Edge Computing for Industrial IoT: Challenges and Solutions

Erkki Harjula, Alexander Artemenko, and Stefan Forsström

Abstract The evolution from local towards virtualized data storage, computing, applications and services - in the forms of Internet of Things (IoT), Everything as a Service (EaaS) and Cloud computing - has changed the way of delivering digital services for consumers and businesses. These technologies have brought clear benefits over traditional systems, such as easy management, universal availability and decreased hardware requirements for end-user devices. The main challenges of the first-generation IoT services are related to communication latency due to high physical and logical distance between end-nodes and server resources, and vulnerability for network problems along the long routes. Thanks to the unveiling of the fifth-generation wireless technology (5G) for cellular networks, the last-mile connection performance - communication latency in particular - is taking a huge leap, and therefore introducing new possibilities for industrial applications. Edge computing is a key technology to unleash the full potential of the arising industrial wireless communication, since it enables deploying computational tasks to computing nodes near the end devices and therefore opens novel business opportunities around real-time cloud services. In this chapter, we introduce the current state of the art and discuss different challenges the edge computing systems are facing particularly in the Industrial IoT (IIoT) domain, as well as present potential solutions for the identified challenges.

Key words: Edge computing, Fog computing, Mist computing, IoT, IIoT, Cloud computing, MEC, Virtualization, Microservices, AI.

Erkki Harjula
University of Oulu, Centre for Wireless Communication, P.O.Box 8000, FI-90014 University of Oulu, Finland. e-mail: erkki.harjula@oulu.fi

Alexander Artemenko
Robert Bosch GmbH, Corporate Sector Research and Advance Engineering, Renningen, Germany e-mail: alexander.artemenko@de.bosch.com

Stefan Forsström
Mid Sweden University, Institution of Information Systems and Technology, Holmgatan 10, 851 70 Sundsvall, Sweden e-mail: stefan.forsstrom@miun.se

1 Introduction

Today, we can observe large global trends in the digitalization of many aspects of our everyday life. In particular, we see applications that can utilize information from sensors attached to things that can also communicate among each other over the Internet. This concept is commonly referred to as the Internet-of-Things (IoT) and provides us with services that are more personalized, automated, and have more intelligent behavior. Related to this, we can also see trends in IoT Cloud Computing (CC) for large scale data storage, big data analysis on a massive amount of gathered data from IoT sources, and incorporation of Cyber-Physical Systems (CPS) into machine to machine (M2M) systems. Concurrent to this development, much work is being done in the Industry 4.0 initiative, including smart cities, smart industry, factories of the future, and smart manufacturing; hence, forming the concept of Industrial IoT (IIoT) [1]. At the same time, the deployment of the 5G wireless communication technology is also increasing everyday around the world [2], enabling a new magnitude in speed and low latency wireless communication with ultra-high reliability and availability.

Edge Computing (EC) enables services to exploit the proximity of devices by providing computational resources closer to end-nodes, therefore enabling ultra-low latency and high data rate communication. At the same time, it provides a means for controlling and limiting the propagation of sensitive data. Multi-access Edge Computing (MEC) is a standard by the European Telecommunications Standards Institute (ETSI) for 5G networks, among others, to offload processing and data storage from mobile and IoT devices to the edge of mobile networks instead of passing all of the data and computation to data centers or handling them locally [3]. Fog Computing (FC) and Mist Computing (MC) are closely related to both EC and CC, as they can be interpreted as low flying cloud computing near the edge [4]. We make the distinction that EC mainly refers to the computational edge infrastructure, FC mainly refers to the logical architectures enabling distributed virtualized services on the edge architecture utilizing the hardware capacity of EC nodes, and MC to the extreme edge of the networks and local edge computations. FC typically covers caching, data processing and analytics occurring near the source of the data that improve the performance at the edges of the network, reduces the burden on data centers and core networks and improves the resilience against networking problems [3]. Figure 1 shows an overview of how cloud, fog, mist, and edge computing fit together in a layered structure, including the scale of each layer and typical operations.

IIoT is one of the most important application areas of IoT, and therefore, it is vital for defining the requirements for EC systems. In recent years, the CC paradigm has found its way into the manufacturing industry addressing the need to process vast data originating from a massive number of sensor devices. It offers centralized resources to perform computationally intensive operations. Here, a representative example is predictive maintenance, which detects conditions leading to malfunctions, and therefore enables flexible manufacturing, that increases the reconfigurability level of production systems enabling batch-size-one products. Due to improved connectivity with guaranteed Quality of Service (QoS), machines do not need to rely on their own

