hanly, Angel Lozano, Anthony C. K. Soong, and Jianzhong Charlie Zhang. 2014. What will 5G be? *IEEE J. Select. Areas Commun.* 32, 6 (2014), 1065–1082.

[15] Luigi Atzori, Antonio Iera, and Giacomo Morabito. 2010. The internet of things: A survey. *Comput. Netw.* 54, 15 (2010), 2787–2805.

[16] Israr Iqbal Awan, Nadir Shah, Muhammad Imran, Muhammad Shoaib, and Nasir Saeed. 2019. An improved mechanism for flow rule installation in-band SDN. *J. Syst. Architect.* 96 (2019), 1–19.

[17] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy. 2017. How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Commun. Surveys Tutor.* 19, 4 (2017), 2359–2391.

[18] Luciano Baresi, Danilo Filgueira Mendonça, and Martin Garriga. 2017. Empowering low-latency applications through a serverless edge computing architecture. In *Service-Oriented and Cloud Computing*. Springer International Publishing, Cham, 196–210.

[19] Ejder Bastug, Mehdi Bennis, Muriel Medard, and Merouane Debbah. 2017. Toward interconnected virtual reality: Opportunities, challenges, and enablers. *IEEE Commun. Mag.* 55, 6 (2017), 110–117.

[20] Michael Till Beck, Martin Werner, Sebastian Feld, and S. Schimper. 2014. Mobile edge computing: A taxonomy. In *Proceedings of the International Conference on Advances in Future Internet*. Citeseer, 48–55.

[21] A. Bekasiewicz and S. Koziel. 2016. Compact UWB monopole antenna for internet of things applications. *Electron. Lett.* 52, 7 (2016), 492–494.

[22] Yihenew Dagne Beyene, Riku Jantti, Olav Tirkkonen, Kalle Ruttik, Sassan Iraji, Anna Larmo, Tuomas Tirronen, and Johan Torsner. 2017. NB-IoT technology overview and experience from cloud-RAN implementation. *IEEE Wireless Commun.* 24, 3 (2017), 26–32.

[23] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. 2014. Fog computing: A platform for internet of things and analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer International Publishing, Cham, 169–186.

[24] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the 1st MCC Workshop on Mobile Cloud Computing*. ACM, New York, NY, 13–16.

[25] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. 2016. Do LoRa low-power wide-area networks scale? In *Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, New York, NY, 59–67.

[26] Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescapé. 2016. Integration of cloud computing and internet of things: A survey. *Future Gen. Comput. Syst.* 56 (2016), 684–700.
[27] Fangbo Cai, Nafei Zhu, Jingsha He, Pengyu Mu, Wenxin Li, and Yi Yu. 2019. Survey of access control models and technologies for cloud computing. *Cluster Comput.* 22, 3 (2019), 6111–6122.
[28] Qing Cao, Tarek Abdelzaher, John Stankovic, and Tian He. 2008. The LiteOS operating system: Towards unix-like abstractions for wireless sensor networks. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'08)*. IEEE, 233–244.
[29] Juan José Martínez Castillo and Karina Aviles Rodriguez. 2012. Security architecture for Ad hoc NOMOHi networks: Development of a project based on emergency rural telecommunications. In *Proceedings of the World Congress on Internet Security (WorldCIS'12)*. IEEE, 183–187.
[30] Korhan Cengiz and Tamer Dag. 2015. A review on the recent energy-efficient approaches for the internet protocol stack. *EURASIP J. Wireless Commun. Netw.* 2015, 1 (2015), 1–22.
[31] Hojung Cha, Sukwon Choi, Inuk Jung, Hyoseung Kim, Hyojeong Shin, Jaehyun Yoo, and Chanmin Yoon. 2007. RETOS: Resilient, expandable, and threaded operating system for wireless sensor networks. In *Proceedings of the International Symposium on Information Processing in Sensor Networks*. IEEE, 148–157.
[32] Tej Bahadur Chandra, Pushpak Verma, and A. K. Dwivedi. 2016. Operating systems for internet of things: A comparative study. In *Proceedings of the International Conference on Information and Communication Technology for Competitive Strategies (ICTCS'16)*. ACM, New York, NY, Article 47, 6 pages.
[33] Hyunseok Chang, Adiseshu Hari, Sarit Mukherjee, and T. V. Lakshman. 2014. Bringing the cloud to the edge. In *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM'14)*. IEEE, 346–351.
[34] Kai Chih Chang, Razieh Nokhbeh Zaeem, and K. Suzanne Barber. 2018. Enhancing and evaluating identity privacy and authentication strength by utilizing the identity ecosystem. In *Proceedings of the Workshop on Privacy in the Electronic Society*. ACM, New York, NY, 114–120.
[35] Mingzhe Chen, Walid Saad, and Changchuan Yin. 2018. Virtual reality over wireless networks: Quality-of-service model and learning-based resource management. *IEEE Trans. Commun.* 66, 11 (2018), 5621–5635.
[36] Songlin Chen, Yixin Jiang, Hong Wen, Wenjie Liu, Jie Chen, Wenxin Lei, and Aidong Xu. 2018. A novel terminal security access method based on edge computing for IoT. In *Proceedings of the International Conference on Networking and Network Applications (NaNA'18)*. IEEE, 394–398.
[37] Xu Chen. 2014. Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. Parallel Distrib. Syst.* 26, 4 (2014), 974–983.
[38] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. 2015. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Trans. Netw.* 24, 5 (2015), 2795–2808.
[39] Zhipeng Cheng, Minghui Min, Zhibin Gao, and Lianfen Huang. 2020. Joint task offloading and resource allocation for mobile edge computing in ultra-dense network. In *Proceedings of the IEEE Global Communications Conference*. IEEE, 1–6.
[40] Shao-Yi Chien, Wei-Kai Chan, Yu-Hsiang Tseng, Chia-Han Lee, V. Srinivasa Somayazulu, and Yen-Kuang Chen. 2015. Distributed computing in IoT: System-on-a-chip for smart cameras as an example. In *Proceedings of the 20th Asia and South Pacific Design Automation Conference*. IEEE, 130–135.
[41] Supratim Das, Amarjeet Singh, Surinder Pal Singh, and Amit Kumar. 2015. A low overhead dynamic memory management system for constrained memory embedded systems. In *Proceedings of the International Conference on Computing for Sustainable Global Development (INDIACom'15)*. IEEE, 809–815.
[42] A. Dastjerdi and R. Buyya. 2016. Fog computing: Helping the internet of things realize its potential. *Computer* 49, 8 (2016), 112–116.
[43] Rustem Dautov and Salvatore Distefano. 2017. Three-level hierarchical data fusion through the IoT, edge, and cloud computing. In *Proceedings of the International Conference on Internet of Things and Machine Learning*. ACM, New York, NY, Article 1, 5 pages.
[44] Josep Domingo-Ferrer, Oriol Farrás, Jordi Ribes-González, and David Sánchez. 2019. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. *Comput. Commun.* 140–141 (2019), 38–60.
[45] Jianbo Du, Liqiang Zhao, Jie Feng, Xiaoli Chu, and F. Richard Yu. 2018. Economical revenue maximization in cache enhanced mobile edge computing. In *Proceedings of the IEEE International Conference on Communications (ICC'18)*. IEEE, 1–6.
[46] Adam Dunkels. 2007. Rime-a lightweight layered communication stack for sensor networks. In *Proceedings of the European Conference on Wireless Sensor Networks*, Vol. 44. Citeseer, 2.
[47] EdgeX. 2018. EdgeX Foundry. Retrieved from https://www.edgexfoundry.org/get-started/.
[48] Hesham El-Sayed, Sharmi Sankar, Mukesh Prasad, Deepak Puthal, Akshansh Gupta, Manoranjan Mohanty, and Chin-Teng Lin. 2018. Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access* 6 (2018), 1706–1717.

