

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,800

Open access books available

142,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Network Slicing for Industrial IoT and Industrial Wireless Sensor Network: Deep Federated Learning Approach and Its Implementation Challenges

*Seifeddine Messaoud, Soulef Bouaafia, Abbas Bradai,  
Mohamed Ali Hajjaji, Abdellatif Mtibaa and Mohamed Atri*

## Abstract

5G networks are envisioned to support heterogeneous Industrial IoT (IIoT) and Industrial Wireless Sensor Network (IWSN) applications with a multitude Quality of Service (QoS) requirements. Network slicing is being recognized as a beacon technology that enables multi-service IIoT networks. Motivated by the growing computational capacity of the IIoT and the challenges of meeting QoS, federated reinforcement learning (RL) has become a propitious technique that gives out data collection and computation tasks to distributed network agents. This chapter discuss the new federated learning paradigm and then proposes a Deep Federated RL (DFRL) scheme to provide a federated network resource management for future IIoT networks. Toward this goal, the DFRL learns from Multi-Agent local models and provides them the ability to find optimal action decisions on LoRa parameters that satisfy QoS to IIoT virtual slice. Simulation results prove the effectiveness of the proposed framework compared to the early tools.

**Keywords:** federated learning, industrial IoT, network slicing, QoS

## 1. Introduction

In the last decade, industrial manufacturing such as healthcare, smart grids based on Cyber-Physical Internet of Thing Systems (CPIoTS) has been widespread [1]. In this context, IIoT network, which is characterized by the unified network physical layer, the QoS constraints, the autonomous connection requirements, is considered one of the key issues. The rapid increase in data amounts with diverse QoS requirements [2] brings several challenges in order to meet the complex requirements, as well as resource and QoS requirements with high data rate and low latency. In fact, the advanced 5G technology has a significant potential to provide IIoT QoS satisfactory

[3]. With their architectural approaches which are founded on a unified physical layer, addressing the diverging performance requirements in terms scalability and availability still a hot challenges topic. Today's drastic digital transformations empowered by emerging technology like Edge Computing, Software Defined Networking (SDN), Network Function Virtualization (NFV), and LoRaWAN can bring smart services for network candidates [4]. Network slicing (NS) is the key solution that provides smart service's connectivity with diverse QoS requirements. Using deep RL at each LoRa agent in the environment. Each agent considered to be a Deep Q-learning (DQL) brain interacts with the environment to find the best action on their parameters that brings the best reward. In addition, it introduces the FL approach to provide better RL based action on each agent, to maximize QoS, and hence throughput revenue. However, NS provides the network availability as a service following the slice instances exploiting NFV and SDN [5]. In this context, a Mini Batch GD and GMM framework is proposed in [7] to provides radio resources for the virtual slice member. In addition, a LoRa network slicing technique based on Maximum Likelihood Estimation proposed, in [6], to allocate network resources in inter and intra mode. Meanwhile, recently supervised learning approach-based resource allocation is also proposed to manage network resources, but due to the training data unavailability or the high computational training process, are not appropriate for large-scale network and cannot satisfy dynamic slices requirements.

RL technique can improve efficient resource management by interacting with the environment, in which Q-learning is the widely used. The RL agent learns the association between taken action and the received feedback in terms of reward. It follows a policy, which is updated according to the maximized revenue via several action series. Therefore, high-quality policies building in a centralized network architecture faces a major challenge, especially when the space of state features is restricted. To deal with these issues, Federated Learning (FL) has been suggested as a decentralized tool for machine learning, which is designed to be a global learning system. In this context, the aim of this work is to propose a deep federated reinforcement learning (DFRL), to equip the slice member with the required channel resources, by tuning LoRa TP and SF parameters [8].

The leftover of this chapter is organized in six sections. Section 2 presents the related work of this chapter. In Section 3, we give a brief overview on the Industrial IoT, the federated learning, and the network slicing. We highlight, in Section 5, the proposed slicing architecture and the system model. After that, in Section 6, the relation between wireless sensor network (WSN) and the IIoT is well highlighted. Next, the proposed slicing resource reservation-based DFRL framework is presented in Section 6. Section 7 evaluates the simulation results. Finally, Section 8 concludes the chapter.