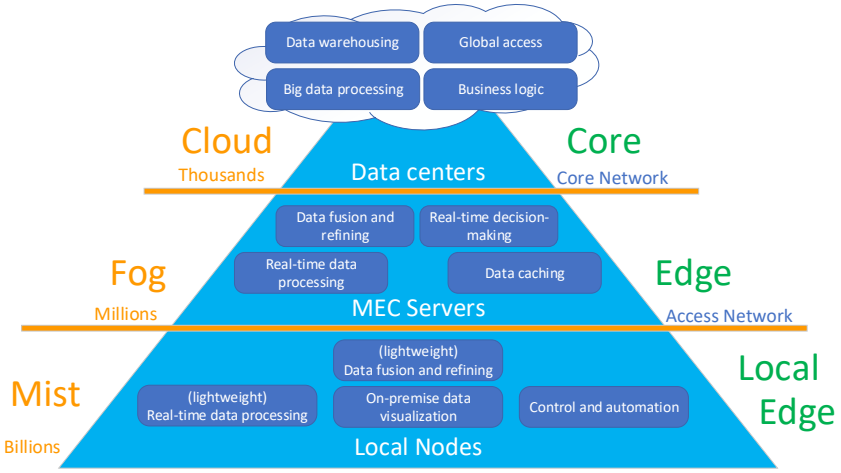

Fig. 1 A simplified view of Cloud, Fog, Mist, and Edge computing layers

dedicated computing hardware anymore, but can rather use connectivity to access the cloud resources [5]. While the use of a centralized CC entity presents several vulnerabilities like single point of failure, backbone congestion, security, and data privacy, EC and MEC introduce computational resources, storage, and services at the edge of a network [6]. Applying EC, FC, and MEC for industrial manufacturing can solve most of the weaknesses of traditional cloud computing. However, several challenges still remain that will be addressed in this chapter.

In order to highlight the advantages, disadvantages, and future research in the intersection of the IIoT and EC, this chapter will give an introduction to the applications, challenges and solutions of MEC. The remainder of this chapter is organized as follows: Section 2 goes further into the state of the art of EC and MEC for industrial use, the standardization, applications, and challenges. Section 3 focuses on solutions and future development potential, the next steps, and research directions. It includes three-tier edge cloud architectures, microservice architectures, SDN and NFV integration, security/privacy management, and the use of artificial intelligence (AI) for MEC. Finally, Section 4 summarizes and concludes the chapter.

2 State of the Art in Edge Computing for IIoT

This section describes the current state of the art in EC with focus on industrial applications. After the introduction of the EC potential for existing and appealing industrial use cases and standardization activities, we focus on the most crucial EC aspects and highlight their relevant open points and challenges.

2.1 Edge Computing Technology for Industrial Use

CC has, already for some time, been a standard in industry, bringing a vast amount of processing capabilities to analyze data generated by a huge number of already operational IoT devices. Many large industrial companies have taken into use their own in-house clouds (aka private clouds), as well as public clouds to satisfy their production needs [7]. Along with this, the vision of a fully automated and flexible factory of the future gets closer to reality. However, novel concepts, such as factory with zero-downtime, digital twins, flexible production planning, pro-active system surveillance, intelligent technical assistance, and batch-size-one products, still require further improvement of the network performance and services on the factory floor [8]. Nevertheless, EC is considered as one of the enabling technologies to unveil the full potential of the proposed smart factory concepts.

The EC technique can support IIoT devices with limited capabilities (with regard to e.g., battery, CPU, and GPU) in their ever-increasing computation demands created by various kinds of novel use cases. High-complexity robotic applications, Automated Guided Vehicles (AGV), real-time interactive multi-user co-working, mobile production cells, sensors and actuators connected over wireless communication technologies, augmented reality (AR), virtual reality (VR) are only a few examples of such industrial use cases [9]. Many of them introduce requirements that differ from those considered in conventional IT systems. Such requirements include ultra-high availability, reliability, predictability, very low latency, and strictly deterministic real-time behavior of all system components. MEC alone cannot satisfy all these requirements. Therefore, a combination of many new enabling technologies is required, e.g., time-sensitive networking, real-time virtualization, software-defined networking, 5G with enhanced mobile broadband, ultra-reliable low-latency and massive machine-type communication, etc. Many of these enabling technologies are covered in this book.

First EC products, known as edge gateways [10], are gaining popularity in the industrial context, connecting thousands of IIoT devices to data processing units at the edge of a network, close to sensors, and hence, avoiding the issues of sending all the data directly to the cloud, which is often not feasible due to cost, privacy, and network issues. Software giants offer first software platforms for deployment and management of edge clouds (e.g., Azure IoT Edge from Microsoft, AWS IoT Greengrass from Amazon, Cloud IoT Core from Google, Bosch IoT Gateway, etc.). Many of these products still present proprietary solutions. To improve this situation,

different standardization activities are working on specifications for different EC aspects.