[49] Hanan Elazhary. 2019. Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. *J. Netw. Comput. Applications* 128 (2019), 105–140.
[50] Sinem Coleri Ergen. 2004. ZigBee/IEEE 802.15. 4 summary. UC Berkeley (Sept. 10, 2004).
[51] Ericsson. 2010. CEO to shareholders: 50 billion connections 2020. Retrieved from https://www.ericsson.com/en/press-releases/2010/4/ceo-to-shareholders-50-billion-connections-2020.
[52] Mohamed Fahim, Brahim Ouchao, Abdeslam Jakimi, and Lahcen El Bermi. 2019. Application of a non-immersive VR, IoT based approach to help moroccan students carry out practical activities in a personal learning style. *Future Internet* 11, 1 (2019), 15.
[53] S. Farah, A. Benachenhou, G. Neveux, D. Barataud, G. Andrieu, and T. Fredon. 2015. Real-time microwave remote laboratory architecture. In *Proceedings of the European Microwave Conference (EuMC)*. IEEE, 1315–1318.
[54] Yuxiang Feng, Wenhao Wang, Yukai Weng, and Huanming Zhang. 2017. A replay-attack resistant authentication scheme for the internet of things. In *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE'17) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC'17)*, Vol. 1. IEEE, 541–547.
[55] Hiro Gabriel Cerqueira Ferreira, Edna Dias Canedo, and Rafael Timóteo de Sousa. 2013. IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino. In *Proceedings of the IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'13)*. IEEE, 53–60.
[56] EdgeX Foundry. 2021. Why EdgeX. Retrieved from https://www.edgexfoundry.org/why_edgex/why-edgex/.
[57] Khusanbek Gafurov and Tai-Myoung Chung. 2019. Comprehensive survey on internet of things, architecture, security aspects, applications, related technologies, economic perspective, and future directions. *J. Info. Process. Syst.* 15 (2019), 797–819.
[58] Padmini Gaur and Mohit P. Tahiliani. 2015. Operating systems for IoT devices: A critical survey. In *Proceedings of the IEEE Region 10 Symposium*. IEEE, 33–36.
[59] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. 2019. 5G evolution: A view on 5G cellular technology beyond 3GPP release 15. *IEEE Access* 7 (2019), 127639–127651.
[60] Tuan Nguyen Gia, Mingzhe Jiang, Amir-Mohammad Rahmani, Tomi Westerlund, Pasi Liljeberg, and Hannu Tenhunen. 2015. Fog computing in healthcare internet of things: A case study on ECG feature extraction. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. IEEE, 356–363.
[61] Dimitrios Glaroudis, Athanasios Iossifides, and Periklis Chatzimisios. 2020. Survey, comparison and research challenges of IoT application protocols for smart farming. *Comput. Netw.* 168, C (2020), 14.
[62] Le Guan, Peng Liu, Xinyu Xing, Xinyang Ge, Shengzhi Zhang, Meng Yu, and Trent Jaeger. 2017. Trustshadow: Secure execution of unmodified applications with arm trustzone. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, New York, NY, 488–501.
[63] Shaoyong Guo, Xing Hu, Song Guo, Xuesong Qiu, and Feng Qi. 2020. Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Trans. Industr. Inform.* 16, 3 (2020), 1972–1983.
[64] Xiuzhen Guo, Yuan He, Jia Zhang, and Haotian Jiang. 2019. WIDE: Physical-level CTC via digital emulation. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'19)*. IEEE, 49–60.
[65] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. 2017. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software: Pract. Exper.* 47, 9 (2017), 1275–1296.
[66] J. C. Haartsen. 2000. The bluetooth radio system. *IEEE Person. Commun.* 7, 1 (2000), 28–36.
[67] Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, and Nicolas Tsiftes. 2016. Operating systems for low-end devices in the internet of things: A survey. *IEEE Internet Things J.* 3, 5 (2016), 720–734.
[68] K. Eric Harper, Thijmen de Gooijer, Johannes O. Schmitt, and David Cox. 2016. Microdatabases for the industrial Internet. Retrieved from https://arxiv.org/abs/1601.04036.
[69] Albert F. Harris III, Vansh Khanna, Guliz Tuncay, Roy Want, and Robin Kravets. 2016. Bluetooth low energy in dense IoT environments. *IEEE Commun. Mag.* 54, 12 (2016), 30–36.
[70] Brian Hayes. 2008. Cloud computing. *Commun. ACM* 51, 7 (2008), 9–11.
[71] John L. Hennessy and David A. Patterson. 2011. *Computer Architecture: A Quantitative Approach*. Elsevier.
[72] P. S. Henry and Hui Luo. 2002. WiFi: What's next? *IEEE Commun. Mag.* 40, 12 (2002), 66–72.
[73] M. Shamim Hossain and Ghulam Muhammad. 2016. Cloud-assisted industrial internet of things (IIoT)—Enabled framework for health monitoring. *Comput. Netw.* 101 (2016), 192–202.
[74] Farhoud Hosseinpour, Payam Amoli, Juha Plosila, Timo Hämäläinen, and Hannu Tenhunen. 2016. An intrusion detection system for Fog computing and IoT based logistic systems using a smart data approach. *Int. J. Dig. Content Technol. Appl.* 10 (12 2016), 34–46.