## **2. Related works**

Recently, several articles have investigated the many challenges typical of the network slicing approach. In particular, in [9] the authors propose an online auction algorithm to realize a resource allocation framework, capable of guaranteeing the diversity of services to users and high levels of social welfare. Differently, the work in [10] deals with a new resource allocation framework to automatically and automatically size the capacity and size of network slices. In this chapter, resource partitioning is done based on both available network bandwidth and LoRa configurations parameters resulting in an optimal trade-off between traffic and network aspects. The authors of [11] focus on the design and implementation of a dynamic slicing sharing

system to ensure minimum user throughput requirements. Therefore, it addresses three sub-issues: admission control, resource allocation, and user abandonment issues. The contextualization of the placement of VNFs to the network slicing problem is presented in [12], where, in particular, the topological information of the network is exploited to provide an appropriate deployment of functions, with regard to different service classes. Moreover, the placement problem of VNFs has also been addressed in [13], which considers the function decomposition and the sub-functions sharing, a profitable heuristic algorithm is proposed based on Linear integer programming formulation of the VNF placement problem. Further, in the work [14], a formulation of mixed integer linear program is exploited to process the number identification of VNFs to use, which aims to meet specific service requirements. To solve the placing VNFs problem in a federated cloud, the coalition formation game is proposed in [14]. Alternatively, a Pareto analysis of the VNF placement problem is the subject of [15]. In [16], a network partitioning policy is developed to take into account the social well-being and the supplier profit of the network. Finally, special cases for the VNF placement problem are discussed and analyzed in [17].

The problem of bandwidth slicing in software-defined networks is studied in [18], where price spikes are exploited to indicate the presence of traffic spikes and network congestion. This work provides a time-based price analysis combined with a Stackelberg game, in which the gain of SP Internet is the gain of income. In a different way, the work cited in [19] studies the correlation between the network slices size and the resource pricing strategy. In addition, an algorithm to vary the prices is proposed by the authors in [19], with the aim of maximizing both the customer profit and the SP. Although FL has not been used in the field of network slicing research, FL has recently reached attention and several papers have presented its use, the methods cited in [20, 21] being prime examples of such a branch of literature. In [20], a new aggregation data scheme for wireless computation is proposed. The strategy exploits the signal overlay property of the wireless channels. Differently, articles [22, 20] focus on maximizing the number of devices involved in the aggregation process, also taking into account the minimization of aggregation error. Thus, it contextualize the FL in an MEC system and apply the distributed gradient descent method to identify the best compromise between local updates and global aggregations, aiming to minimize the loss function, in taking into account certain resource constraints. Likewise, the article in [21] analyzes the MEC environment and presents the application of hybrid filtering on stacked encoders to predict fluctuation in file popularity in the content caching problem. Moreover, the article cited [23] modifies the proposed, federated averaging algorithm with the stochastic gradient descent algorithm, to train the data in a distributed way, thus reducing communication costs. The multitasking learning problem is studied in the work cited in [24], authors proposed a new Mocha contextual optimization approach that used in combination with the FL system. The work cited in [25] analyzed the End-to-end delay in a blockchain framework, in which an FL blockchain structure is developed to perform a distributed consensus strategy. In order to improve the transmission and computational costs in a hybrid IoT-MEC network, authors in [26] proposed to use the FL powered by the multiple deep reinforcement learning agents. In addition, ultra-dense scenarios are also considered in [27], where an approach based on the technique of deep learning of short-term long-term memory is applied to forecast local network traffic in order to avoid congestion.

This chapter aims to address the problem of network slicing using deep federated RL at each LoRa agent in the environment. Each agent considered to be a Deep Q-learning (DQL) brain interacts with the environment to find the best action on their parameters

