Industrial Internet of Things (IIoT) Applications of Edge and Fog Computing: A Review and Future Directions

G.S.S. Chalapathi, Vinay Chamola, Aabhaas Vaish, and Rajkumar Buyya

Abstract With rapid technological advancements within the domain of Internet of Things (IoT), strong trends have emerged which indicate rapid growth in the number of smart devices connected to IoT networks and this growth cannot be supported by traditional cloud computing platforms. In response to the high volume of data being transferred over these networks, the edge and fog computing paradigms have emerged. These paradigms are extremely attractive frameworks that shift computational and storage resources from the centralized cloud servers to distributed LAN resources and powerful embedded devices at the edge of the network. These computing paradigms, therefore, have the potential to support massive IoT networks of the future and have fuelled the advancement of IoT systems within industrial settings, leading to the creation of the Industrial Internet of Things (IIoT). IIoT is revolutionizing industrial processes in a variety of domains. In this chapter, we elaborate on the impact and viability of edge and fog computing paradigms in IIoT through a use-case approach. Finally, we conclude with the future directions of these paradigms in IIoT. Among other research directions, the security and privacy of IoT devices are very important criteria for the adoption of IoT for industries. Thus we highlight some of the major security and privacy concerns concerning edge and fog computing in IIoT. We also discuss the relevance of Blockchains for IIoT for secure industrial deployments and potential issues in such implementations.

G. S. S. Chalapathi
Department of Electrical and Electronics Engineering, BITS Pilani, Pilani Campus, India.
Previously with Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Australia. e-mail: gssc@pilani.bits-pilani.ac.in

Vinay Chamola, Aabhaas Vaish
Department of Electrical and Electronics Engineering, BITS Pilani, Pilani Campus, India. e-mail: {vinay.chamola, f2016370}@pilani.bits-pilani.ac.in

Rajkumar Buyya
Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, The University of Melbourne, Australia. e-mail: rbuyya@unimelb.edu.au

1 Introduction

The Internet of Things (IoT) [1] refers to a system of smart devices that are connected through the Internet. The basic structure of IoT systems involves the use of a large number of smart devices that can acquire, process, transmit, and receive data between one another. IoT devices thereby enable us to reliably monitor and precisely control any environment, control system, or device through this system of interconnected smart devices. With forecasts predicting an estimated 28.5 billion network-connected devices to become active by 2022 [2], the IoT technology is poised to make a total economic impact between $3.9 trillion and $11.1 trillion per year in 2025 [3]. While most of the IoT systems developed until now have been consumer-centric, the disruptive nature of this technology has enabled the adoption of this technology in a gamut of industrial settings thus leading to the development of **Industrial Internet of Things (IIoT) technology** [4]. IIoT technology, in essence, refers to a system of interconnected smart devices in an industrial setting. IIoT connects industrial resources including sensors, actuators, controllers, machines with each other as well as intelligent control systems. These intelligent control systems analyse the acquired data and optimize the ongoing industrial processes to improve execution speed, reduce involved costs, and dynamically control the industrial environment [4].

One of the most important reasons behind the meteoric rise of IIoT systems in various industries is that IIoT systems can lead to a significant improvement in efficiency, throughput, and response time of operations inside these industries [5]. IIoT has already revolutionized companies in many major industries across the globe, including the mining industry where IIoT systems have led to the installation of wireless access points in mining tunnels, and RFID tracking technology has helped companies in tracking vehicles leading to an increase in production levels by 400% [6]. Proposed IIoT systems in agricultural settings can help farmers in nutrient monitoring as well as automated irrigation to improve crop yield [7]. The medical industry can also benefit from the capabilities of Industrial IoT systems where emergency services can access data from patients, ambulances, and doctors to help all stakeholders in making informed decisions and improve resource utilization [8]. Pilot projects in China have successfully implemented an NB-IoT (Narrow Band IoT) system for smart electrical meters which allows real-time collection of power consumption data thereby enabling the energy grid officials to improve the electricity supply strategy in any area [9]. Similarly, NB-IoT smart parking systems have been deployed in cities to help drivers easily find parking spaces. Further, integration of this parking system with payment solutions has led to automated transaction authorization for parking payment which has subsequently improved utilization of parking bays [10]. The railway industry can also leverage the power of IIoT solutions to improve the functioning of surveillance systems, signaling systems, predictive maintenance, and passenger or freight information systems to improve services and safety [11]. Supply Chain Management (SCM) can also benefit by adopting IIoT based systems which will directly enhance tracking and traceability while also aiding in the optimization of shipment routes based on rapidly changing customer

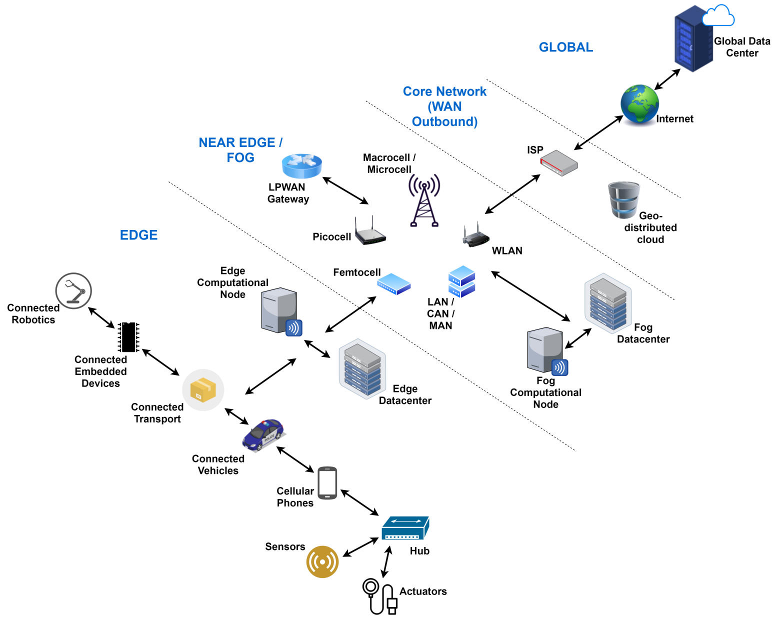

Fig. 1 Edge, Fog, and Cloud Tiers

requirements [12]. While the IIoT shows immense potential as a transformative technology, it is important to know the critical requirements that must be validated and verified in the design of IIoT systems to maximize the efficiency and performance of these systems [13, 14]. These requirements arise from the challenges often faced by Cyber-Physical Systems (CPS). The requirements of IIoT systems include Scalability, Fault Tolerance or Reliability, Data Security, Service Security, Functional Security, and Data Production and Consumption Proximity.

With the rise in computational power being offered by computing systems in general in recent years, the focus of most industries has shifted towards garnering practical and useful patterns from their data which has been aided by the rapid development in statistical analysis and learning-based algorithms. Today, industries that are making use of IIoT solutions want to utilize the massive amount of data being generated to collect useful insights which can help in the reduction of unplanned downtimes, improve the efficiency of production, lower energy consumption, etc. However, to process such massive amounts of data, IIoT systems generally require cloud computing services which often experience large round-trip delays and poor Quality of Service (QoS) as a large amount of data needs to be transferred to centralized data-centers for computation [15]. Since most sensors and data acquisition devices in IIoT systems operate at the periphery of the network, more data tends to be produced near the periphery of the network, which implies that processing the data at the edge of the network would be more efficient [16]. Therefore, efforts in shifting the computational power towards the periphery of the network have given rise to the edge and fog computing paradigms.

Edge Computing refers to the computing paradigm in which computations are performed at the edge of the network instead of the core of the network. In this scenario, the "edge" refers to any resource located on any network path between data acquisition devices (situated near the periphery of the network) and the cloud datacentre (situated at the core of the network) [16]. The basis of the edge computing paradigm is that the computations should be done on the "edge" which is in the proximity of the data sources and this avoids the latency associated with data transfer to the network's core.

The **Fog Computing** paradigm is similar to edge computing in that it also has a decentralized architecture for computation but with the fundamental difference being that Fog Computing can expand to the core of the network as well [17]. This means that resources located at both edge and core can be used for computations and consequently, fog computing can aid in the development of multi-tier solutions that can offload service demand to the core of the network as the load [17]. However, in most fog computing systems, the computational power is concentrated with the LAN resources which are closer to the data sources and further away from the network core, thus reducing the latency associated with data transfer to the core as seen in edge computing as well. Therefore, the fundamental difference between the edge and fog computing paradigms is basically in the location where the computational power and intelligence are stored. In the case of edge computing, this computational power is concentrated at the edge of the network usually in powerful embedded devices like wireless access points or bridges whereas, in the case of fog computing, the computational power is usually in the LAN resources. The rest of this chapter is organized as follows: Section II discusses the background of edge and fog computing systems and how these paradigms address the requirements of modern IIoT systems. Section III describes various applications of edge computing in industrial settings. Section IV elaborates on fog computing applications. In Section V we present several outstanding issues and challenges with these computing paradigms that can be interpreted as future directions for research in this domain. Finally, in Section VI we conclude with the salient points of this chapter.