2.2 MEC Standardization

Shortly after an introduction of small cloud data centers close to the data source called Cloudlets [6], the Open Edge Computing (OEC) and OpenFog Consortium (OFC) initiatives have been generated to accelerate the standardization and dissemination of the EC technology. Thereafter, multiple committees, working groups and standardization bodies around the world have been created. According to [11], the most important ones are the following:

- **Multi-access Edge Computing initiative** as an Industry Specification Group within the European Telecommunications Standards Institute (ETSI),
- **MEC in 5G networks** within the 3rd Generation Partnership Project (3GPP),
- **MEC system as Service-oriented RAN** within the China Communications Standards Association (CCSA).

The majority of the core partners in all standardization entities come from telecommunications industry. This is reflected in the core activities, as well as the goals of working groups. All bodies perform different conceptual, architectural, and functional work and intend to develop a standardized, open environment that will allow efficient and seamless integration of third-party applications across multi-vendor platforms [11]. From the perspective of the authors of this book, however, ETSI MEC initiative considers the broadest range of applications and architecture scenarios among all EC standardization entities. This is reflected in the aspect that the EC platform is not bound to any access technology, which is reflected in the title of MEC.

ETSI MEC introduces a reference architecture and technical requirements enabling efficient and seamless execution as well as interoperability and deployment of a wide range of EC scenarios that include IIoT. The multi-vendor proof-of-concept projects visualize key aspects of MEC technology and prove it is feasible and valuable. Important aspects like latency, energy efficiency, system resource utilization, network throughput, and quality of service are constantly highlighted.

2.3 MEC Applications: Industrial IoT

As mentioned above, EC excels in application scenarios where there is a need for low latency, high bandwidth, and high resilience computation and communication in order to enable its real-time, intelligent, and autonomous decision-making. This can be required, for example, in different smart appliances, such as smart vacuum cleaners using sensor information available inside the house. But also, edge device

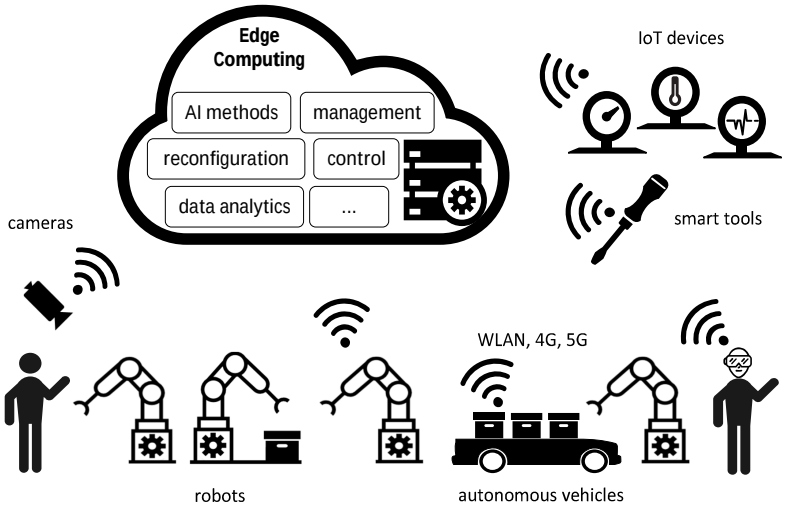

Fig. 2 EC technology in industrial applications

video analysis, mobile big data analysis, connected vehicles, smart building control, and safety monitoring present appealing use cases in the IIoT context. A new trend on the factory floor is represented by different kinds of mobile vehicles, e.g., Unmanned Aerial Vehicles (UAV) and Automated Guided Vehicles (AGV), which cooperatively solve certain tasks. Some typical industrial applications include edge services such as industrial production robots, where the low latency and resilience of EC is paramount. Cooperating robotic arms in a production line show a good and appealing example of the robotic cooperation on the factory floor. Here, EC supports production systems by offloading of data analytics and smart data processing in the close proximity to the data sources. Furthermore, smart industrial environment monitoring, grid system controls and self-organized massive wireless sensor and actuator networks constantly continue to attract manufacturers attention. Many of the applications, mentioned above, have already been implemented using the current technology base. Some of the use cases are shown in Figure 2. Many further useful applications present a source for discussions due to the challenges they introduce to the infrastructure.

First products are available in the IIoT market in this context, proving the benefits of the EC paradigm. As an example, Bosch IoT Gateway presents an IIoT solution with support of open APIs, a variety of development tools for creation of edge applications, providing autonomy and intelligence at the edge [12]. The product is in use in many scenarios including IoT platforms with EC support for intelligent data processing, optimization of electric vehicle charging and smart field device connectivity at the edge [13].