[75] Junqin Huang, Linghe Kong, Guihai Chen, Min-You Wu, Xue Liu, and Peng Zeng. 2019. Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism. *IEEE Trans. Industr. Inform.* 15, 6 (2019), 3680–3689.

[76] Jonathan Hui and Pascal Thubert. 2011. *Compression format for IPv6 datagrams over IEEE 802.15. 4-based networks.* Technical Report 6282. Internet Engineering Task Force (IETF), 1–23. Retrieved from https://www.rfc-editor.org/rfc/rfc6282.txt.

[77] Yaser Jararweh, Mahmoud Al-Ayyoub, Elhadj Benkhelifa, Mladen Vouk, Andy Rindos, et al. 2015. SDIoT: A software defined based internet of things framework. *J. Ambient Intell. Human. Comput.* 6, 4 (2015), 453–461.

[78] Yaser Jararweh, Mohammad Alsmirat, Mahmoud Al-Ayyoub, Elhadj Benkhelifa, Ala' Darabseh, Brij Gupta, and Ahmad Doulat. 2017. Software-defined system support for enabling ubiquitous mobile edge computing. *Comput. J.* 60, 10 (2017), 1443–1457.

[79] Yaser Jararweh, Ahmad Doulat, Omar AlQudah, Ejaz Ahmed, Mahmoud Al-Ayyoub, and Elhadj Benkhelifa. 2016. The future of mobile cloud computing: Integrating cloudlets and mobile edge computing. In *Proceedings of the 23rd International Conference on Telecommunications (ICT'16).* IEEE, 1–5.

[80] Farhana Javed, Muhamamd Khalil Afzal, Muhammad Sharif, and Byung-Seo Kim. 2018. Internet of things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Commun. Surveys Tutor.* 20, 3 (2018), 2062–2100.

[81] Fatma Ben Jemaa, Guy Pujolle, and Michel Pariente. 2016. Cloudlet-and NFV-based carrier Wi-Fi architecture for a wider range of services. *Ann. Telecommun.* 71, 11 (2016), 617–624.

[82] Sunmi Jun, Yoohwa Kang, Jaeho Kim, and Changki Kim. 2020. Ultra-low-latency services in 5G systems: A perspective from 3GPP standards. *ETRI J.* 42, 5 (2020), 721–733.

[83] Ajay Kakkar. 2020. A survey on secure communication techniques for 5G wireless heterogeneous networks. *Info. Fusion* 62 (2020), 89–109.

[84] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. 2019. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet Things J.* 6, 3 (2019), 4660–4670.