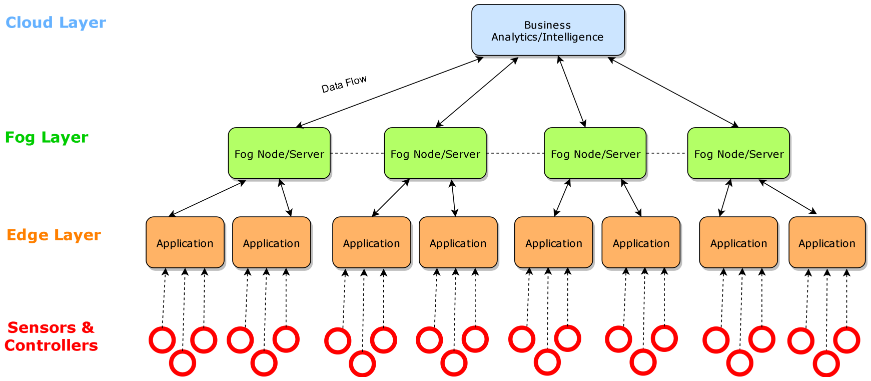

Fig. 2 Industrial IoT Data Processing Layer Stack

2 Relevant Computing Paradigms and Requirements

IoT is seen as a major technological turn-around in various applications. However, due to the high volume of data which is generated by several IoT devices, it is extremely difficult to forward all this data to a central cloud server for processing as it lays heavy stress on the network. Also, it increases the latency involved in processing the data on the cloud server and receiving the results or carrying out a response on the IoT devices. Edge computing paradigm is a computing technology that enables data to be processed almost exclusively on the "edge" of the network, which refers to locations between the end devices (like sensors, controllers, and actuators) and the centralized cloud servers. The rationale behind the development of this technology is that computations performed closer to the end devices will lead to lower latency in the system. This is because the system does not need to transfer data between edge devices and central cloud servers as the computations have been offset to closer locations on the edge. Therefore, in edge computing systems, edge devices can not only request content and services from the cloud servers but can also perform computational offloading, caching, storage, and processing, thereby making the edge devices both data producers and consumers [16].

The fog computing paradigm can be understood as an extension of the traditional cloud computing model wherein additional computational, data handling, and networking resources (nodes) are placed at locations on the network which are close to the end devices [18]. The consequence of this extension is that processes involving data management, data processing, networking, and storage can occur not only on the centralized cloud servers but also on the connections between end devices and the cloud servers [19]. Fog computing, therefore, can be extremely useful for low latency applications as well as applications that generate an enormous amount of data that cannot be practically transferred to cloud servers in real-time due to bandwidth constraints [20].

As discussed in the previous section, there are many requirements that cyber-physical systems need to maintain to become a viable supplement for real-world operations and applications. These include the following:

1. **Scalability** which ensures that the increased data transfer between nodes does not degrade latency or response time.
2. **Fault tolerance and reliability** which guarantees that the system functions normally under variable external factors like under high load conditions.
3. **Data security** which ensures that the system is resistant to external attacks attempting to steal confidential information stored in the system or network.
4. **Service security** to make the system resistant to external attacks which are attempting at disrupting the service provided by the system to the industry such as through Denial-of-service (DoS) attacks or Blackhole attacks.
5. **Functional security** so that physical accidents such as fires, explosions, leaks do not occur at any time especially in industries handling potentially hazardous substances such as nuclear plants, chemical plants, and oil rigs

6. **Data production and computation proximity** which ensures that the devices collecting the data and the systems processing the data are close to each other over the network to reduce latency.

To realize the benefits offered by the edge and fog computing paradigms, IIoT systems must be designed as per the network structures of these paradigms since these paradigms adhere to all the requirements of cyber-physical systems:

1. Edge and fog computing-based systems are **scalable** since increased data transfer between nodes can be addressed by the introduction of additional edge devices to compensate for the added computational load without degrading the network's latency since these devices function in proximity to end devices, and hence, do not increase the data transfer delays over the network.
2. Edge and fog computing systems are **reliable and fault tolerant** especially when compared with cloud-based systems since faults in the centralized cloud servers would result in a total loss of service but the decentralized nature of Edge and Fog Computing systems ensures that even if some of the computational nodes fail, the remaining healthy nodes can still maintain partial service. Furthermore, if the computational load of the failed nodes can be offset to the remaining healthy nodes, then the system can still run full service while corrective action is undertaken.
3. Edge and fog computing systems maintain **data security** within the system due to data decentralization which means that if an adversary wants to breach the system, it would need to breach each one of a large number of decentralized computing nodes to collect the entire system's data.
4. Edge and fog computing systems maintain **service security** by using advanced defense mechanisms such as per-packet-based detection, data perturbation, and isolation networks for the identification of and defense against attacks [21].
5. Edge and fog computing systems ensure **functional security** since these systems as they can be used to create extremely stable and robust multi-loop control systems for functionally sensitive industrial operations such as temperature control [22].
6. Edge and fog computing systems were developed with the rationale that **data consumption** (processing, storing, caching, etc.) and **production** are always in **proximity** which is ensured by the fundamental structure of these systems where computational nodes are located on the edges of the network, which are close to the end devices at the periphery of the network.

The distributed nature of edge and fog computing systems leads to several advantages in terms of reduced communication times and improved reliability, which makes these systems especially useful in a variety of industrial settings that require reliable, latency-sensitive networks for process automation. By realizing the inherent advantages of these paradigms, a large number of industries have started to utilize these paradigms in their system designs and we shall look at several such use cases in the following sections of this chapter.

3 Industrial Applications of Edge Computing

3.1 Manufacturing Industry

To understand the applications of edge computing in manufacturing, we will be considering the system architecture for a manufacturing-based setup as presented in Fig. 3. After describing this architecture, a case study is presented which is based on the implementation of an active maintenance system on a prototype platform. Finally, this subsection concludes with a summary of the tests and results from this case study, as presented in [24].

3.1.1 System Architecture

As depicted in Fig. 3 the architecture has been divided into four domains as follows:

a. The application domain is responsible for providing a comprehensive oversight over the entire manufacturing system to aid in the active administration of the system. This oversight includes services such as monitoring of data flow and network health, as well as the capacity for control of the system. The application domain, therefore, allows the system to provide flexible, generalized, and interoperable intelligent applications while also aiding in the maintenance of service security.

b. The data domain is responsible for providing services such as data cleaning, feature extraction, and intelligent inference, which enables the system to optimize system operations to improve the system's throughput and efficiency. Another important feature of this domain is that it allows end nodes to quickly access data, due to the proximity of the edge computing node and the end devices, which aids in generating real-time responses for specific events. Therefore, this is a critical part of dynamically controlled manufacturing systems.

c. The network domain, in essence, is responsible for connecting the end devices with the data platform and this domain utilizes the Software-Defined Networking (SDN) architecture [25] to manage operations involved in the control plane and network transmission. A Time-Sensitive Network (TSN) protocol is also employed within this domain to handle time-sensitive information and is used extensively in processing the information related to the network in sequence. This domain also offers universal standards for sustaining and supervising the time-sensitive nodes, making it a critical part of the overall system architecture.

d. The device domain refers to the devices located or embedded within the field apparatus like machine tools, controllers, sensors, actuators, and robots. This domain must be able to sustain an infrastructure for flexible communication models to maintain a variety of communication protocols by maintaining nodes that change the system's execution strategies dynamically based on the inputs obtained from the sensors. We normally observe that on the edge nodes, the information model is built with popular protocols such as OPC UA [26] and Data

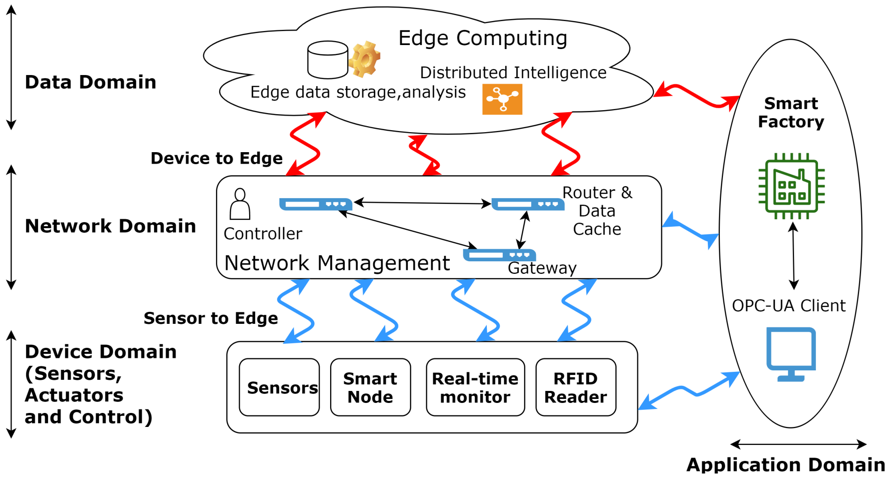

Fig. 3 Architecture of IoT and Edge Computing-based Manufacturing

Distributed Service (DDS) [27]. Finally, the unified semantics of information communication is realized within this domain of the system architecture, and it is also responsible for maintaining data privacy and security.

3.1.2 Active Maintenance Case Study

With the proliferation of cyber-physical systems, a wide variety of industrial projects are being migrated to edge computing-based frameworks because of the promise of improved efficiency, ease of maintenance, and real-time adaptability offered by this computing paradigm. We shall be reviewing a case study on a customized production line for candy packaging, as entailed in [28]. In this study, a private cloud was used to provide service to customer orders. To make stable and high-speed communications possible, an ad-hoc network was built connecting the edge nodes. Furthermore, to achieve a proper exchange of information, a standardized version of the DDS protocol and ethernet were integrated before the deployment of the system. The functioning of the system can be summarized as:

i. Candy packaging tasks were associated with each robot and these tasks were also linked to the cloud. After getting their assigned tasks, the robots were required to pick up the particular candy assigned to them and keep the candy into the relevant open packaging. In this operation, backbone network nodes were represented by the robots.

ii. System was also capable of shifting nodes to different positions on the production line in case of any failures. Therefore, a system with multiple agents was established to improve the self-governing functionality in this scenario.

iii. The agents of the system, physically represented by robots, were independent and self-directed which means that their objective and behaviour was not constrained by other agents of the system.
iv. This system of multiple agents was deployed to complete tasks efficiently by assigning different agents with various tasks and procedures.
v. CNP (Contract Net Protocol) was used to assign different tasks to different agents by using techniques such as winning modes, bidding and open tendering.
vi. By the means of contests and discussions the agents can bargain and resolve their conflicts and so this self-organized system can efficiently complete the assigned tasks.