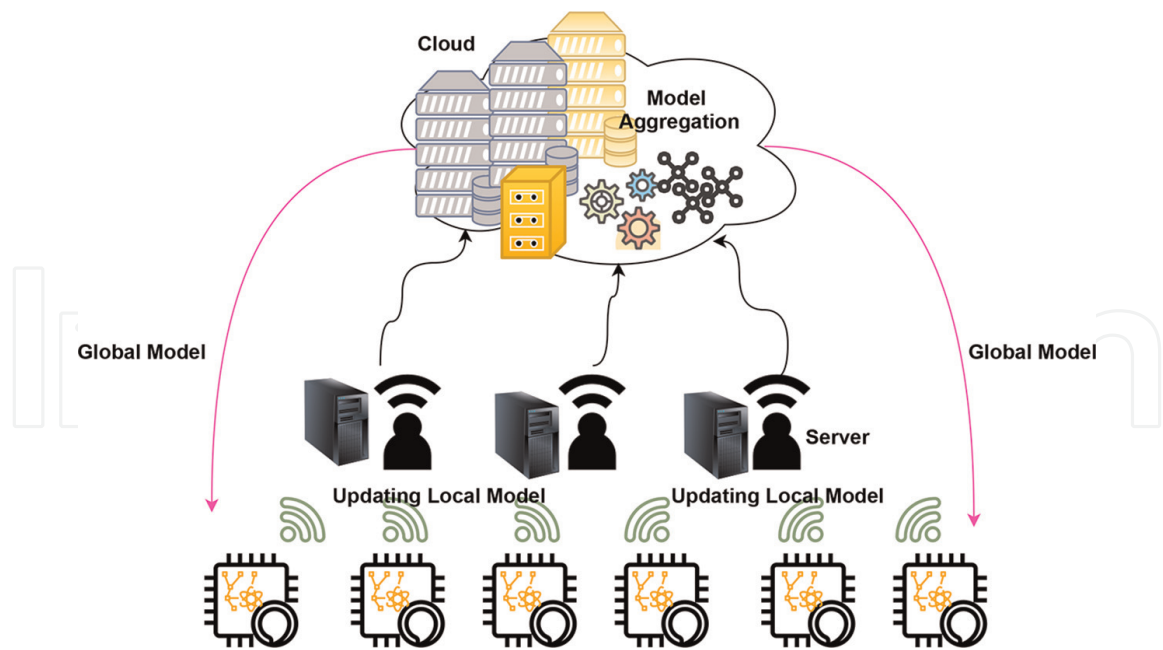


that brings the best reward. In addition, it introduces the FL approach to provide better RL-based action on each agent, to maximize QoS, and hence throughput revenue.

### **3. Overview on industrial IoT, federated learning and network slicing**

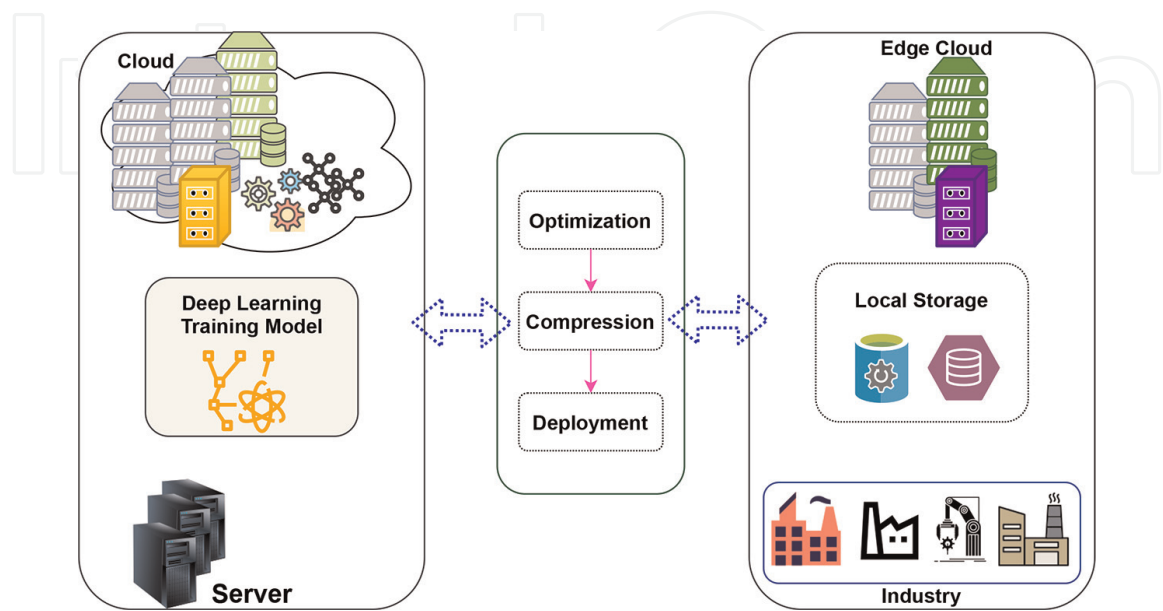
The development and evolution of modern information and communication technologies lead us to the Fourth Industrial Revolution, in which the Industrial Internet of Things (IIoT) is supposed to be one of the key aspects to realize the industry 4.0. With an unprecedented increase in the number of Internet of Things (IoT) devices and emerging applications, a large amount of traffic is created every day. Such an increase represents a heavy load on the Internet network and also requires significant investments for the upgrade of the infrastructure. However, with the development of big data analysis and artificial intelligence (AI) techniques such as deep learning (DL) and machine learning (ML), the data collected can be effectively exploited for many purposes. From a communication point of view, the last few years have seen the emergence of AI applications in various fields. For example, ML is used to study efficient antenna selection in multi-antenna wireless systems [28], DL is used to handle the computational offload problem in IoT systems with edge computing [29], and Deep reinforcement learning (DRL) is used to optimize resource allocation issues at the edge of the network, such as traffic classification, edge caching, network security, and data offload [30]. However, conventional AI models generally require central processing of the data collected from all users on the network, i.e. users have to upload their own data to a central server to train the learning model. However, a key concern with central learning is data privacy, i.e., some users want to keep track of their local data and do not want to transmit their local data to the central server. Training the learning model centrally requires a central cloud with extremely powerful compute and storage capabilities. Meanwhile, recent advancements in computer hardware and the proliferation of smart devices in our daily lives have shown that every IoT device can be equipped with reasonable levels of compute and storage, which is closely comparable to a desktop computer there was. is 10 years old [31]. Therefore, the standard ML model is not easily applicable to large scale IoT networks and cannot exploit the availability of distributed computing. This requires a new learning model that leaves training data distributed across individual IoT devices instead of being centralized.