[85] Hajime Kanzaki, Kevin Schubert, and Nicholas Bambos. 2017. Video streaming schemes for industrial IoT. In *Proceedings of the International Conference on Computer Communication and Networks (ICCCN'17).* IEEE, 1–7.

[86] Wazir Zada Khan, Ejaz Ahmed, Saqib Hakak, Ibrar Yaqoob, and Arif Ahmed. 2019. Edge computing: A survey. *Future Gen. Comput. Syst.* 97 (2019), 219–235.

[87] Song Min Kim and Tian He. 2015. FreeBee: Cross-technology communication via free side-channel. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom'15).* ACM, New York, NY, 317–330.

[88] Shinji Kitagami, Tadashi Ogino, Takuo Suganuma, and Norio Shiratori. 2017. Proposal of a multi-agent based flexible IoT edge computing architecture harmonizing its control with cloud computing. In *Proceedings of the International Symposium on Computing and Networking (CANDAR'17).* IEEE, 223–229.

[89] Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proc. IEEE* 103, 1 (2015), 14–76.

[90] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava. 2013. A survey of computation offloading for mobile systems. *Mobile Netw. Appl.* 18, 1 (2013), 129–140.

[91] Vijay Kumar, George Oikonomou, Theo Tryfonas, Dan Page, and Iain Phillips. 2014. Digital investigations for IPv6-based wireless sensor networks. *Dig. Invest.* 11, S2 (2014), S66–S75.

[92] Patrick Kurp. 2008. Green computing. *Commun. ACM* 51, 10 (2008), 11–13.

[93] O. Landsiedel, K. Wehrle, and S. Gotz. 2005. Accurate prediction of power consumption in sensor networks. In *Proceedings of the IEEE Workshop on Embedded Networked Sensors.* IEEE, 37–44.

[94] Erik G. Larsson, Ove Edfors, Fredrik Tufvesson, and Thomas L. Marzetta. 2014. Massive MIMO for next generation wireless systems. *IEEE Commun. Mag.* 52, 2 (2014), 186–195.

[95] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. Industry 4.0. *Bus. Info. Syst. Eng.* 6, 4 (2014), 239–242.

[96] Bo Li, Qiang He, Feifei Chen, Hai Jin, Yang Xiang, and Yun Yang. 2020. Auditing cache data integrity in the edge computing environment. *IEEE Trans. Parallel Distrib. Syst.* 32, 5 (2020), 1210–1223.

[97] Jiliang Li, Zhou Su, Deke Guo, Kim-Kwang Raymond Choo, Yusheng Ji, and Huayan Pu. 2021. Secure data deduplication protocol for edge-assisted mobile CrowdSensing services. *IEEE Trans. Vehic. Technol.* 70, 1 (2021), 742–753.

[98] Shichao Li, Ning Zhang, Siyu Lin, Linghe Kong, Ajay Katangur, Muhammad Khurram Khan, Minming Ni, and Gang Zhu. 2018. Joint admission control and resource allocation in edge computing for internet of things. *IEEE Netw.* 32, 1 (2018), 72–79.

[99] Fuhong Lin, Yutong Zhou, Xingsuo An, Ilsun You, and Kim-Kwang Raymond Choo. 2018. Fair resource allocation in an intrusion-detection system for edge computing: Ensuring the security of internet of things devices. *IEEE Consum. Electron. Mag.* 7, 6 (2018), 45–50.

[100] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. 2017. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet Things J.* 4, 5 (2017), 1125–1142.

[101] Dan Liu, Zheng Yan, Wenxiu Ding, and Mohammed Atiquzzaman. 2019. A survey on secure data analytics in edge computing. *IEEE Internet Things J.* 6, 3 (2019), 4946–4967.

[102] Yang Lu and Li Da Xu. 2018. Internet of things (IoT) cybersecurity research: A review of current research topics. *IEEE Internet Things J.* 6, 2 (2018), 2103–2115.

[103] Xin Ma and Wei Luo. 2008. The analysis of 6LowPAN technology. In *Proceedings of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, Vol. 1. IEEE, 963–966.

[104] Pavel Mach and Zdenek Becvar. 2017. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surveys Tutor.* 19, 3 (2017), 1628–1656.

[105] Shahid Mahmood, Amin Ullah, and Anas Khalid Kayani. 2019. Fog computing trust based architecture for internet of things devices. *Int. J. Comput. Commun. Netw.* 1, 1 (2019), 18–25.

[106] Luca Mainetti, Luigi Patrono, and Antonio Vilei. 2011. Evolution of wireless sensor networks towards the internet of things: A survey. In *Proceedings of the International Conference on Software, Telecommunications and Computer Networks*. IEEE, 1–6.

[107] Nitin Mangalvedhe, Rapeepat Ratasuk, and Amitava Ghosh. 2016. NB-IoT deployment study for low power wide area cellular IoT. In *Proceedings of the IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'16)*. IEEE, 1–6.

[108] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surveys Tutor.* 19, 4 (2017), 2322–2358.

[109] Yuyi Mao, Jun Zhang, and Khaled B. Letaief. 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Select. Areas Commun.* 34, 12 (2016), 3590–3605.

[110] Diego Mendez Mena, Ioannis Papapanagiotou, and Baijian Yang. 2018. Internet of things: Survey on security. *Info. Secur. J.: Global Perspect.* 27, 3 (2018), 162–182.