The implementation of this scenario was made possible with various setups, which include the following:

i. With the help of the Hadoop architecture, a distributed data processing system was built wherein at the local database level, real-time mining and analysis was performed with the help of Hadoop MapReduce and Hadoop Distributed File System (HDFS).
ii. Information such as machine status and logs constituted the sensory data which was used to create a reasoning-based model which was loaded onto a Raspberry Pi system.
iii. On the Raspberry Pi, an OPC UA server was made functional to perform pre-processing tasks on the transmission data that was acquired from different sensory devices. This data was raw in nature and hence, had to be transmitted safely and reliably which was made possible by the use of OPC UA server.
iv. To integrate the data received from multiple sources, a semantic model was also built which reformed the data to maintain consistency, accuracy, and merit of the information. This semantic model used data fusion to provide generate features as inputs from the acquired data. Finally, this data was used as input to the reasoning-based model.

3.1.3 Tests performed

To estimate the difference in performance obtained by using an edge computing-based system instead of a centralized cloud computing system, a cloud-based system was also set up in this system [28]. This system had a centralized control server that managed the different agents of the system. To test the time of operation on the systems, both were tasked with completing the same orders under similar conditions of distribution of candy types. The number of candies to be packed was varied and the average time for robot operation completion was recorded for both systems. The results are summarized in the following two points:

i. With an increase in the number of orders, we observe that the self-organized version built on edge nodes is far more efficient and agile than the centralized system when the number of orders rises above 2000, as the operation completion time for the self-organized system becomes consistently lower that of the centralized system.

ii. With a stable production line, the speed of the backbone network in the centralized version was observed to be around 16 Mb/s. However, after the deployment of the self-organized system, the backbone network speed dropped to around 5-6 Mb/s which represents a 65% drop in speed.

The results of this study [28] suggest, that a decentralized and self-organizing system can become extremely useful in mass-production scenarios due to the reduced operation completion time. While the study shows that a decentralized system leads to a reduction in transmission speeds within the backbone network, the system can still function efficiently as the reduced operation completion time outweighs the drop in the backbone network speed thereby increasing the effective system throughput.

3.2 Supply Chain Management

Supply Chain Management (SCM) can be understood as a set of activities that are used to control, plan, and monitor the flow of products from their production to their distribution in the most efficient manner. While modern industries have already adopted cloud-based technologies to support their supply chains, an increasing number of these chains have begun to generate massive amounts of data from a diverse set of sensors and end devices located at different points along the supply chain. In such situations, it becomes impractical to store and process data in remote servers due to several reasons such as network bandwidth restrictions, large latency, and need for better fault tolerance. These restrictions, coupled with the proliferation of Radio Frequency Identification (RFID) technology, have given rise to edge computing-based solutions for the supply chain management.

Using the case study of a blackberry (fruit) supply chain as proposed in [29], we shall attempt to explain how industries can augment their supply chain management systems to leverage the power of edge computing. The proposed system has a three-layer architecture which is explained below:

1. **Layer 0**: This layer includes the data producing end-devices (primarily RFID embedded sensors) responsible for generating relevant data such as the Electronic Product Code (EPC), temperature, internal pressure, humidity, air quality, and other important parameters.
2. **Layer 1**: This layer is primarily responsible for monitoring and control purposes which entails the generation of actuator commands, execution of the control logic, and generation of relevant alarms. With the use of active and smart edge nodes along with onboard decision support units, this layer aid administrators in improved quality monitoring as well as in the execution of real-time corrective actions.
3. **Layer 2**: This layer consists of the traditional, centralized servers which can be used for long-term pattern recognition and analysis of offloaded sensor data,

giving valuable insights that can be useful while optimizing production and distribution pipelines.

As illustrated in the case study, the introduction of edge computing-based technology can enable efficient monitoring and actuation in all three stages of the supply chain:

- **In the field**: Edge nodes deployed at farms can aid in the real-time monitoring of blackberries. Through sensor information, the edge nodes can predict and notify farmers when the blackberries are ready for harvesting, thus improving shelf-life for the berries while also ensuring that all berries are harvested at the correct time.
- **In transit**: Edge processing nodes and sensors installed in transport vehicles can monitor various environmental parameters of berries such as temperature, relative humidity, and light. While these systems can continuously provide updates to the system managers, they can also execute instant corrective actuation methods in response to variations in environmental parameters such as controlling the air conditioning of the vehicle, adjustment of air filters, and notifying the driver about a possible opening of the vehicle doors.
- **At the packing location**: The data from the fog nodes can be used to determine the priority of cooling of incoming crates or pallets of berries which can enhance the freshness of the products while also minimizing any wastage resulting from spoilt berries.

This case study illustrates how an edge computing-based system can drastically improve the quality of monitoring for supply chains while also offering low-latency actuation techniques for system managers. Furthermore, due to the proximity of computational resources and end-devices, the amount of data transferred to the cloud servers is reduced drastically, thereby reducing the strain on the network. This leads to improved efficiency of these supply chains and while also resulting in reduced delays associated with the networks supporting these supply chains.

3.3 Food Industry

Modern food manufacturing industries have started to rely heavily on automated food production systems in factories to improve the quality and speed of production of consumable items. However, unlike other industries, the food industry constantly deals with perishable items - whether it is milk or sugar as raw materials or chocolates as finished products. Therefore, the food industry must invest in resources and systems that help in product traceability in all stages of production, processing, and distribution. These resources not only aid in the optimization of the manufacturing and distribution pipeline but also enable the industry to perform product recalls (such as in the case of some contamination) with minimal losses. In this regard, edge computing solutions have emerged as viable frameworks due to their distributed nature and the introduction of these systems can be extremely beneficial for the food manufacturing industry.

In the system proposed in [30], food manufacturing industries can rely on QR codes, barcodes, RFID tags, or transponders implanted onto objects such as primary and secondary packaging, pallets, trucks or containers, throughout the supply chain to aid in their identification and tracking along the production and supply pipeline. Edge-computing enabled sensors can be used in the process of product identification at different points along the production and supply pipelines to ensure that the flow of products is maintained. Within this system, the edge devices can rely on ad-hoc networks to communicate with each other to determine bottle-necks along the production and supply pipelines and automatically optimize these pipelines. The centralized cloud database can also be linked with this ad-hoc network and can maintain a global database of the products for administrative supervision. Therefore, with the use of such an edge-computing powered system, the industry can rely on a latency-sensitive system that functions with reduced response times, unlike a traditional cloud computing-based system.

3.4 Distributed Synchronization Services

One of the biggest use cases of cloud computing-based storage is distributed data storage, commonly referred to as cloud storage services wherein files can be accessed from anywhere on the planet by connecting a system with cloud storage servers which periodically synchronize data on different devices to enable access of files. However, even for small applications like office suite software, cloud storage services can often lead to unnecessary bandwidth consumption while also compromising latency. The EdgeCourier [31] is a file storage solution that can overcome the problems of traditional cloud computing-based distributed storage options by making use of the edge-hosted personal services (EPS) technique in conjunction with the *ec-sync* incremental synchronization approach. The essence of EPS is to make use of computational resources on the edge nodes (like access points or base stations) to provide localized services for mobile wireless users connected to these edge nodes. The *ec-sync* synchronization approach requires two participants: the *sync-sender* and *sync-receiver*, both of which are instrumental in the synchronization process which is explained as follows:

- The *sync-sender* detects if there is any document that requires synchronization with the receiver and is responsible for capturing the changes made within the document, by going through every sub-document within the document.
- To capture sub-document changes, the *sync-sender* compares two files: the edited document and the last-synced version of the same file.
- Thereafter, the *sync-sender* places the detected changes into a file known as *edit-patch*, which is transmitted to the *sync-receiver*.
- Upon receiving the *edit-patch* file, the *sync-receiver* applies the edit-patch differences to the relevant sub-documents from the last-synced version of the same file to obtain the edited document.

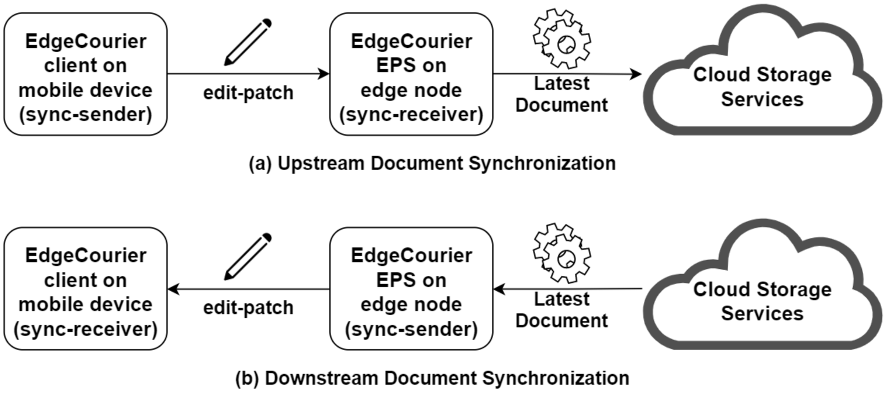
EdgeCourier client on mobile device (sync-sender)
edit-patch
EdgeCourier EPS on edge node (sync-receiver)
Latest Document
Cloud Storage Services
(a) Upstream Document Synchronization
EdgeCourier client on mobile device (sync-receiver)
edit-patch
EdgeCourier EPS on edge node (sync-sender)
Latest Document
Cloud Storage Services
(b) Downstream Document Synchronization

Fig. 4 System overview for EdgeCourier

- This edited document is then also shared with the cloud storage services to transmit it to various EPS instances or nodes across the network for global synchronization.

Furthermore, an important advantage of having different EPS instances is that they can be managed by a centralized management service (on a cloud service), which can migrate data to and from the edge nodes if needed. This, therefore, leads to better oversight and increased fault tolerance as data can be migrated to different resources for analysis or in response to outages experienced at edge nodes. The overview of the EdgeCourier system can be seen in Fig. 4. Laboratory-based studies on the Edge Courier system [31] showed that with the rise in the size of documents that need to be synchronized, the time spent on network transmission becomes notably lower for the EdgeCourier system as seen with a document size of 1 MB which takes 0.6 seconds lesser on the EdgeCourier system than on the direct sync system. Such distributed synchronization systems can be particularly useful in the software development industry for real-time code synchronization in large team projects. Similarly, the banking industry can also derive some critical applications from these systems such as in the real-time synchronization of transactions and other banking data. These examples clearly show that edge computing powered data synchronization systems find a lot of applications in modern industries that require reliable network services. As we have seen, these systems lead to reduced data transmission over the network, resulting in reduced latency and lesser strain on the network's bandwidth capabilities, hence leading to dependable network services.