2.4 Edge Computing Challenges

IoT systems can greatly benefit from the EC technology, but several challenges still remain, related to e.g. performance, efficiency, reliability, availability, scalability, security and privacy [14]. The following sections discuss these challenges in more detail.

2.4.1 Performance and Quality of Service

Novel 5G wireless technologies enable low-latency communication, which is crucial in various IIoT scenarios requiring real-time functionality [15]. As mentioned earlier, MEC (and EC in general) is another of the two main enablers of reliable low-latency wireless services since it minimizes the route length between the local nodes and computing resources [16, 17, 18, 14]. EC also helps improving other QoS factors since it is easier to remove and manage performance bottlenecks on short routes compared to longer ones. Most of the challenges related to maintaining high performance of EC/MEC systems concern special situations, such as fast mobility of end-nodes and rapid changes in both service demand and density of end devices, which can be the case in many industrial applications or, e.g., during public mass events. Therefore, important research topics are related to, e.g., placement of edge resources and deciding where to deploy computation and data in different scenarios. The ability to rapidly migrate data and computations among MEC servers or between MEC and core network servers to ensure QoS in dynamic scenarios, is among the most important research areas of EC.

2.4.2 Reliability and Availability

The resilience against computing and network infrastructure problems is one of the most important research areas of EC/MEC [19]. In many IIoT systems, the processing of sensor data, at least some degree of the decision-making logic and the control of actuators is beneficial to manage locally on site, because the connection with the access network may occasionally become unreliable or low in performance [20, 18]. Therefore, it is beneficial to bring some EC capacity also within local IoT clusters. This is called “local edge computing”. The challenges related to reliability of EC systems concern the ability to adapt to dynamically changing situations, related to, e.g., mobility, network failures, disturbances, and hardware failures. The EC system should automatically manage, analyze and optimize its operation, including, e.g., placement of data management and computational tasks, based on the current situation and foreseen changes. With regard to availability, the critical questions are, where are the system components located, and who/where are the users? Availability becomes a particular problem in cases where the different stakeholders of the services are logically and geographically distributed.

2.4.3 Scalability and Deployability

In IIoT systems, sensor information is gathered from a high number of devices connected with short-range or novel long-range low-power wireless technologies [1, 21]. Furthermore, advanced sensors, control systems, surveillance video streaming and still image capturing devices are already producing huge amounts of data to be processed [22]. In traditional systems, all of this data processing and related decision-making logic has been handled at data centers, which is becoming problematic from the viewpoint of scalability, performance, and reliability. In this context, EC helps by providing computational capacity near the source of the data, allowing various data pre-processing, refining and analysis functions to reduce the amount of data to be sent to cloud servers and therefore reducing the load inflicted to core networks and data centers. The important research challenge in this area is to develop intelligent algorithms for deciding on which tier to manage different functions and prioritizing tasks when limited resources do not allow globally optimal solutions.

2.4.4 Security, Privacy and Trust

Since the digital world penetrates deeper and deeper in the industry and business processes, as well as our everyday life, a particular concern is related to preserving the privacy and security of networked systems. We are living in a world where the data collected from the users is ruthlessly exploited by different organizations and authorities around the world. Centralized cloud-systems are an inherently challenging environment from the viewpoint of security and privacy, since all data need to pass several links and devices, owned by a wide set of stakeholders between end devices and servers, not forgetting the chance for data leaks at public servers [14, 23, 24]. What is even worse, IoT, surrounding us almost everywhere, gives cybersecurity attackers further tools to even threaten our health or life by infiltrating to systems affecting our physical safety [25]. Therefore, the need for technologies allowing local data management and decision-making to limit data propagation towards public networks is obvious. In this context, EC is a centric building block for service providers to guarantee customer data preservation within set boundaries. Regarding security and trust challenges in the EC and IIoT domain, there are still many open and difficult challenges [26], including end device security, protocol and network security, cloud/fog security, end user application security, data protection, malicious attacks as well as identity and authentication management, access control, trust management, intrusion detection systems, privacy, virtualization, and forensics.

2.4.5 Resource-Efficiency

Resource-efficiency, including energy-efficiency, is a powerful measure for promoting sustainability in technological evolution. Internet and Communications Technology (ICT) is one of the main tools for improving the resource-efficiency of the

infrastructures around us [27], but its intrinsic resource demand is rising rapidly [28]. In this context, local data pre-processing, refining and analysis functions enabled by EC, help reducing the load inflicted to various components of the cloud systems and therefore promote sustainability through improved energy- and resource-efficiency. IoT systems include numerous low-power sensors, actuators and other devices that are resource-constrained in their nature [18, 14]. In order to maintain both the system-level performance and resource-efficiency of constrained-capacity nodes, IoT systems need to take into account the limited hardware and energy capacity of the end-nodes. One of the main measures for achieving this is to offload computing and data management to higher layers on the IoT architecture. The traditional IoT systems do this by offloading computation to cloud servers. In data-intensive computing, such as video surveillance, this is not optimal from the viewpoint of network utilization, since all data need to pass several communication links along the way from the end-node to the server. A more efficient approach would be to handle as much of the data-intensive computing near the source of data as possible. In this context, EC is in a key position to improve resource-efficiency. The challenges related to resource-efficiency concern, e.g., how to reliably measure and communicate resource usage, how to minimize resource consumption while still maintaining the availability of nodes in highly interactive scenarios, and how to prioritize resource-efficiency with several other constantly changing requirements in complex multi-tier IoT systems.