[111] Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Niels Bouten, Filip De Turck, and Raouf Boutaba. 2016. Network function virtualization: State-of-the-art and research challenges. *IEEE Commun. Surveys Tutor.* 18, 1 (2016), 236–262.

[112] Takuho Mitsunaga, Yoshifumi Manabe, and Tatsuaki Okamoto. 2010. Efficient secure auction protocols based on the Boneh-Goh-Nissim encryption. In *Advances in Information and Computer Security*. Springer, Berlin, 149–163.

[113] Muhammad Baqer Mollah, Md. Abul Kalam Azad, and Athanasios Vasilakos. 2017. Secure data sharing and searching at the edge of cloud-assisted internet of things. *IEEE Cloud Comput.* 4, 1 (2017), 34–42.

[114] N. Montavont and T. Noel. 2002. Handover management for mobile nodes in IPv6 networks. *IEEE Commun. Mag.* 40, 8 (2002), 38–43.

[115] Roberto Morabito, Vittorio Cozzolino, Aaron Yi Ding, Nicklas Beijar, and Jorg Ott. 2018. Consolidate IoT edge computing with lightweight virtualization. *IEEE Netw.* 32, 1 (2018), 102–111.

[116] Vladimir Moravcevic, Milan Tucic, Roman Pavlovic, and Aleksandar Majdak. 2015. An approach for uniform representation and control of ZigBee devices in home automation software. In *Proceedings of the IEEE 5th International Conference on Consumer Electronics (ICCE'15)*. IEEE, 237–239.

[117] Arsalan Mosenia and Niraj K. Jha. 2017. A comprehensive study of security of internet-of-things. *IEEE Trans. Emerg. Top. Comput.* 5, 4 (2017), 586–602.

[118] Carla Mouradian, Diala Naboulsi, Sami Yangui, Roch H. Glitho, Monique J. Morrow, and Paul A. Polakos. 2018. A comprehensive survey on Fog computing: State-of-the-art and research challenges. *IEEE Commun. Surveys Tutor.* 20, 1 (2018), 416–464.

[119] Geoff Mulligan. 2007. The 6LoWPAN architecture. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*. ACM, New York, NY, 78–82.

[120] Arslan Munir, Prasanna Kansakar, and Samee U. Khan. 2017. IFCIoT: Integrated Fog cloud IoT: A novel architectural paradigm for the future Internet of Things. *IEEE Consum. Electron. Mag.* 6, 3 (2017), 74–82.

[121] Arslan Musaddiq, Yousaf Bin Zikria, Oliver Hahm, Heejung Yu, Ali Kashif Bashir, and Sung Won Kim. 2018. A survey on resource management in IoT operating systems. *IEEE Access* 6 (2018), 8459–8482.

[122] Karan Nair, Janhavi Kulkarni, Mansi Warde, Zalak Dave, Vedashree Rawalgaonkar, Ganesh Gore, and Jonathan Joshi. 2015. Optimizing power consumption in IoT based wireless sensor networks using bluetooth low energy. In *Proceedings of the International Conference on Green Computing and Internet of Things (ICGCIoT'15)*. IEEE, 589–593.

[123] Rick Nelson. 2017. Smart factories leverage cloud, edge computing. *Eval. Eng.* 56, 6 (2017), 14.