3.5 Healthcare

With the recent advancements made in the domain of medical IoT devices, the healthcare industry has started to adopt IoT solutions that provide vital medical services such as the monitoring of Electrocardiogram (ECG) data and processing of

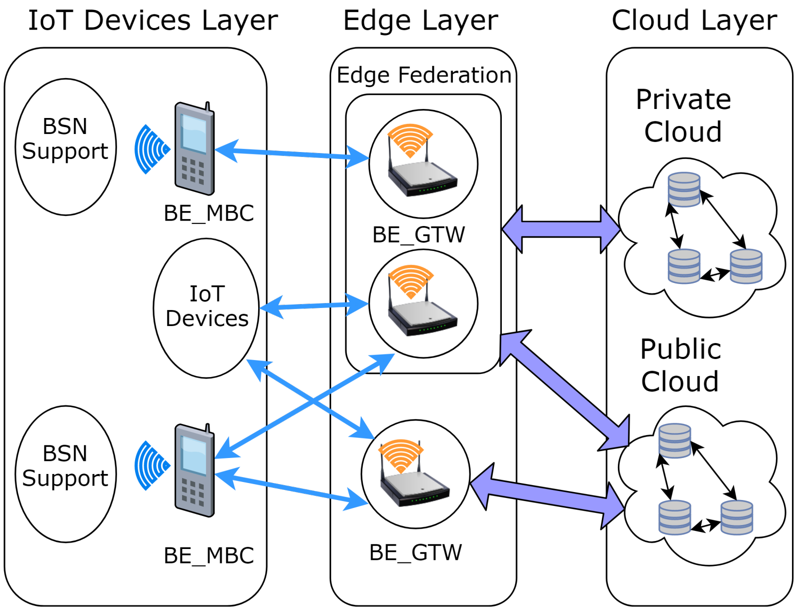

Fig. 5 The BodyEdge [32] Architecture

Magnetic Resonance Imaging (MRI) data. However, most of the traditional IoT based solutions for healthcare rely heavily on cloud-based processing as well as storage which has started to create problems for these solutions as the massive amount of data being generated is straining the communication network's capacity. This often leads to unpredictable delays in communication while also promoting increased latency in the network which can significantly impact healthcare operations within the hospital or clinic especially in time-sensitive situations that require urgent reactions such as in heart attacks or strokes. Therefore, modern medical IoT systems require a flexible multi-level network architecture that can cohesively work with heterogeneous sensors and process the relevant data with minimal latency to produce relevant results and responses. These requirements have led to the adoption of the edge computing paradigm in medical IoT systems due to the benefits it can provide in terms of reduced latency and improved reliability, both of which are critical for these systems. In this subsection, we will be reviewing the BodyEdge architecture [32] as shown in the figure below, which is structured and inspired by the edge computing paradigm to achieve the following goals:

- Reduced communication delay and latency.
- Wide support for scalability and responsiveness.
- Limited cost in terms of bandwidth for data transmission (i.e. only limited statistics data needs to be transmitted to the cloud).
- Improved Privacy (since the edge network may be interpreted as a private cloud).

This architecture consists of two complementary parts. The first is a mobile client called BodyEdge Mobile BodyClient (BE-MBC) which is primarily responsible as a relay node for communication between the sensors and the edge client using multi-radio communication technology. The second is a performing gateway known as the BodyEdge Gateway (BE-GTW), which is placed at the edge of the network and is primarily responsible for acquiring device data and locally processing it to produce valuable insights and patterns that can be relayed back to the end devices or sensors. In addition to this, the gateway also ensures communication with the cloud to allow users to maintain oversight over this system.

To validate the BodyEdge architecture, it was physically implemented in [32] and compared with a cloud-based architecture for the task of stress detection using cardiac sensors. Within the implementation, the BE-MBC module was installed on a smartwatch which was paired with a chest band to acquire ECG signals. The BE-GTW was installed on an independent hardware platform (Raspberry Pi3) as well as on an Azure cloud virtual machine to perform the comparative study. Finally, the edge-based system with the BE-GTW installed on the Raspberry Pi3 was tested on 100 athletes to determine stress levels using the Heart Rate Variability (HRV) technique [33] and the average round trip delay time (RTT) for this case was 152 ms. The same experiment was then conducted with the cloud-based system which yielded an average round trip delay time (RTT) of 338 ms. This result, therefore, corroborates our assumptions about the performance benefits offered by edge-computing-based systems in terms of reduced latency and indicates that medical IoT systems should indeed adopt edge computing-based network architectures.

3.6 Agriculture

Modern agriculture has extensively embraced automation and modern technology to improve and optimize existing agricultural processes due to the improved connectivity between agricultural resources. As technology is becoming increasingly interconnected, edge computing-based infrastructures have started to dominate most network-based applications, and to tackle the growing amount of data being generated by end devices, the agricultural industry has also started edge computing-based architectures to create latency-sensitive applications for agricultural processes. The concept of Precision Agriculture (PA) has seen a significant rise in popularity due to the improvement in sensor technologies, and several systems based on edge computing have been proposed, like the precision agriculture platform [34]. These systems make use of intelligent algorithms in conjunction with smart sensors and actuators in the field to providing real-time monitoring services that enable control services to maintain optimal environments for crop growth. In the system proposed in [34], the architecture is divided into 3 tiers namely: crop (Cyber-Physical System or CPS) tier, edge computing tier, and the cloud tier. The architecture has been illustrated in Fig 6. The crop (CPS) tier is majorly comprised of sensors that aid in real-time monitoring of various environmental parameters such as temperature, humidity, pH, CO_2 levels, solar radiation, and other important factors. In addition to sensors, this tier also supports various actuation devices such as soil nutrition pumps, valves, irriga-

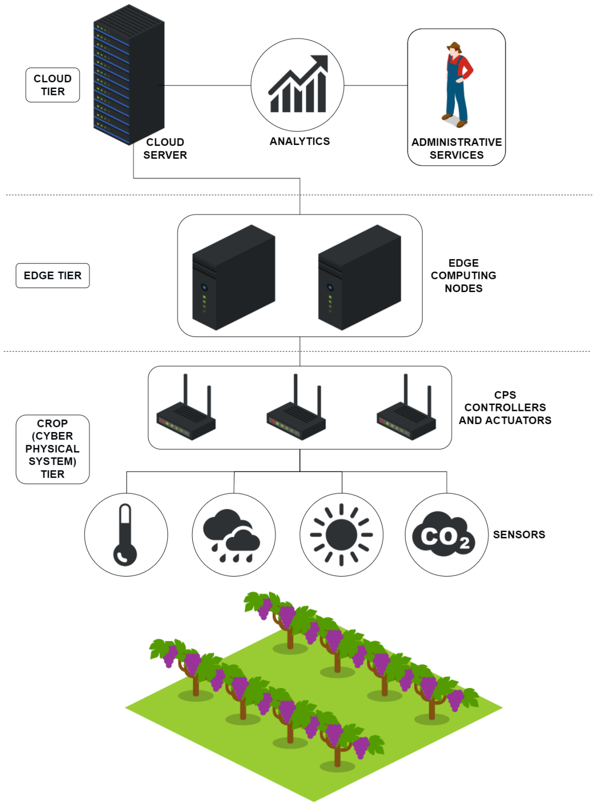

Fig. 6 Architecture Overview for Agricultural Monitoring System

tion devices, ventilation devices, and light-control devices. Within this architecture, operations at this tier require low latency and high reliability in communication so that emergency services can be enacted without human intervention, which is made possible through the edge computing-based computational nodes situated closer to the data sources. In continuation, edge nodes within the edge computing tier are responsible for executing commands through actuation devices based on inputs received from sensor networks in the crop tier. Therefore, this layer is responsible for the control of irrigation, climate control, nutrition control, and other auxiliary tasks like alarm and energy management. Finally, the cloud tier is responsible for long-term data analytics and system management services. The physical implementation of this system showed savings of more than 30% in terms of water consumption along with savings of nearly 80% in terms of some soil nutrients when compared with a regular open crop. In addition to environmental monitoring, edge computing powered systems can also be employed for video analytics through UAVs that can help farmers in optimized weeding and harvesting. This clearly illustrates the impact of automation on the agricultural industry and shows how edge computing-based architectures can replace cloud computing frameworks especially in applications that require low-latency and high reliability.

4 Industrial Applications of Fog Computing

4.1 Smart Grids

Conventional energy grid systems have been powering industries and countries for the past 100 years, and with the tremendous rise in demand for electrical power, the domain of IoT has emerged to be the pioneering technology that is leading developments in the smart grid systems. Traditional grid operations relied on simple analog meters to record units of power flowing per month to each household or industry, but with the evolution of intelligent and autonomous systems, modern smart grids offer solutions that allow comprehensive oversight over energy distribution which is beneficial to both consumers and producers. For power producers, these smart grid solutions allow accurate monitoring of energy demands and supplies which allows them to effectively control pricing as well as load balancing to sustain the healthy functioning of the grid. On a similar note, consumers can monitor their energy consumption in real-time for each device which allows them to effectively and reliably manage their energy spending. The framework of such a smart grid, therefore, involves a heavy dependence on the collection and aggregation of real-time data from every device within each household or industry that is powered by the grid. This will inevitably lead to the generation of a large amount of data that needs to be efficiently managed and analyzed while maintaining the security of the data. To manage such massive amounts of data, it is easy to perceive that the cloud computing paradigm cannot be a viable network architecture for these IoT powered smart grid solutions since the sheer volume of the data would not adhere to any conventional network's transmission capacity. To reduce the strain on the network capacity, fog computing-based grid systems can become a viable option since the fog computing architecture allows computational offloading from the centralized cloud servers to fog nodes that are situated closer to the end devices. This distributed nature allows the network to function with low latency and improved reliability while also maintaining data security, and these are exactly the properties that a modern smart grid system requires.