Being not complete, the list of challenges, presented in this section, gives an overview of open points, which show a potential for further improvements.

3 Solutions and Future Development Potential

In the previous section, we pointed out some important research challenges for EC in IIoT. To address those challenges, in this section we discuss on some of the most relevant research directions and potential solutions.

3.1 Three-tier IoT Edge Architecture

To deal with the vast amount of data originating from a massive number of sensor devices, the risk of connectivity problems, and to limit the propagation of sensitive data, at least some degree of processing of the sensor data and decision-making/control logic is beneficial to be managed locally. Since it cannot be expected that local IoT clusters include devices with sufficient stability and hardware capacity to accommodate full-functional MEC servers, decentralized solutions become essential to accommodate the local processing, data management and decision-making. To make this possible, a three-tier IoT Edge model has been proposed by the authors in [18]. In this model, the data and processing can be deployed on three alternative

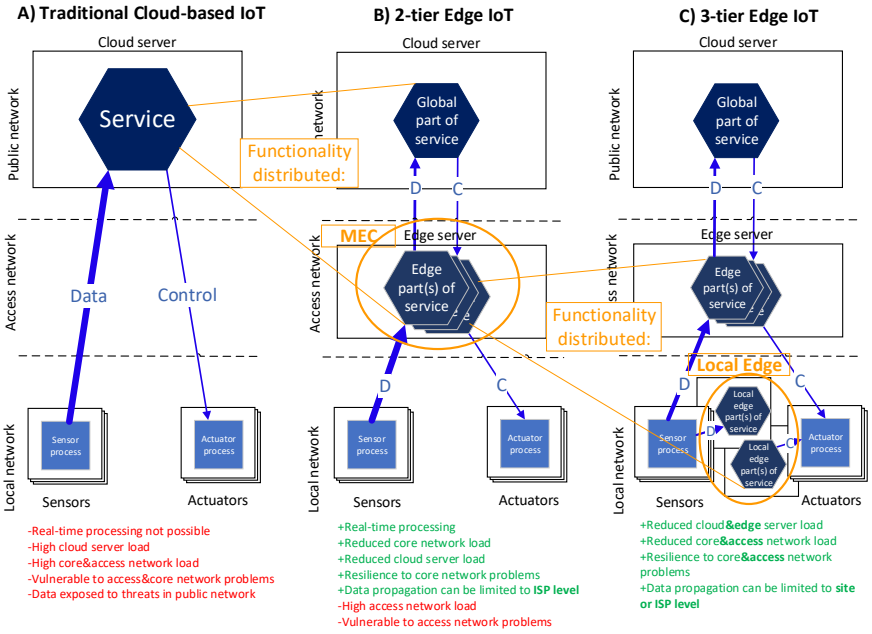

Fig. 3 Three-tier IoT Edge Architecture

levels of operation: 1) public servers, e.g., in data centers, 2) MEC servers, and 3) local nodes as virtualized functions. Fig. 3 illustrates the model and its benefits. The model enables dynamic optimization of service deployments, based on the service requirements, available computational and network capacity, and load. We see high potential in using microservice and serverless architectures, introduced in Section 3.2, and AI (Section 3.5), in defining the optimal deployments for different types of services in the three-tier architecture.

3.2 Microservices and Serverless Architectures

IoT services have traditionally been designed as monolithic cloud applications associating multiple software components into a single entity. Due to their ponderous maintenance and deployment, the current industry trend is towards microservice paradigm, an architectural style to build, manage, and evolve service architectures consisting of small, self-contained units, microservices [29, 30], which enable the development of distributed service compilations. Microservices are small and self-acting virtualized components, typically based on container technology (such as Linux or Docker containers), that are relatively easy to develop in isolation and maintain as standalone software components. Microservice architectures enable

continuous software evolution, seamless technology integration, optimal runtime performance, horizontal scalability and reliability through fault tolerance.