Motivated by this problem, Google invented the concept of federated learning (FL) for on-device learning and data privacy preservation [32]. Using the FL approach, each IoT device can train its model based on locally collected data. Local data from IoT devices does not need to be sent to the centralized cloud. The centralized cloud only needs to collect the updated local training model from individual users. Due to its characteristics, FL has been adopted in many applications, for example FL for improving Google keyboard suggestions [33], FL for healthcare [34], and FL for smart city detection [35]. To illustrate the concept of FL, an overview of FL in IoT systems is shown in **Figure 1**. In general, each IoT device has its own set of data and the aggregation server can either be located at the edge of the network or in a virtual cloud in the remote cloud computing system [36]. Each FL model has its own advantages and disadvantages, depending on various factors. For example, FL with the server at the edge of the network is suitable for applications requiring low latency, location awareness and contextual information on the network while cloud-based FL is suitable for applications with IoT devices massive over multiple regions and computing power requirements/storage capacities.



**Figure 1.**  
*An overview of FL in IoT systems.*

Recently, the integration of DL models with IoT and edge devices has become more popular, which provides real-time analytics with limited resources [60]. Thus, Federated DL (FDL) allows Industry 4.0 companies to integrate DL into IoT devices and provides a secure framework using FL, as shown in **Figure 2**. DL is computationally expensive, which requires resources and an expensive framework. Thus, the decentralization of DL models is a multidimensional problem that requires a framework of new technologies to integrate DL with advanced computing and the IIoT. The main goal of FDL is to provide the IIoT with advanced capabilities using optimized DL that would turn Industry 4.0 factories into smart factories. Some of the parameters required to create the FDL model in IIoT are the FDL model, FDL networking, FDL security, and FDL optimization.



**Figure 2.**  
*Federated DL in IIoT.*

An FDL model can be implemented on both client side and server side. On the client side, private networks are defined, the DL model of which is tuned and optimized from the general model present in the cloud. Optimized and fitted models are then deployed on the client side, where the model is trained with data generated locally from the end device. Finally, the final device contains a highly quantized and compressed FDL model. On the server side, the model in the cloud is continuously updated by differentially integrating the gradients of each private network. Each local DL network in turn is responsible for continuously uploading and uploading the currently updated gradients to the cloud model. Thus, a distributed selective stochastic gradient descent approach is presented in [37] which can be applied in the cloud model to frequently update the local private model. The first decentralized model called "Model chain" uses blockchain technology [38, 39] to allow the preservation of confidentiality in the transfer of data. In addition to this, asynchronous stochastic gradient descent can also be used when a single model can be trained in parallel among all devices, aggregated and processed.