The basic architecture of smart grid systems is generally composed of advanced metering infrastructure (AMI) along with area networks, data centers, and integrated substation centers. Within this architecture, AMI ensures two-way communication is maintained between the end devices and the fog nodes which leads to a secure, reliable, and cost-effective service. The model proposed in [35] is a three-tier architecture as shown in Figure 7.

The first tier is comprised of the smart meters which are responsible for collecting data regarding energy consumption as well as for inter-tier and intra-tier communication. The second tier comprises the resource-rich fog nodes which are responsible

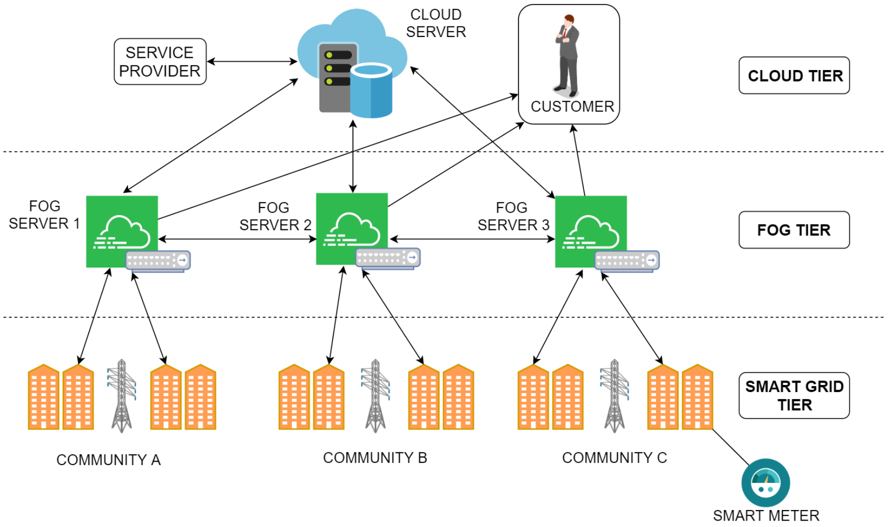

Fig. 7 Structure of Fog Computing enabled Smart Grid

for delivering the majority of computational services to the network. Finally, the third tier comprises the traditional cloud servers which are usually responsible for oversight and maintenance of the entire grid. This structure allows inter-tier communication within the first and second tiers which enables different geographical sub-grids to communicate with one another.

Through the following points, we can appreciate the benefits offered by fog computing architecture:

- The smart metering technology enables the energy producers to monitor power loads in real-time which helps them in drafting an effective load-balancing methodology, with extremely low latencies and transmission delays.
- The smart meters allow consumers to monitor the energy consumed by each device in real-time and this can aid them in controlling device usage dynamically to minimize their energy costs.
- While the smart meters maintain a local database of the profile of energy consumed by each device, they aggregate this data for the complete household or industry and forward this encrypted aggregate to the fog servers. These fog servers can then store this data securely within storage systems that are localized in that geographical area, and because the encryption key is only known to the fog node and the respective smart meter, the system maintains privacy even if the data is accessed by someone through the cloud server.
- Finally, the varied geographical location of fog computing nodes can be beneficial to the grid in an interesting way: specifically for the case of electric vehicles which can be charged at any location inside the grid while the grid maintains the correct billing information. For instance, if an electric vehicle is charged in any neighbourhood, the smart meter deployed in that neighbourhood can identify

the owner of the car using a unique identifier and transfer the billing information via the fog node tier to the owner's smart meter, thereby ensuring consistency in billing within the smart grid.

4.2 Satellite Communication

With the recent advances made in satellite technology, the communication industry has started to rely heavily on satellites to provide access to people situated in remote locations. Satellite-Terrestrial Networks (STN) are communication networks that have emerged as one of the most promising low-cost technology which can lead to ubiquitous access to internet connectivity across the globe. A majority of STN setups make use of Low Earth Orbit (LEO) satellites to provide connectivity to sparsely distributed users by interconnecting small terrestrial terminal stations which are placed in remote locations to ensure maximum area coverage, as shown in Figure 8. But, with the evolution of smartphones and tablets, the amount of data that

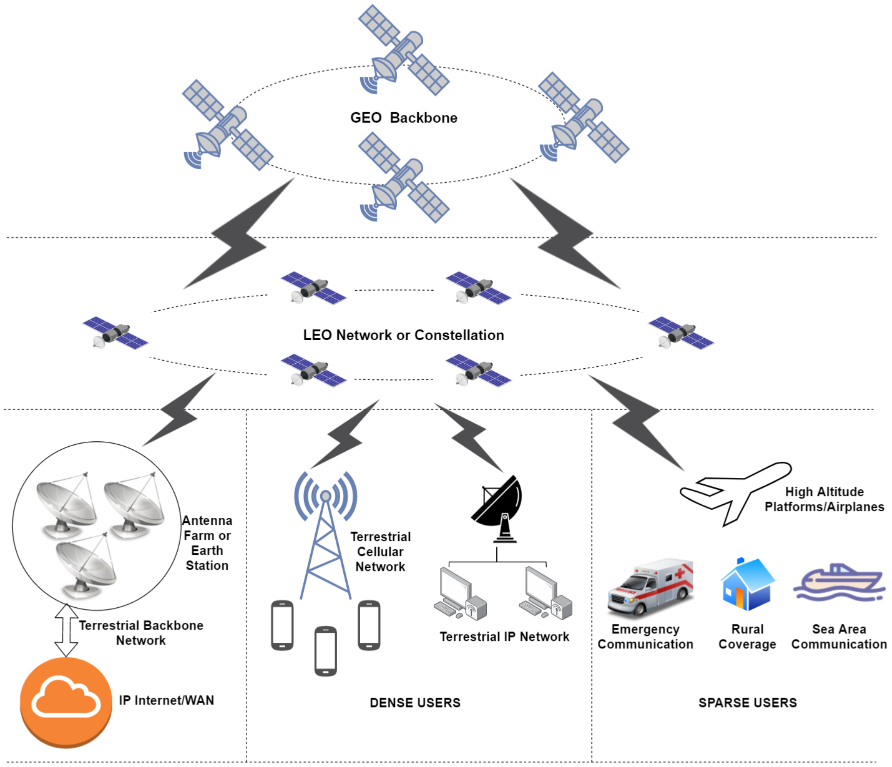
GEO Backbone
LEO Network or Constellation
Antenna Farm or Earth Station
Terrestrial Backbone Network
IP Internet/WAN
Terrestrial Cellular Network
Terrestrial IP Network
DENSE USERS
High Altitude Platforms/Airplanes
Emergency Communication
Rural Coverage
Sea Area Communication
SPARSE USERS

Fig. 8 Traditional Satellite Terrestrial Network

needs to be transferred across the network has increased drastically, particularly because of an increase in the number of applications such as speech recognition and gaming that make use of cloud services to process user-generated data. This puts a strain on the network's data transfer capacity, and so we must look towards computational offloading to help alleviate this problem. In this situation, satellite mobile edge computing (SMEC) [42] can be a possible solution that can offload computation as well as storage to local servers, thereby leading to an improved QoS, increased reliability, and reduced latency. This technology, although dubbed as edge computing, is better classified as a fog computing-based technology as computational resources are essentially an extension of the cloud servers. Therefore, the introduction of fog computing resources near the end-devices can lead to content caching and other storage facilities which effectively reduces that traffic in the overall STN. In terms of computational offloading, the fog sites can be located at 3 different locations, and these are:

- **Proximal Terrestrial Offloading (PTO)**: In this situation, satellite mobile fog computing servers are deployed at terrestrial stations, as shown in Figure 9 (b). The advantage of this system is that the communication latency is significantly reduced because backhaul transmission through the satellite is avoided. While such a system would be extremely useful for terrestrial terminal stations that cater to dense user areas, it would not be practical for terrestrial terminal stations that are placed in spare user areas especially because these stations do not hold extra computational facilities and are remote.
- **Satellite Borne Offloading (SBO)**: In this situation, the satellite mobile fog computing servers are deployed in LEO satellites, as shown in Figure 9 (c). With this network extension, both sparse and dense users will benefit from reduced latencies while the traffic in the terrestrial backbone network will also reduce significantly. However, the latency in this situation would be higher than that of PTO and it would significantly increase the power consumption of satellites which will be performing the offloaded computations which will not be practical for satellites with limited power sources.
- **Remote Terrestrial Offloading (RTO)**: In this situation, the satellite mobile fog computing servers are deployed to the terrestrial backbone network, as shown in Figure 9 (d). In this situation, the delays in transmission over the WAN IP that connects with the Remote Cloud servers can be avoided and this translates to a reduced latency when compared to the situation with no edge computing offloading. The latency in this network scheme is higher than PTO and SBO, but it is the most practical scheme to implement and maintain.