A new class of applications is emerging, namely "serverless applications". They are exemplified by Function-as-a-Service (FaaS) [31, 32] systems that enable the future "service anywhere" architecture. FaaS has been considered as one of the technologies to realize lightweight microservices, also called as FaaS functions or "nanoservices" [18]. Whereas traditional microservices have larger role and are expected to be always available, FaaS-functions are considered as smaller logical units that become alive when needed, then execute, and terminate when not needed anymore. Since FaaS functions typically do not run long periods and their size is small, their deployment does not require dedicated servers. Based on this, FaaS functions can be deployed on any device providing sufficient computational capacity, and therefore FaaS is also called as serverless computing. In FaaS-based agile development of microservices, the developers do not need to consider the computing infrastructure, as both resource provisioning and scaling are automated.

Microservices and serverless architectures can be seen as a technology for implementing various types of fog services on the three-tier IoT edge architecture, where service functions can be deployed on the most optimal tier, based on the current conditions of the network and computational environment.

3.3 Integration with SDN and NFV

EC per se represents a distributed approach that combines end devices and processing capabilities of remote servers. The network and its performance become a vital part of the EC paradigm. Network resources require a simple and flexible management to deal with the low latency and reliability requirements in addition to the huge data transmissions involved. Software-defined Networking (SDN) is a new networking approach that decouples the network control from the data forwarding hardware. The network intelligence is logically located in software-based controllers (aka control plane) and the network devices become mere packet forwarding entities (aka data plane) [33]. SDN enables a new level of network management including better control, higher flexibility, and scalability. Moreover, SDN introduces fast network reconfiguration and self-healing that address important issues, such as user and application mobility, as well as uninterrupted service provisioning in the case of factory automation. Integrating SDN mechanisms with EC helps to provide the required computation resources and to satisfy the unique quality of service requirements of the applications, which is one of the MEC challenges as mentioned in Section 2.4. Having been researched over a long time, SDN has seen many novel approaches for network management, control, fast reconfiguration, healing, network function abstraction, placement, etc. Especially in wireless and mobile scenarios, SDN is able to make networks more controllable and programmable, update routing tables according to the often predictable mobility patterns of the nodes, and thus select the most appropriate paths from or to the end devices (e.g., [34]). Thus, SDN

addresses many weak and challenging aspects of EC. However, there are only a few works that take those SDN approaches to the factory floor where they could be highly beneficial.

Another complementary network technology is presented by Network Function Virtualization (NFV). It adds on further hardware abstraction and can be combined with SDN to extend the virtualization approach towards higher layer network functions like load balancing, firewalls, intrusion detection, WAN acceleration, etc. The integration of SDN and NFV with EC brings a lot of new possibilities that improve the overall network and computational performance. In an industrial context, however, the establishment of any hardware abstraction is only feasible if such vital capabilities like availability, reliability, predictability, and deterministic behavior of the resulting system are not harmed. These aspects are still not sufficiently covered in the state of the art and present a research opportunity for both academia and industry.

3.4 Security, Privacy and Trust Management

Many different potential security attack vectors and risks for privacy breaches exist in an IIoT value chain from sensors, via gateways and fog nodes, to data centers, including end-user applications. The remainder of this section will present details of some of the identified open challenges in each of these areas.

The end devices are an integral part of EC systems due to the additional responsibilities that have been given to those. However, securing devices against unauthorized access by a malicious person is extremely difficult. Thus, key management, storing the keys and handling them in a secure way becomes paramount. It is also not uncommon to see hard-coded keys or group-key systems on IIoT devices, where a single compromised device can compromise the whole system security. There are many examples of extracting keys from devices if one has access to a physical device. Examples include physical side channel attacks, tampering, reverse engineering, power/electromagnetic analysis, timing attacks, known fault attacks, and clock glitches. End devices also tend to be the target of malicious software, including trojan horses, spyware, viruses, and other malware.

Network security is also a difficult but integral part of IIoT EC systems. The broad and heterogeneous network architecture with multiple network components using different hardware and software implementations is a challenging environment for security management. Different networks have their own vulnerabilities and weaknesses, for example, Local Area Networks (LAN), Wide Area Networks (WAN), low-power wide-area networks (LPWAN), and industrial networks. Therefore MEC and IIoT systems need to take a broad range of network types into consideration, making this a difficult challenge. Additionally, the wireless communication medium, which is often used in the IIoT, introduces an extra vulnerability and an opening for a wide range of attacks such as eavesdropping and jamming.

Another important challenge is dealing with trust and securing sensitive industrial data. This includes hiding and protecting the sensitive industrial data, such as sensor

values, algorithms, and industrial process information, where a data breach can lead to competitors gaining an advantage over them. Therefore, the need for technologies allowing local data management and decision-making to limit data propagation towards public networks is obvious. In this context, EC is a centric building block for providing guarantees for customers to keep their data within set boundaries. However, the systems consisting of functions distributed on computing nodes on several architectural tiers, owned and managed by different stakeholders with their own security policies, are inherently very complex, which requires attention in the future research. There is a clear need for Security as a Service-type components, capable of running in constrained-capacity nodes [35]. Furthermore, building trust between stakeholders of these complex systems, based on, e.g., Blockchain [36] is an interesting avenue for future research.