[124] Kim Thuat Nguyen, Maryline Laurent, and Nouha Oualha. 2015. Survey on secure communication protocols for the internet of things. *Ad Hoc Netw.* 32 (2015), 17–31.
[125] Dennis Kengo Oka, Takahiro Furue, Lennart Langenhop, and Tomohiro Nishimura. 2014. Survey of vehicle IoT bluetooth devices. In *Proceedings of the IEEE 7th International Conference on Service-Oriented Computing and Applications*. IEEE, 260–264.
[126] Babatunji Omoniwa, Riaz Hussain, Muhammad Awais Javed, Safdar Hussain Bouk, and Shahzad A. Malik. 2019. Fog/Edge computing-based IoT (FECIoT): Architecture, applications, and research issues. *IEEE Internet Things J.* 6, 3 (2019), 4118–4149.
[127] Jose A. Onieva, Ruben Rios, Rodrigo Roman, and Javier Lopez. 2019. Edge-assisted vehicular networks security. *IEEE Internet Things J.* 6, 5 (2019), 8038–8045.
[128] Pouya Ostovari, Abdallah Khreishah, and Jie Wu. 2013. Cache content placement using triangular network coding. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'13)*. IEEE, 1375–1380.
[129] Michael O'Keeffe. 2008. The paillier cryptosystem. *Math. Dept.* 18 (Apr. 2008), 1–16.
[130] Jianli Pan and James McElhannon. 2018. Future edge cloud and edge computing for internet of things applications. *IEEE Internet Things J.* 5, 1 (2018), 439–449.
[131] Heejin Park, Shuang Zhai, Long Lu, and Felix Xiaozhu Lin. 2019. StreamBox-TZ: Secure stream analytics at the edge with TrustZone. In *Proceedings of the USENIX Annual Technical Conference (ATC'19)*. 537–554.
[132] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. 2018. PLoRa: A passive long-range data network from ambient LoRa transmissions. In *Proceedings of the ACM Special Interest Group on Data Communications (SIGCOMM'18)*. ACM, New York, NY, 147–160.
[133] Charith Perera, Chi Harold Liu, Srimal Jayawardena, and Min Chen. 2014. A survey on internet of things from industrial market perspective. *IEEE Access* 2 (2014), 1660–1679.
[134] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context aware computing for the internet of things: A survey. *IEEE Commun. Surveys Tutor.* 16, 1 (2014), 414–454.
[135] Tara Petrić, Mathieu Goessens, Loutfi Nuaymi, Laurent Toutain, and Alexander Pelov. 2016. Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN. In *Proceedings of the IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'16)*. IEEE, 1–7.
[136] Vitaly Petrov, Andrey Samuylov, Vyacheslav Begishev, Dmitri Moltchanov, Sergey Andreev, Konstantin Samouylov, and Yevgeni Koucheryavy. 2018. Vehicle-based relay assistance for opportunistic crowdsensing over narrowband IoT (NB-IoT). *IEEE Internet Things J.* 5, 5 (2018), 3710–3723.
[137] Juha Petäjäjärvi, Konstantin Mikhaylov, Marko Pettissalo, Janne Janhunen, and Jari Iinatti. 2017. Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage. *Int. J. Distrib. Sensor Netw.* 13, 3 (2017), 1550147717699412.
[138] Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. 2018. Survey on multi-access edge computing for internet of things realization. *IEEE Commun. Surveys Tutor.* 20, 4 (2018), 2961–2991.
[139] Zane D. Purvis and Alexander G. Dean. 2008. TOSSTI: Saving time and energy in TinyOS with software thread integration. In *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium*. IEEE, 354–363.
[140] Zhijing Qin, Grit Denker, Carlo Giannelli, Paolo Bellavista, and Nalini Venkatasubramanian. 2014. A software defined networking architecture for the internet-of-things. In *Proceedings of the IEEE Network Operations and Management Symposium (NOMS'14)*. IEEE, 1–9.
[141] Meikang Qiu, Sun-Yuan Kung, and Keke Gai. 2020. Intelligent security and optimization in Edge/Fog Computing. *Future Generation Computer Systems* 107 (2020), 1140–1142. DOI: https://doi.org/10.1016/j.future.2019.06.002
[142] Wajid Rafique, Lianyong Qi, Ibrar Yaqoob, Muhammad Imran, Raihan Ur Rasool, and Wanchun Dou. 2020. Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Commun. Surveys Tutor.* 22, 3 (2020), 1761–1804.
[143] Amir M. Rahmani, Tuan Nguyen Gia, Behailu Negash, Arman Anzanpour, Iman Azimi, Mingzhe Jiang, and Pasi Liljeberg. 2018. Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A Fog computing approach. *Future Gen. Comput. Syst.* 78 (2018), 641–658.
[144] Muhammad Raheel Raza, Asaf Varol, and Nurhayat Varol. 2020. Cloud and Fog computing: A survey to the concept and challenges. In *Proceedings of the International Symposium on Digital Forensics and Security (ISDFS'20)*. IEEE, 1–6.
[145] Biljana L. Risteska Stojkoska and Kire V. Trivodaliev. 2017. A review of internet of things for smart home: Challenges and solutions. *J. Cleaner Prod.* 140 (2017), 1454–1464.
[146] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. 2018. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Gen. Comput. Syst.* 78 (2018), 680–698.