4.3 Manufacturing Process Monitoring

With rapid globalization, industries across the globe have started to adopt modern process control systems which rely heavily on sensor networks that efficiently monitor

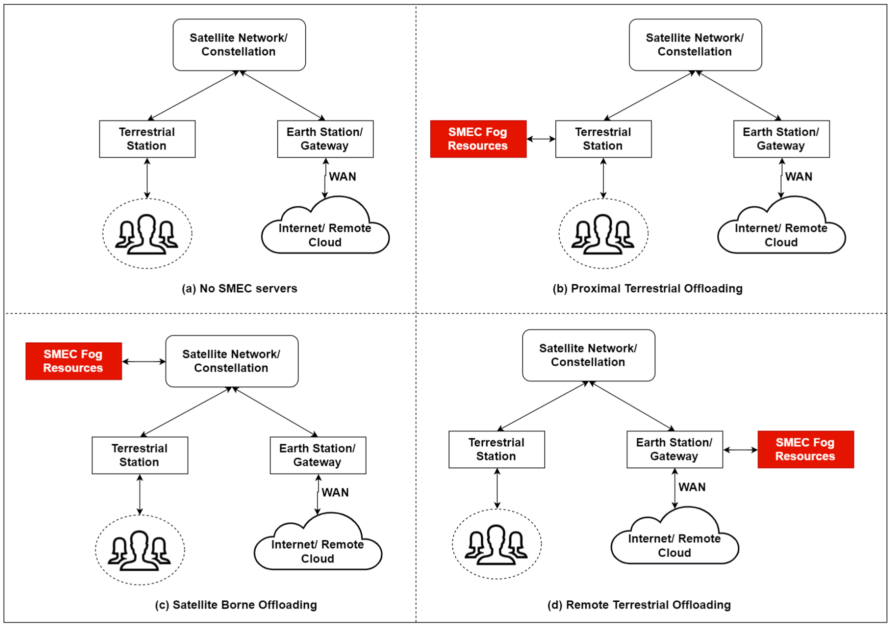

Fig. 9 SMEC with offloading at different fog sites

production lines and processes while collecting valuable data which can be used to identify faults before they occur while also aiding in optimization efforts to improve the throughput and performance of the industry. In this regard, we shall be looking at a fog computing-based framework for process monitoring in different production environments. The proposed system architecture in [40] is described sequentially:

- Step 1: Collect machine data from the production environment that streams real-time data from various sensor networks and communication adapters that function on protocols such as Simple Object Access Protocol (SOAP), MTConnect, and Open Platform Communications Unified Architecture (OPC UA).
- Step 2: Stream the raw data to a private computational fog node that is responsible for real-time monitoring and providing time-sensitive control signals to the production environment. This allows the system to function with low response times, improves reliability, and reduces the strain on the network's capacity as data is processed in a fog computing node that is situated close to the production environment.
- Step 3: Also, various samples from this data can be sent to high-performance cloud data centers which can be used to build models for predictive maintenance and process optimization. Since these samples are small in size and sporadically transferred to the cloud, the strain on the network's capacity is minimal while the models built with the sampled data can be extremely beneficial for the industry in terms of improved throughput and reduced unplanned downtimes.

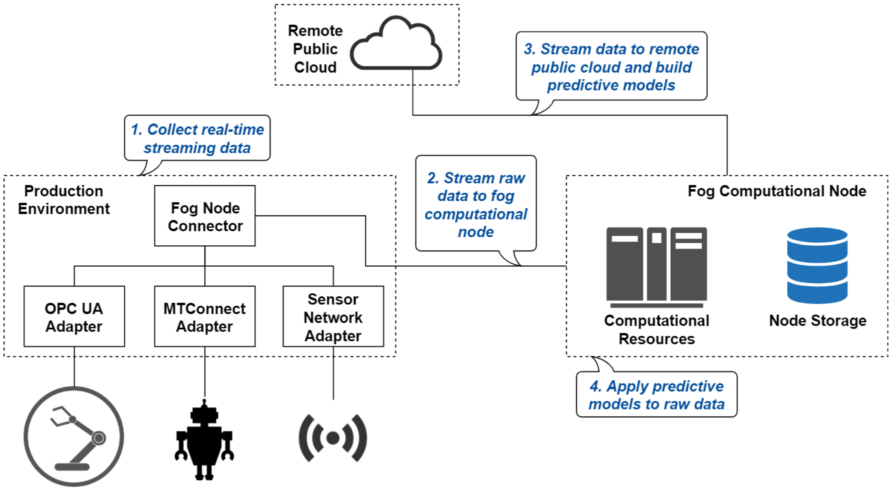
Remote Public Cloud
3. Stream data to remote public cloud and build predictive models
1. Collect real-time streaming data
2. Stream raw data to fog computational node
Production Environment
Fog Node Connector
OPC UA Adapter
MTConnect Adapter
Sensor Network Adapter
Fog Computational Node
Computational Resources
Node Storage
4. Apply predictive models to raw data

Fig. 10 Architecture for the Process Monitoring System

- Step 4: Apply these predictive models to raw data and obtain tangible insights into the production environment's real-time health and performance.

5 FUTURE RESEARCH DIRECTIONS

The edge and fog computing paradigms are considered as powerful extensions to the cloud computing paradigm, however, they face some common challenges [16] that are yet to be addressed. In this section, we describe some of the major issues and challenges faced by these paradigms as well as some potential research directions for these paradigms.

5.1 Programmability and Task Partitioning

In the traditional cloud computing-based architectures, users generally program their back-end applications on an abstract platform, without worrying about the exact configuration of the cloud server. The benefit of this abstraction is programmability since the user is not aware of the exact configuration of the platform which means that the cloud service providers can easily compile the application and run it on a single runtime of the cloud server which can have a variable configuration, unknown to the user. However, with the rise of the edge and fog computing paradigms, back-end processing is distributed across an array of distributed computational nodes - all of which can have slightly different run-times. This creates interesting and challenging problems for system designers, who need to design optimized methods for distributing computation as well as storage across nodes while making sure that synchronization processes do not impact the network's transmission capacities and ensure low latency in intra-network transmissions.

An important issue that arises with the evolution of distributed computing paradigms like edge and fog computing, is the issue of task partitioning. Within these paradigms, it is imperative that the system design takes into account the optimization of task partitioning and process scheduling, to facilitate concurrent execution across distributed nodes. An optimized task partitioning scheme allows the system to autonomously locate edge or fog nodes in real-time, and intelligently allocate tasks to these nodes, while taking into account various factors such as the computational power associated with the nodes as well as the associated overheads involved in exchanging data between these nodes.

In consideration of these issues, system designers should also think about control - whether the system should allow users to implicitly or explicitly control computational resources. In case of implicit control, which can be seen in the case of Amazon's Lambda@Edge [43], where the users need not worry about server administration, as the web services are responsible for running and scaling the application at resources available closest to the end-users. This leads to reduced complexity of programming for the users, while giving system administrators greater control over the network. In contrast, explicit control of the network gives users greater flexibility in terms of resource allocation, which can often lead to improved efficiency and increased reliability. This explicit control, however, comes at the cost of increased complexity in terms of programmability and goes to illustrate how designers need to make trade-offs while planning the layout of edge and fog computing-based systems.

5.2 Security and Privacy

With an increased interest in the edge and fog computing paradigms, people have started to appreciate the capabilities of these paradigms which enable the extension of storage, networking, and processing resources of cloud computing servers toward the edge of the network. However, with this rise in flexibility and distribution leads to many security and privacy concerns [44] that must be addressed by system designers. After analyzing several different aspects of network security, we can summarize the major security and privacy concerns as follows:

1. **Trust and Authentication**: Edge and Fog Computing-based networks are expected to provide secure and reliable services to all users and this leads to an important requirement in that all devices on the network should be able to trust one another. Therefore, trust plays a two-way role within edge and fog computing-based networks. This implies that fog or edge nodes that offer services to the network must be in a position to validate whether the resources requesting these services are indeed genuine. Similarly, edge or fog nodes that are transmitting data to or requesting services from network resources should also be able to verify whether these resources are genuine or not. These concerns have given rise

to various authentication mechanisms that can be used to authenticate network resources before transmissions and requests. Systems can employ mechanisms such as permissioned blockchain networks like TrustChain [46] for authentication, cryptographic authentication schemes like SAKA-FC [47], and hardware-based authentication schemes like Physically Unclonable Functions (PUF) [48], to authenticate network resources.

2. **Integrity**: Edge and Fog Computing systems should always ensure that data transmission within the network should be done securely so that transmitted data is not altered or modified by attackers. The most prominent method to ensure the integrity of data in networks is through cryptographic signature verification systems like the GNU Privacy Guard (GPG) system [49] which is used to digitally sign transmitted data. The received data is then verified at the receiving station to establish the integrity of the data, which is extremely important in edge and fog computing-based systems as they rely heavily on intra-network data transfers due to their distributed topology.
3. **Availability**: The availability of information refers to the ability of the system to ensure that authorized parties can access relevant information whenever needed. The biggest concern concerning the availability of information is Denial of Service (DoS) attacks that hamper or eliminate accessibility to information. Edge and Fog Computing-based systems are generally well equipped to handle DoS attacks since these systems have distributed computational resources, however, Distributed Denial of Service (DDoS) attacks can still impact these systems and to protect networks or applications against DoS attacks, designers often make use of smart DNS resolution services, Web Application Firewalls (WAF), and other intelligent traffic management techniques to ensure service security.
4. **Confidentiality**: The confidentiality of information represents the ability of the system to protect information from being disclosed to unauthorized parties. This implies that edge and fog computing paradigms should ensure that information is stored securely to prevent data leaks, which is especially likely due to the distributed architecture of these paradigms. Edge and Fog Computing-based architectures often use homomorphic encryption schemes as well as cryptographic hashing techniques to store confidential data at different distributed locations within the network. Due to the use of these techniques, even if attackers can gain access to secure databases, they will not be able to understand the data as it will be in an encrypted format.
5. **Data Ownership**: This issue extends from the fact that unlike cloud computing-based systems, edge and fog computing-based systems store data in distributed locations across the network which means that the system can store data locally at the computational nodes, thereby providing complete access and ownership to the end-users. However, these paradigms often involve the transmission of data between nodes especially when processing or computations have been offloaded to different nodes on the network, and this creates a problem in the data ownership. Thus system designers should take this behaviour into account while drafting the privacy policy of the network. This also involves thinking about legal jurisdictions, such as when data crosses international borders, it may be subject

to different regulations. This means that data transfer methods should consider the compatibility of data with two different data regulation policies for the source and destination.

5.3 Blockchain for IIoT

In recognition of the prevalent security concerns within the domain of edge and fog computing, researchers have started looking into Blockchain as a potential solution for these security concerns [45]. Following this technology's conceptualization in 2008, industries across the world have extensively adopted it for several applications such as in the authentication of financial transactions, in the preservation of digital contracts, and in the identification of agents in distributed systems, making it an extremely viable solution for the security concerns of edge and fog computing.