3.5 Use of Artificial Intelligence for EC Optimization

AI has become a very important technique in many different domains. Being now for a long time successfully used in applications like speech and image recognition, strategic game systems (chess and Go), autonomous robots, etc., AI methods can be applied in EC for the optimization of many different aspects. In this section, we want to highlight several approaches of Machine Learning, being an area of AI, applied for task offloading.

The EC servers are usually densely distributed close to end devices to reduce the cost for offloading of computational tasks to these servers through wireless or wired links. Among many benefits, users can observably reduce the experienced delay of applications, energy consumption and improve the QoS with the help of offloading. However, a list of unresolved questions arise [37]:

- **What** part of an application needs to be offloaded considering the complexity of the application, data to be shared between the user device and the EC server as well as the available network capacity?
- **When** is an optimal time to start the offloading considering the dynamic behavior of the end device, the available network capacity, as well as the dynamic load of the server?
- **Where**, on **which** node and at **which** architectural tier (local node/EC server/cloud server) should the offloaded task be processed considering the CPU and GPU availability on different nodes as well as the distance to these nodes from the user device?
- **How** should the offloading be organized?

In a complex scenario, offloading becomes a multi-objective decision-making problem. Designing an offloading strategy does not have a straightforward solution due to the dynamic behavior of EC systems. Stochastic characteristics of edge environment can make pre-decided offloading strategy impractical. Reinforcement Learning (RL), an area of Machine Learning, can be applied in training an AI agent to

observe the current state of the EC system, to make an intelligent offloading decision based on specified criteria and to learn from the history of such decisions. However, conventional RL algorithms cannot scale well as the number of edge devices increase, since the explosion of state space will make traditional tabular methods of RL infeasible. Another approach from the Machine Learning area is based on Deep Learning (DL), aka Deep Neural Networks (DNN). It operates efficiently with a large number of state spaces. The benefits of using DNN methods in EC is to extract hidden patterns from large and complex data sets of heterogeneous applications. A combined strategy, called Deep Reinforcement Learning (DRL) [38], shows a good offloading performance in various complicated EC scenarios. DRL methods treat the complicated EC system as a black box and interact with it to learn the optimal policies without modeling the system dynamics. Although there are significant advantageous in DRL methods, notable challenges related to dynamic behavior of considered applications remain in applying DRL to solve task-offloading problems in EC.

4 Conclusion

The unveiling of novel 5G and EC technologies will be one of the major driving factors in increasing productivity and therefore key enablers for long-envisaged vertical applications in various sectors including IIoT. In this book chapter, we have given an introduction to the applications, challenges and solutions of EC including an overview of the state of the art in EC for IIoT, different standardization activities, open challenges and future development potential. Based on this, we believe that EC is an important piece of the IIoT puzzle and a key concept to meet the demands of future industrial services. The open challenges and research directions mentioned in this chapter represent attractive points for improvement and active work in both academia and industry. For example, the solutions of using three-tier IoT edge architecture, microservices and serverless architectures, integration with SDN and NFV, the use of AI for EC optimization as well as aspects of security, privacy and trust management have just recently become popular discussion hotspots around EC technology. In each of the mentioned areas, we have highlighted the advantages, disadvantages, and needed future research for the proliferation of the IIoT and EC in particular.

References

1. Li Da Xu, Wu He, and Shancang Li. Internet of Things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.
2. M. Shafi, A. F Molisch, P. J Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder. 5G: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE journal on selected areas in communications*, 35(6):1201–1221, 2017.