[147] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. 2017. IoT goes nuclear: Creating a ZigBee chain reaction. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'17)*. IEEE, 195–212.
[148] Dario Sabella, Alessandro Vaillant, Pekka Kuure, Uwe Rauschenbach, and Fabio Giust. 2016. Mobile-edge computing architecture: The role of MEC in the internet of things. *IEEE Consum. Electron. Mag.* 5, 4 (2016), 84–91.
[149] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. 2015. Security and privacy challenges in industrial internet of things. In *Proceedings of the ACM/EDAC/IEEE Design Automation Conference (DAC'15)*. IEEE, 1–6.
[150] Yuvraj Sahni, Jiannong Cao, and Lei Yang. 2019. Data-aware task allocation for achieving low latency in collaborative edge computing. *IEEE Internet Things J.* 6, 2 (2019), 3512–3524.
[151] Ahsan Saleem, Abid Khan, Saif Ur Rehman Malik, Haris Pervaiz, Hassan Malik, Masoom Alam, and Anish Jindal. 2020. FESDA: Fog-enabled secure data aggregation in smart grid IoT network. *IEEE Internet Things J.* 7, 7 (2020), 6132–6142.
[152] Ola Salman, Imad Elhajj, Ali Chehab, and Ayman Kayssi. 2018. IoT survey: An SDN and Fog computing perspective. *Comput. Netw.* 143 (2018), 221–246.
[153] Ola Salman, Imad Elhajj, Ayman Kayssi, and Ali Chehab. 2015. Edge computing enabling the internet of things. In *Proceedings of the IEEE 2nd World Forum on Internet of Things (WF-IoT'15)*. IEEE, 603–608.
[154] Zihao Sang, Songtao Guo, Quyuan Wang, and Ying Wang. 2021. GCS: Collaborative video cache management strategy in multi-access edge computing. *Ad Hoc Netw.* 117 (2021), 102516.
[155] Subhadeep Sarkar, Subarna Chatterjee, and Sudip Misra. 2018. Assessment of the suitability of Fog computing in the context of internet of things. *IEEE Trans. Cloud Comput.* 6, 1 (2018), 46–59.
[156] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. 2015. Edge analytics in the internet of things. *IEEE Pervas. Comput.* 14, 2 (2015), 24–31.
[157] Pallavi Sethi and Smruti R. Sarangi. 2017. Internet of things: Architectures, protocols, and applications. *J. Electric. Comput. Eng.* 2017 (2017), 9324035.
[158] Mansoor Shafi, Andreas F. Molisch, Peter J. Smith, Thomas Haustein, Peiying Zhu, Prasan De Silva, Fredrik Tufvesson, Anass Benjebbour, and Gerhard Wunder. 2017. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE J. Select. Areas Commun.* 35, 6 (2017), 1201–1221.
[159] Sajjad Hussain Shah and Ilyas Yaqoob. 2016. A survey: Internet of things (IOT) technologies, applications and challenges. In *Proceedings of the IEEE Smart Energy Grid Engineering (SEGE'16)*. IEEE, 381–385.
[160] Majlesi Shahrbanoo, Mehrpour Ali, and Mehran Mohsenzadeh. 2012. An approach for agile SOA development using agile principals. Retrieved from https://arxiv.org/abs/1204.0368.
[161] Cong Shi, Jian Liu, Hongbo Liu, and Yingying Chen. 2017. Smart user authentication through actuation of daily activities leveraging WiFi-enabled IoT. In *Proceedings of the ACM International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (Mobihoc'17)*. ACM, New York, NY, Article 5, 10 pages.
[162] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE Internet Things J.* 3, 5 (2016), 637–646.
[163] Kyung-Ah Shim. 2019. Universal forgery attacks on remote authentication schemes for wireless body area networks based on internet of things. *IEEE Internet Things J.* 6, 5 (2019), 9211–9212.
[164] Jose Costa Sapalo Sicato, Sushil Kumar Singh, Shailendra Rathore, and Jong Hyuk Park. 2020. A comprehensive analyses of intrusion detection system for IoT environment. *J. Info. Process. Syst.* 16, 4 (2020), 975–990.
[165] Rashmi Sharan Sinha, Yiqiao Wei, and Seung-Hoon Hwang. 2017. A survey on LPWA technology: LoRa and NB-IoT. *ICT Express* 3, 1 (2017), 14–21.
[166] S. Smys, B. Abul, and W. Haoxiang. 2020. Hybrid intrusion detection system for internet of things (IoT). *J. ISMAC* 2, 4 (2020), 190–199.
[167] Alexandru Stanciu. 2017. Blockchain based distributed control system for edge computing. In *Proceedings of the International Conference on Control Systems and Computer Science (CSCS'17)*. IEEE, 667–671.
[168] Mark W. Storer, Kevin Greenan, Darrell D. E. Long, and Ethan L. Miller. 2008. Secure data deduplication. In *Proceedings of the ACM International Workshop on Storage Security and Survivability*. ACM, New York, NY, 1–10.
[169] Hung-Min Sun, Shih-Pu Hsu, and Chien-Ming Chen. 2007. Mobile jamming attack and its countermeasure in wireless sensor networks. In *Proceedings of the International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, Vol. 1. IEEE, 457–462.
[170] Xiang Sun and Nirwan Ansari. 2016. EdgeIoT: Mobile edge computing for the internet of things. *IEEE Commun. Mag.* 54, 12 (2016), 22–29.
[171] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. 2017. Mobile edge computing potential in making cities smarter. *IEEE Commun. Mag.* 55, 3 (2017), 38–43.
[172] Tuyen X. Tran, Abolfazl Hajisami, Parul Pandey, and Dario Pompili. 2017. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Commun. Mag.* 55, 4 (2017), 54–61.

[173] Fan-Hsun Tseng, Li-Der Chou, and Han-Chieh Chao. 2011. A survey of black hole attacks in wireless mobile ad hoc networks. *Hum.-centric Comput. Info. Sci.* 1, 1 (2011), 4.

[174] Carlo Vallati, Antonio Virdis, Enzo Mingozzi, and Giovanni Stea. 2016. Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications. *IEEE Consum. Electr. Mag.* 5, 4 (2016), 77–83.

[175] M. Vellanki, S. P. R. Kandukuri, and A. Razaque. 2016. Node level energy efficiency protocol for internet of things. *J. Theoret. Comput. Sci.* 3 (2016), 5.

[176] Prabal Verma and Sandeep K. Sood. 2018. Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet Things J.* 5, 3 (2018), 1789–1796.

[177] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, and Rajiv Ranjan. 2016. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Comput.* 3, 6 (2016), 76–83.

[178] G. V. Vivek and M. P. Sunil. 2015. Enabling IOT services using WIFI-ZigBee gateway for a home automation system. In *Proceedings of the IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN'15)*. IEEE, 77–80.

[179] Haoqin Wang, Zhen Chen, Guanping Xiao, and Zheng Zheng. 2016. Network of networks in Linux operating system. *Physica A: Stat. Mech. Appl.* 447 (2016), 520–526.