Regarding FDL communication and networking, is that the main benefit of using FDL is to run DL models in IoT devices and involve the model in the decision making process. This type of decentralized DL process improves the robustness, operational efficiency and reliability of IoT devices. FDL provides two types of communication, namely intra-communication channel and inter-communication channel. Train transmits data between all levels of the framework. FDL communicates between the IoT and the cloud tier where the cloud-optimized model is deployed on the end device. However, security and confidentiality must be maintained in the FDL during communication. In inter communication channels, the components of each layer communicate with each other in three different ways, such as cloud, edge, and end device. The main objective of FDL is to minimize intra-communication and to maximize inter-communication, which would greatly reduce the cost of communication. By the way, to maintain privacy and security, FDL builds DL models that do not expose information about the data to the cloud. Security issue on the server side includes sharing of DL models on the cloud that leads to confidentiality and security risk. Security issue on the client side is done by encrypting the data during the training process before sending it to the cloud server. Some mechanisms and homomorphic encryption technique controls the amount of data to be shared on the cloud. Since peripherals have limited memory and computational requirements, DL models must be optimized so that they can be deployed to IoT or peripherals efficiently. In terms of hardware optimization, the GPU provides low-power computation that reduces computation time. The FPGA and Google's Tensor Processing Unit [40] are other DL devices that enhance DL network processing. In terms of memory optimization, algorithms such as shared memory allocation algorithms for DL models can be used. Dynamic scheduling [41] is one of the main processes used to improve performance on a cloud server.

## **4. Wireless sensor networks and its relation to industrial internet of things**

### **4.1 Relation between IIoT and IWSN**

At the heart of the IIoT are the WSNs, that include of multi-functional nodes, low-cost, along with sensing, have both communication and processing capabilities. In order to communicate wirelessly over short distances, these little, inexpensive sensor nodes have built-in transceivers and processors. They are densely exploited in an area of

interest to collect sensory data, by coordinating and collaboratively exchanging information by training ad hoc wireless networks. Due to the small size and the batteries use, Sensor nodes are limited in processing, communication, and power. A unique feature of WSNs is their network processing attribute, whereby sensor nodes do not send raw sense data directly to the gateway but merge it locally to make it more consistent and save significant communication costs. Their application field is multiple and they are now ubiquitous components of intelligent environments, due to their unique attributes. Their various area of application covers home, surveillance, military, smart city, patient health monitoring, automation, etc. WSNs are used in telehealth applications in patient healthcare monitoring scenarios. As example, to monitor patients with chronic diseases and regularly check their various parameters such as heart rate, blood sugar and send this information wirelessly to a doctor remotely for further diagnosis. In order to help the elderly and disabled in their daily tasks, the WSNs are also used.

Indeed, they have seen major deployments in a diversity of applications, including agriculture, industrial process automation and control, transportation, and supply chain management over the past decades. Due to their ubiquitous presence and considering the potential benefits of these networks, such as simple deployment, cheap installation cost, no cabling cost, less complexity and mobility, they are increasingly used. in IIoT applications, which gave rise to IWSNs. WSNs can be used in an IIoT environment such as automation and control, process monitoring, and safety and emergency applications. In automation and process control applications, several tasks may require active nodes named actuators, which have the capability to act autonomously on the physical environment based on the detected measurements. For example, in the automation and control of feedback-based chemical processes, sensors measure temperature; if the temperature crosses a certain threshold value, they inform the actuators to reduce the temperature to a desired value so that the process remains in a stable state. Such applications place strict constraints on low latency and reliability because the sensor measurements must reach the actuator in a timely and reliable manner in order for the valve control action to be performed on time [42].

Today's sensor nodes have more processing power, longer battery life and memory, due to recent technological improvements compared to the first resource-constrained sensor nodes. This allowed them to be used in IIoT applications and resulted in IWSN. IWSN makes processes independent and autonomous, especially in difficult areas, to get actuation and control information, sensory. Sensor nodes in the WSN field detect process variables (e.g. temperature, pressure, etc.) and pass them to the well or gateway. The sink then passes it to the process controller whose job is to control the process variable under some required value. The receiver is responsible for the sensor network management and is controlled and managed by the host application management. The Network and Security Manager is responsible for entire network monitoring and ensuring security against attacks. Therefore, WSN has the potential to improve production processes and quality of products without compromising the IIoT QoS. Actuation and control, and sensing, are also imperative in majority industrial applications. In these applications, the sensors detect the data and the actuators act on the data based on certain control decisions made by the process controller.