The essence of the blockchain technology lies in the decentralized transaction ledger that maintains the record of all exchanges and transactions. The most essential components of the distributed ledger are the blocks that comprise the blockchain, where each block acts as a record of the transactions or exchanges which are being monitored by the blockchain. To preserve the authenticity of transactions and exchanges, blockchains rely on two main aspects of the blockchain framework. First, each block within the ledger possesses a cryptographic hash that uniquely identifies that block. This hash acts as a fingerprint that is referenced by the next block after being added to the ledger. This reference-linking structure allows the distributed ledger to easily identify distortion within the blockchain, since changing the transaction within any block in the blockchain leads to change in the block's unique cryptographic hash, thereby leading to a break in the chain as the references for each of the following blocks in the ledger need to be changed to remake the blockchain. Second, each transaction is verified by different agents that maintain independent copies of the ledger, which makes tampering even more difficult as any unwarranted changes in the ledger by one agent can be easily identified by other agents through a block-by-block comparison. These two aspects allow the blockchain to effectively control distortion to maintain a transparent and verifiable transaction history.

The tamper-proof nature of blockchain can be extremely useful in addressing the trust and authentication problems of edge and fog computing. Specifically, these paradigms can adopt blockchain-based authentication systems such as BSeIn [23] that allows fine-grained access control, user anonymity, and mutual authentication while allowing networks of these paradigms to scale as usual. Similarly, the tamper-proof nature of the blockchain ledger allows the system to detect adversarial end devices which may be attempting to manipulate raw sensor data, allowing the system to take immediate countermeasures. Furthermore, the distributed nature of the edge and fog computing paradigms match the decentralized nature of the blockchain technology, making such systems resistant to node failures as the loss of any one node does not compromise the health of the complete system as data remains distributed across the network.

Although the blockchain technology is relatively new, several interesting systems have been proposed in the past, which have integrated the blockchain technology with IIoT architectures, to improve the security and reliability of such systems. In the BPIIoT platform [36], systems can make use of a peer-to-peer network as well as smart contracts to allow end devices to share information across the network after successfully verifying the authenticity of their counterparts within the network. This enables machine-to-machine (M2M) communication that is safe, transparent, and verifiable thereby improving the security of systems that utilize this technology. Similarly, to design edge and fog computing systems that are suited for distributed data storage, the blockchain technology can be extremely useful in verifying device identity, while advanced encryption schemes can be used to encrypt transactions and store them within the blockchain in a chronological fashion. In such a situation, if an adversarial end device attempts to alter the blockchain, it will be identified immediately by the other agents in the system while the chronological arrangement of transactions can be used to determine the exact time of the attack by the adversary - leading to easier data recovery. Similarly, edge and fog computing-based IIoT systems that handle a large variety of goods, identities, and credentials, can use blockchains to store information relating to these domains. Furthermore, for physical assets such as end devices, the cryptographic hashes of the device firmware can be stored in a separate private blockchain to ensure the authenticity of the hardware within the system.

While the integration of the blockchain technology with edge and fog computing-based IIoT systems is a viable future direction, we must understand at the existing limitations of the blockchain technology and work to improve the effectiveness of this technology. These limitations include:

1. **Poor performance on scaling**: In contrast with traditional centralized databases, blockchains tend to slow down with scale due to the complexity of the consensus mechanisms leading to lower transactional throughput and increased latency.
2. **Energy inefficient algorithms**: The complexity of proof-of-work (PoW) increases as the number of transactions conducted increases which makes these algorithms extremely energy-hungry leading to several issues on battery-powered IoT devices which are power-constrained and cannot afford to make such computations.
3. **Lack of flexible test platforms**: It is imperative to have flexible test platforms in place that allow people to experiment with different configurations of blockchains in various IIoT applications to test the stability, performance, scalability, and security of these applications.

5.4 Virtualization

Due to the resource-constrained nature of fog and edge devices, most applications utilizing these distributed computing paradigms need to run multiple operating environments on a single edge or fog device. Typically, each edge or fog device needs to run two different environments for different users, leading to two important requirements in this respect. First, we require separation of services, wherein different tasks and user environments must be maintained separately on every node within the system. Second, we want to ensure application fairness, wherein resource allocation and distribution of computational power should be monitored by resource management algorithms that enforce fairness in allocation procedures.

In this regard, virtualization technologies for task partitioning can be viable options for the encapsulation of services from various users and applications into separate Virtual Machines (VM). Through the use of VMs, virtualization technologies can run different operating environments on each node. While extensively developed virtualization technologies for cloud computing exist [41], their support for edge and fog based systems remains limited. Therefore, within the field of virtualization for edge and fog computing paradigms, there are many standing issues and challenges, including:

1. **Service Encapsulation**: In traditional cloud-oriented virtualization, techniques for service encapsulation to VMs tend to be resource-intensive which makes them infeasible for virtualization at edge and fog layers. An interesting approach can be through the deployment of services in containers [37], [38], [39]. In this approach, we use containers as operating system-level virtualization objects to execute different services on resource-constrained computational nodes, allowing service encapsulation and improved task allocation procedures.
2. **Resource or Container Allocation**: The allocation of resources in the form of containers running on different nodes requires sophisticated allocation algorithms to ensure virtualization in edge and fog computing systems. These virtualization algorithms should be able to perform resource (computational capacity) estimation and overhead estimation in real-time to identify the optimal strategy for task partitioning. Furthermore, these algorithms should not be resource-intensive and must be able to aggregate the results obtained from distributed computations to accomplish the overall task, thereby leading to efficient virtualization.
3. **Container Migration**: As seen in the earlier sections, the distributed and decentralized nature of the edge and fog computing paradigms enables us to design fault-tolerant and reliable systems. To maintain this fault-tolerance, virtualization techniques must be responsive enough to ensure the migration of services to other containers (nodes) in response to the failure of nodes that are currently executing tasks or applications. Therefore, these virtualization techniques should be able to identify, allocate, and migrate tasks quickly, so that the system experiences minimal downtime while maintaining a reliable service. A robust container migration policy will be similarly beneficial for systems under dynamic workloads and will improve the efficiency of these systems.

Given the success of virtualization techniques in the scenario of traditional cloud computing services, the development of robust virtualization techniques for edge and fog computing-based IIoT systems can become an influential force in the future development of these paradigms.

5.5 Resource Allocation

We already know that edge and fog nodes are constrained in terms of computational resources, memory, network elements, and energy, and therefore, efficient management of resources is imperative for the success of these paradigms and to achieve efficient resource management, resource allocation mechanisms should work in conjunction with virtualization techniques [20]. To facilitate the process of task partitioning through virtualization, resource estimation mechanisms, running as software middleware, should be able to accurately estimate the various resources available with the different edge or fog nodes in the system and must be able to do so with low computational overhead. Similarly, resource allocation mechanisms should work to ensure the highest QoS for the end-user, and at the same time, it should ensure fairness of allocation for distinct services, wherein tasks with higher priority (like real-time content streaming) should receive larger bandwidth when compared with lower priority tasks. Finally, resource allocation should also take care of dynamic network situations where IoT devices enter and exit the network at will, making technologies such as software-defined networking (SDN) important factors in system design. SDN can aid the system in the management of dynamic network resources to ensure continued connectivity between them while also improving oversight and control. By working towards the improvement of resource allocation mechanisms, system designers can establish better virtualization techniques which, in turn, will lead to the superior edge and fog computing-based IIoT systems.

5.6 System Metrics

While there exists a large variety of advantages that arise due to the architecture of the edge and fog computing paradigms, there are some associated metrics that also need to be considered while designing these paradigms. Importantly, system designers often deal with the design of policies which govern task partitioning and work offloading from one computational node to others, and in such situations, they should give importance to the following metrics:

1. **Energy**: Edge and fog nodes often consist of embedded devices such as wireless access points, routers, or switches, which often have power sources in the form of batteries. Due to the limited capacity of the batteries, system designers should always consider if it would be energy efficient to offload some task to a particular node, while also taking into account the computational power associated with that node and the expected amount of computation that is required for the task being offloaded. An important environmental benefit in this regard is that the energy requirement of fog and edge nodes is smaller than that of cloud servers. This means that the edge and fog nodes can use renewable energy sources for their power requirements, leading to an overall reduction in CO_2 emissions,

which shows that the edge and fog computing paradigms are also much more eco-friendly when compared to cloud computing.

2. **Cost**: While migrating applications to edge and fog computing-based architectures often leads to reduced latency, improved reliability, and increased fault tolerance, it still comes at the expense of increased cost. With thousands of embedded computational nodes in modern edge and fog computing-based systems, the cost is generally much higher than traditional cloud services, which means that systems within the edge and fog computing paradigms should be cost-efficient, to justify their development in response to improved user experience.
3. **Bandwidth**: The edge and fog computing paradigms need to be designed while considering bandwidth especially for low-cost systems which generally have low bandwidths within the network. In the edge and fog computing paradigms, we see a lower amount of data transmission whenever a larger amount of data is processed closer to the edge since no data needs to reach the remote cloud server. However, the distributed nature of the system can often increase the number of transmissions within the system, especially in co-operative systems that rely heavily on inter-node communication. Therefore, system designers can appreciate these major factors that influence bandwidth consumption and can organize their system accordingly.

6 Summary and Conclusions

With the recent advances within the domain of IIoT, people have started to observe strong trends that indicate rapid growth in the number of smart devices connected to IIoT networks and this growth cannot be supported by traditional cloud computing platforms. In response to this, edge and fog computing systems have emerged as important frameworks that have the potential to support the growing demands of automation in different industrial settings. As these paradigms are inherently distributed in nature, their resources are distributed along the edges of the network. This in turn leads to reduced latency and improved reliability of services associated with edge and fog computing-based systems. Through this chapter, we have described the fundamentals of the edge and fog computing paradigms while comprehensively exploring the benefits offered by these systems over the traditional cloud-based platforms. Furthermore, the chapter discusses several industrial applications for both edge and fog computing through an in-depth analysis of proposed system architectures for the different industrial use cases. With several supporting case studies and experiments explained in the chapter, we practically demonstrate the superiority of these computing paradigms and build a strong case for the adoption of these paradigms in modern industrial systems. Finally, we present the major issues and challenges faced by these paradigms, along with some plausible solutions which serve as future research directions.