[180] Tian Wang, Guangxue Zhang, Anfeng Liu, Md Zakirul Alam Bhuiyan, and Qun Jin. 2019. A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing. *IEEE Internet Things J.* 6, 3 (2019), 4831–4843.

[181] Xiaoding Wang, Sahil Garg, Hui Lin, Georges Kaddoum, Jia Hu, and M. Shamim Hossain. 2020. A secure data aggregation strategy in edge computing and blockchain empowered internet of things. *IEEE Internet Things J.* (2020), 1–1.

[182] Mohammad Wazid, Poonam Reshma Dsouza, Ashok Kumar Das, Vivekananda Bhat K. Neeraj Kumar, and Joel J. P. C. Rodrigues. 2019. RAD-EI: A routing attack detection scheme for edge-based internet of things environment. *Int. J. Commun. Syst.* 32, 15 (2019), e4024.

[183] Hua Wei, Hong Luo, Yan Sun, and Mohammad S. Obaidat. 2020. Cache-aware computation offloading in IoT systems. *IEEE Syst. J.* 14, 1 (2020), 61–72.

[184] Juan Wen, Kaibin Huang, Sheng Yang, and Victor O. K. Li. 2017. Cache-enabled heterogeneous cellular networks: Optimal tier-level content placement. *IEEE Trans. Wireless Commun.* 16, 9 (2017), 5939–5952.

[185] Bernd W. Wirtz, Jan C. Weyerer, and Franziska T. Schichtel. 2019. An integrative public IoT framework for smart government. *Govern. Info. Quart.* 36, 2 (2019), 333–345.

[186] Junjuan Xia, Chao Li, Xiazhi Lai, Shiwei Lai, Fusheng Zhu, Dan Deng, and Liseng Fan. 2020. Cache-aided mobile edge computing for B5G wireless communication networks. *EURASIP J. Wireless Commun. Netw.* 2020, 1 (2020), 1–10.

[187] Chenhan Xu, Kun Wang, Peng Li, Song Guo, Jiangtao Luo, Baoliu Ye, and Minyi Guo. 2018. Making big data open in edges: A resource-efficient blockchain-based approach. *IEEE Trans. Parallel Distrib. Syst.* 30, 4 (2018), 870–882.

[188] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of things in industries: A survey. *IEEE Trans. Industr. Inform.* 10, 4 (2014), 2233–2243.

[189] Xiaolong Xu, Qingxiang Liu, Yun Luo, Kai Peng, Xuyun Zhang, Shunmei Meng, and Lianyong Qi. 2019. A computation offloading method over big data for IoT-enabled cloud-edge computing. *Future Gen. Comput. Syst.* 95 (2019), 522–533.

[190] Zhiwei Xu, Lu Chao, and Xiaohui Peng. 2019. T-REST: An open-enabled architectural style for the internet of things. *IEEE Internet Things J.* 6, 3 (2019), 4019–4034.

[191] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. 2018. Blockchain Technology Overview. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD. DOI: https://doi.org/10.6028/NIST.IR.8202

[192] Shanhe Yi, Cheng Li, and Qun Li. 2015. A survey of Fog computing: Concepts, applications and issues. In *Proceedings of the Workshop on Mobile Big Data*. ACM, New York, NY, 37–42.

[193] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. 2016. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Trans. Wireless Commun.* 16, 3 (2016), 1397–1411.

[194] Wei Yu, Fan Liang, Xiaofei He, William Grant Hatcher, Chao Lu, Jie Lin, and Xinyu Yang. 2018. A survey on the edge computing for the internet of things. *IEEE Access* 6 (2018), 6900–6919.

[195] Faisal A. Zaman, Abdallah Jarray, and Ahmed Karmouch. 2019. Software defined network-based edge cloud resource allocation framework. *IEEE Access* 7 (2019), 10672–10690.

[196] John K. Zao, Tchin Tze Gan, Chun Kai You, Sergio José Rodríguez Méndez, Cheng En Chung, Yu Te Wang, Tim Mullen, and Tzyy Ping Jung. 2014. Augmented brain computer interaction based on Fog computing and linked data. In *Proceedings of the International Conference on Intelligent Environments*. IEEE, 374–377.

[197] Jiale Zhang, Bing Chen, Yanchao Zhao, Xiang Cheng, and Feng Hu. 2018. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. *IEEE Access* 6 (2018), 18209–18237.

[198] Ma Zhaofeng, Wang Xiaochang, Deepak Kumar Jain, Haneef Khan, Gao Hongmin, and Wang Zhen. 2019. A blockchain-based trusted data management scheme in edge computing. *IEEE Trans. Industr. Inform.* 16, 3 (2019), 2013–2021.

[199] Jiehan Zhou, Teemu Leppanen, Erkki Harjula, Mika Ylianttila, Timo Ojala, Chen Yu, Hai Jin, and Laurence Tianruo Yang. 2013. CloudThings: A common architecture for integrating the internet of things with cloud computing. In *Proceedings of the IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD'13)*. IEEE, 651–657.

[200] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. ZiFi: Wireless LAN discovery via ZigBee interference signatures. In *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom'10)*. ACM, New York, NY, 49–60.

Received 2 May 2021; revised 22 July 2022; accepted 2 August 2022