## **4.2 Federated learning implementation challenges in IIoT and IWSN**

To implement FL's full potential in IIoT and IWSN, there are still several fundamental challenges that need to be addressed. In this section, we describe the challenges followed by very promising opportunities to meet those challenges.



#### *4.2.1 Limited computational resources*

Indeed, the FL deployment on IIoT and IWSN networks relies deeply on the computational resources and memory of edge devices. Consequently, people often focus only on the IoT devices capabilities to gather data while ignoring their limited memory and compute resources, which makes it hard for most IoT and sensor devices to finish local computation with massive data or sophisticated models. In order to address this challenge, lightweight AI techniques have been explored, which can be implemented in resource-constrained FL-IoT and WSN environments, such as improved resource management approaches to accelerate FL training on devices.

#### *4.2.2 Device heterogeneity*

In the multi-device settings, participants under the FL framework have various system resources, such as compute and memory resources. As the trend in machine learning is for larger and deeper models, the hardware heterogeneity within the IIoT and IWSN systems pose several challenges for the FL structure. They could easily train large models as devices with powerful memory and computing resources while other devices with limited resources could only train smaller models. Reader speed will also vary across devices, even for the same model size, which can trigger the problem of asynchronous communication discussed above. Due to the availability of the resources, an FL framework for IIoT and IWSN should provide a graceful adaptation of data and compute load on diverse devices.

#### *4.2.3 Limited networking bandwidth*

Communication overload is considered to be one of the main challenges in FL-based IIoT and IWSN environments. Currently, most IoT and WSN devices communicate using wireless networks that have a much lower bandwidth than the wired network bandwidth. As more and more devices join the system, the communication problem arises when the clients have different resource allocations. The limited network bandwidth not only makes the communication between clients and the server inefficient, but also triggers the presence of late clients, which fail to share their local update with the server during the communication cycle. To meet this challenge, some key ideas can be used, such as decentralized training, data compression and participant selection.

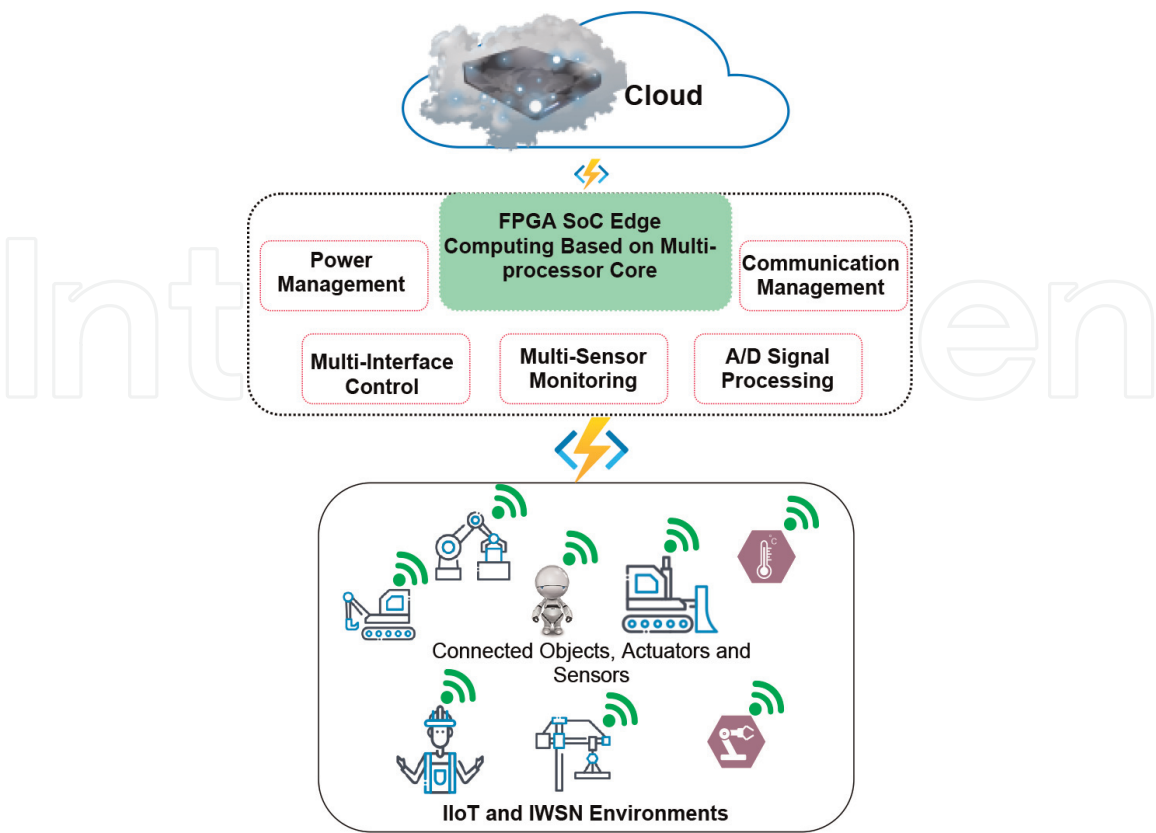
#### *4.2.4 Adversarial attack and defense*