3. P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. Edge-centric computing: Vision and challenges. *SIGCOMM Comput. Commun. Rev.*, 45(5):37–42, September 2015.
4. C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana. Fog computing for the Internet of Things: A survey. *ACM Trans. Internet Technol.*, 19(2), April 2019.
5. G. Mohanarajah, D. Hunziker, R. D'Andrea, and M. Waibel. Rapyuta: A cloud robotics platform. *IEEE Transactions on Automation Science and Engineering*, 12(2):481–493, 2014.
6. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, Oct 2009.
7. P. Cardoso, J. Monteiro, J. Semi ao, and J. Rodrigues. *Harnessing the Internet of Everything (IoE) for Accelerated Innovation Opportunities*. IGI Global, 02 2019.
8. X. Chen, L. Jiao, W. Li, and X. Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, October 2016.
9. K. Govindaraj, D. Grewe, A. Artemenko, and A. Kirstaedter. Towards zero factory downtime: Edge computing and SDN as enabling technologies. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 285–290, Oct 2018.
10. R. Morabito, R. Petrolo, V. Loscri, and N. Mitton. LEGIoT: A lightweight edge gateway for the Internet of Things. *Future Generation Computer Systems*, 92, 11 2018.
11. Y. Ai, M. Peng, and K. Zhang. Edge computing technologies for Internet of Things: a primer. *Digital Communications and Networks*, 4(2):77 – 86, 2018.
12. Edge computing: the perfect complement to the cloud in IoT. https://www.bosch-iot-suite.com/edge-computing/. Accessed: 2020-03-09.
13. Connectivity and intelligence at the edge of IoT. https://developer.bosch-iot-suite.com/service/gateway-software/. Accessed: 2020-03-09.
14. P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb. Survey on multi-access edge computing for Internet of Things realization. *IEEE Communications Surveys Tutorials*, 20(4):2961–2991, Fourthquarter 2018.
15. L. Chettri and R. Bera. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet of Things Journal*, 7(1):16–32, Jan 2020.
16. T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili. Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, 55(4):54–61, April 2017.
17. P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys Tutorials*, 19(3):1628–1656, thirdquarter 2017.
18. E. Harjula, P. Karhula, J. Islam, T. LeppÃđnen, A. Manzoor, M. Liyanage, J. Chauhan, T. Kumar, I. Ahmad, and M. Ylianttila. Decentralized IoT edge nanoservice architecture for future gadget-free computing. *IEEE Access*, 7:119856–119872, 2019.
19. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645 – 1660, 2013. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications âĂŤ Big Data, Scalable Analytics, and Beyond.
20. F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., USA, 2006.
21. O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu. Context-aware computing, learning, and big data in Internet of Things: A survey. *IEEE Internet of Things Journal*, 5(1):1–27, Feb 2018.
22. G. Xu, E. C. . Ngai, and J. Liu. Ubiquitous transmission of multimedia sensor data in Internet of Things. *IEEE Internet of Things Journal*, 5(1):403–414, Feb 2018.
23. A. Sadeghi, C. Wachsmann, and M. Waidner. Security and privacy challenges in industrial Internet of Things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2015.
24. P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits. Denial-of-service detection in 6lowpan based Internet of Things. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 600–607, Oct 2013.

25. BMWs connected drive feature vulnerable to hackers. https://www.autoblog.com/2015/02/03/bmws-connected-drive-feature-vulnerable-to-hackers. Accessed: 2020-01-26.
26. R. Roman, J. Lopez, and M. Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 78:680–698, 2018.
27. A. Kramers, M. Hoejer, N. Loevehagen, and J. Wangel. Smart sustainable cities – exploring ICT solutions for reduced energy use in cities. *Environmental Modelling & Software*, 56:52 – 62, 2014. Thematic issue on Modelling and evaluating the sustainability of smart solutions.
28. B. Schlomann, W. Eichhammer, and L. Stobbe. Energy saving potential of information and communication technology. *International Journal of Decision Support Systems*, 1(2):152–163, 2015.
29. C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi. Cloud container technologies: A state-of-the-art review. *IEEE Transactions on Cloud Computing*, 7(3):677–692, July 2019.
30. R. Rodger. *The Tao of Microservices*. Manning, 2018.
31. J. Kuhlenkamp and S. Werner. Benchmarking FaaS platforms: Call for community participation. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pages 189–194, Dec 2018.
32. P. García López, M. Sánchez-Artigas, G. París, D. Barcelona Pons, Á. Ruiz Ollobarren, and D. Arroyo Pinto. Comparison of FaaS orchestration systems. In *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, pages 148–153, Dec 2018.
33. B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys Tutorials*, 16(3):1617–1634, Third 2014.
34. M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V Vasilakos. Software-defined and virtualized future mobile and wireless networks: A survey. *Mobile Networks and Applications*, 20(1):4–18, 2015.
35. P. Ranaweere, V. N Imrith, M. Liyanage, and A. D. Jurcut. Security as a service platform leveraging multi-access edge computing infrastructure provisions. In *International Conference on Communications (ICC), 2020 IEEE*. IEEE, 2020.
36. M. Cinque, C. Esposito, and S. Russo. Trust management in fog/edge computing by means of blockchain technologies. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1433–1439, July 2018.
37. H. Wu. Multi-objective decision-making for mobile cloud offloading: A survey. *IEEE Access*, 6:3962–3976, 2018.
38. J. Wang, J. Hu, G. Min, W. Zhan, Q. Ni, and N. Georgalas. Computation offloading in multi-access edge computing using a deep sequential model based on reinforcement learning. *IEEE Communications Magazine*, 57(5):64–69, May 2019.