References

1. Gubbi J, Buyya R, Marusic S, Palaniswami S (2003) Internet of Things (IoT): A vision, architectural elements, and future directions. Future Gener Comp Sy 29(7):1645-1660.
2. Cisco Systems Inc. (2019) Cisco Visual Networking Index: Forecast and Trends, 2017–2022. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf Accessed: 19 June 2020.
3. Manyika J, Michael Chui M (2015) Open interactive popup McKinsey Global Institute. The Internet of Things: Mapping the value beyond the Hype. https://www.mckinsey.com/mgi/overview/in-the-news/by-2025-internet-of-things-applications-could-have-11-trillion-impact Accessed: 19 June 2020.
4. Sisinni E, Saifullah A, Han S, Jennehag U, Gidlund M (2018) Industrial Internet of Things: Challenges, Opportunities, and Directions. IEEE Trans. Industr. Inform. 14 (11):4724-4734.
5. Lu Y (2017) Industry 4.0: A survey on technologies, applications and open research issues.J. Ind. Inf. Integr. 6:1-10.
6. Cisco Systems, Inc. (2014) Mining firm quadruples production, with Internet of Everything. https://www.cisco.com/assets/global/BE/tomorrow-starts-here/pdf/c36-730784-01_dundee_precious_metals_cs_v3a_en_be.pdf Accessed: 28 November 2019.
7. Tzounis A, Katsoulas N, Bartzanas T, Kittas C (2017) Internet of Things in agriculture, recent advances and future challenges. Biosyst. Eng. 164:31-48.
8. Xu B, Xu L D, Cai H, Xie C, Hu J, Bu F (2014) Ubiquitous Data Accessing Method in IoT-Based Information System for Emergency Medical Services. IEEE Trans. Industr. Inform. 10(2):1578-1586.
9. GSMA (2018). China Mobile Electric Smart Metering – Internet of Things Case Study. https://www.gsma.com/iot/wp-content/uploads/2018/03/iot_china_mobile_metering_04_18.pdf Accessed: 19 June 2020
10. GSMA(2018). China Mobile Smart Parking – Internet of Things Case Study. https://www.gsma.com/iot/wp-content/uploads/2018/03/iot_china_mobile_parking_04_18.pdf Accessed: 19 June 2020
11. Fraga-Lamas P, Fernández-Caramés TM, Castedo L (2017) Towards the Internet of Smart Trains: A Review on Industrial IoT-Connected Railways. Sensors 17(6):1457.
12. Shah S, Ververi A (2018) Evaluation of Internet of Things (IoT) and its Impacts on Global Supply Chains In: Proceedings of the 2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Marrakech, Morocco, pp. 160-165.
13. Antão L, Pinto R, Reis J, Gonçalves G (2018) Requirements for Testing and Validating the Industrial Internet of Thing In: Proceedings of the IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Vasteras, pp. 110-115.
14. Breivold HP, Sandström K (2015) Internet of Things for Industrial Automation – Challenges and Technical Solutions In: Proceedings of the IEEE International Conference on Data Science and Data Intensive Systems, Sydney, pp 532-539.
15. Chamola V, Tham C, Chalapathi GSS (2017) Latency aware mobile task assignment and load balancing for edge cloudlets In: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, Kona, HI, pp 587-592.
16. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge Computing: Vision and Challenges. IEEE Internet Things J. 3(5) pp 637-646.
17. Mahmud R, Kotagiri R, Buyya R (2018) Fog Computing: A Taxonomy, Survey and Future Directions. In: Di Martino B., Li KC., Yang L., Esposito A. (eds) Internet of Everything(Algorithms, Methodologies, Technologies and Perspectives) Springer, Singapore, pp 103-130.
18. Vaquero L (2014) Finding your way in the fog: Towards a comprehensive definition of fog computing. ACM SIGCOMM Computer Communication Review: 44(5):27–32.
19. Yousefpour A, Fung C, Nguyen T, K. Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: A complete survey. J. Sys. Arch. 98:289-330.

20. Ahmed A, Arkian H, Battulga D, Fahs A, Farhadi M, Giouroukis D, Gougeon A, Gutierrez F, Pierre G, Souza Jr P, Ayalew Tamiru M, Wu L (2019) Fog Computing Applications: Taxonomy and Requirements. https://arxiv.org/pdf/1907.11621.pdf.Accessed: 19 June 2020.
21. Xiao Y, Jia Y, Liu C, Cheng X, Yu J, Lv W (2019) Edge Computing Security: State of the Art and Challenges. Proc.of the IEEE 107(8): 1608-1631.
22. Lyu L, Chen C, Zhu S, Cheng N, Yang B, Guan X (2018) Control Performance Aware Cooperative Transmission in Multiloop Wireless Control Systems for Industrial IoT Applications. IEEE Internet Things J. 5(5):3954-3966.
23. Lin C, He D, Huang X,Choo KR, Vasilakos AV (2018) BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. J. Netw. Comput. Appl. 116:42-52.
24. Chen B, Wan J, Celesti A, Li D, Abbas H, Zhang Q (2018) Edge Computing in IoT-Based Manufacturing. IEEE Commun. Mag. 56:103:109
25. Wan J, Tang S, Shu Z, Li D, Wang S ,Imran M, Vasilakos AV (2016) Software-Defined Industrial Internet of Things in the Context of Industry 4.0. IEEE Sens. J. 16(20):7373-7380.
26. Gîrbea A, Nechifor S, Sisak F, Perniu L (2011) Design and Implementation of an OLE for Process Control Unified Architecture Aggregating Server for a Group of Flexible Manufacturing Systems. Software Lett 5(4):406-414.
27. Kang W, Kapitanova K, Son SH (2012) RDDS: A Real-Time Data Distribution Service for Cyber-Physical Systems. IEEE Trans. Industr. Inform., vol. 8, no. 2, pp. 393-405, May 2012.
28. Wang S, Wan J, Zhang D, Li D, Zhang C (2016) Towards Smart Factory for Industry 4.0: A Self-Organized Multi-Agent System with Big Data Based Feedback and Coordination. Comput. Netw. 101:158-168.
29. Musa Z, Vidyasankar K (2017) A Fog Computing Framework for Blackberry Supply Chain Management. Procedia Computer Science 113:178-185.
30. Industrial Internet Consortium White Paper (2018) Introduction to Edge Computing in IIoT. https://www.iiconsortium.org/pdf/Introduction_to_Edge_Computing_in_IIoT_2018-06-18.pdf
31. Hao P, Bai Y, Zhang X, Zhang Y, (2017) Edgecourier: an edge-hosted personal service for low-bandwidth document synchronization in mobile cloud storage services In: Proceedings of the Second ACM/IEEE Symposium on Edge Computing (SEC '17). , San Jose / Silicon Valley, CA, pp 1-14.
32. Pace P, Aloi G, Gravina R, Caliciuri G, Fortino G, Liotta A (2019) An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0, IEEE Trans. Industr. Inform. 15(1): 481-489.
33. Bernardi L, Wdowczyk-Szulc J, Valenti C, Castoldi S, Passino C, Spadacini G, Sleightp (2000) Effects of controlled breathing, mental activity and mental stress with or without verbalization on heart rate variability. J. Amer. College Cardiol. 35(6) 1462-1469.
34. Zamora-Izquierdo MA, Santa J, Juan A. Martínez JA, Martínez V, Skarmeta AF (2019) Smart farming IoT platform based on edge and cloud computing," Biosyst. Eng. 177:4-17, 2019.
35. Okay FY, Ozdemir S (2016) A fog computing-based smart grid model In: Proceedings of International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, pp 1-6.
36. Bai L, Hu M, Liu M, Wang J (2019) BPIIoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT. IEEE Access 7:58381-58393.
37. Kaur K, Dhand T, Kumar N, Zeadally S (2017) Container-as-a-Service at the Edge: Trade-off between Energy Efficiency and Service Availability at Fog Nano Data Centers. IEEE Wirel Commun 24(3):48-56.
38. Yin L, Luo J, Luo H (2018) Tasks Scheduling and Resource Allocation in Fog Computing-based on Containers for Smart Manufacturing. IEEE Trans. Industr. Inform. 14 (10):4712-4721.
39. Santoro D, Zozin D, Pizzolli D, De Pellegrini F, Cretti S (2017) Foggy: A Platform for Workload Orchestration in a Fog Computing Environment In Proceedings of the IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp 231-234.

40. Wu D, Liu S,Zhang L, Terpenny J, Gao RX, Kurfess T, Guzzo JA (2017) A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing. J. Manuf. Syst. 43:25-34.
41. Nurmi D, Wolski R, Grzegorczyk C et al (2009) The Eucalyptus Open-Source Cloud-Computing System In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai pp 124-131.
42. Zhang Z, Zhang W, Tseng F (2019) Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques. IEEE Netw. 33(1): 70-76.
43. Amazon Web Services Lambda@Edge. https://aws.amazon.com/lambda/edge/. Accessed: 28 November 2019
44. Mukherjee M, Matam R, Shu L (2107) Security and Privacy in Fog Computing: Challenges. IEEE Access 5:19293-19304.
45. Tuli S, Redowan Mahmud R, Tuli S, Buyya R (2019) FogBus: A Blockchain-based Lightweight Framework for Edge and Fog Computing. J. Syst. Software 154:22-36.
46. Jayasinghe U, Lee GM, MacDermott Á, Rhee WS (2019) TrustChain: A Privacy Preserving Blockchain with Edge Computing. Wirel Commun Mob Comput. doi:10.1155/2019/2014697
47. Wazid M, Das AK, Kumar N, Vasilakos AV (2019) Design of secure key management and user authentication scheme for fog computing services. Future Gener Comp Syst. 19: 475-492.
48. Huang B, Cheng X, Cao Y, Zhang L (2018) Lightweight Hardware-Based Secure Authentication Scheme for Fog Computing In: Proceedings of the IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, USA, pp 433-439.
49. GNU Privacy Guard. https://www.gnupg.org/. Accessed: 28 November 2019
50. Roman R, Lopez J, Mambo M (2018) Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. Future Gener. Comput. Syst. 78:680–698.