The IoT devices prevalence also poses an attractive target in the real-world deployment for adversaries seeking to launch attacks, such as identity theft, phishing, and distributed denial of service (DDoS). Many IoT and ISN devices do not have the compute resources to do so, although these attacks can be easily defended by installing security patches. It is critical for the IIoT and IWSN systems to detect the malicious or broken IoT devices that will ruin the model training with limited resource. To address this challenge, one of the promising directions is to implement a lightweight security protocol in the IIoT and IWSN systems for the detection of broken and malicious devices.

4.2.5 Expected future solutions

Undeniably, the IIoT and IWSN ecosystems continue to evolve at a breakneck pace, exceeding all growth expectations and ubiquity barriers. From sensor to cloud, this giant network keeps breaking technological bounds in several domains, and wireless sensor nodes are expected to be predominant as the number of IoT devices grows toward the trillions to connect the unconnected world and things. However, their future in the IIoT and IWSN ecosystems still seems foggy, where several challenges, such as device’s connectivity, artificial intelligence (AI) at the edge, security and privacy concerns, growing energy needs, the right technologies to be used and keep pulling in opposite directions. To address these issues, which are caused by the complexity and variability of the environment, advanced computing related technologies are widely applied. However, the edge computing is limited by cost, volume, power consumption, and other conditions, so the capacity of edge computing cannot be fully exploited. So that edge computing fully exploits its characteristics of flexible management, federated and collaborative execution and heterogeneous environment, the reconfigurable real-time computer system based on FPGA SoC is strongly recommended. The system, as depicted in **Figure 3** can be built in real time as needed, by the characteristics of the FPGA SoC, including its reconfigurability, partial and total and precise clock control.

A multi-threaded huge number, computing requirements, and parallel heterogeneous data processing are persistently proposed in many environments of manufacturing. However, depending on the multi-environment’s requirements with different multiple tasks and several scenes, a single algorithm can no longer face the



**Figure 3.**  
Reconfigurable edge computing system based on FPGA SoC.

requirements so that numerous complicated tasks require the algorithm to be reconfigured and replaced. Without a doubt, the FPGA employment gratifies this multitude of requirements. However, it can rebuild the logic of the chip by means of configuration and reconfiguration of the resources inside the chip to form hardware with different functions by means of software. Therefore, in addition to the programmability and flexibility of the software, the FPGA also exhibits high throughput, low power, and low latency characteristics. In addition, due to its rich In-Output, FPGA SoCs are also very relevant for use On-chip protocols applications and interface conversion. The main benefits from employing FPGA for the edge computing are as follows:

- A constant throughput can be provided by the FPGA with a constant load size-based application, so that can integrate multiple service requests from several sensors in the IoT.
- Large-scale temporal and spatial parallelism is provided by the FPGA with fine granularity, so that ensures a high concurrency and high dependency algorithm with high acceleration performance.
- Compared with the processor, the FPGA has the lower power consumption and faster computing speed, which can provide the stability and lower task energy consumption.

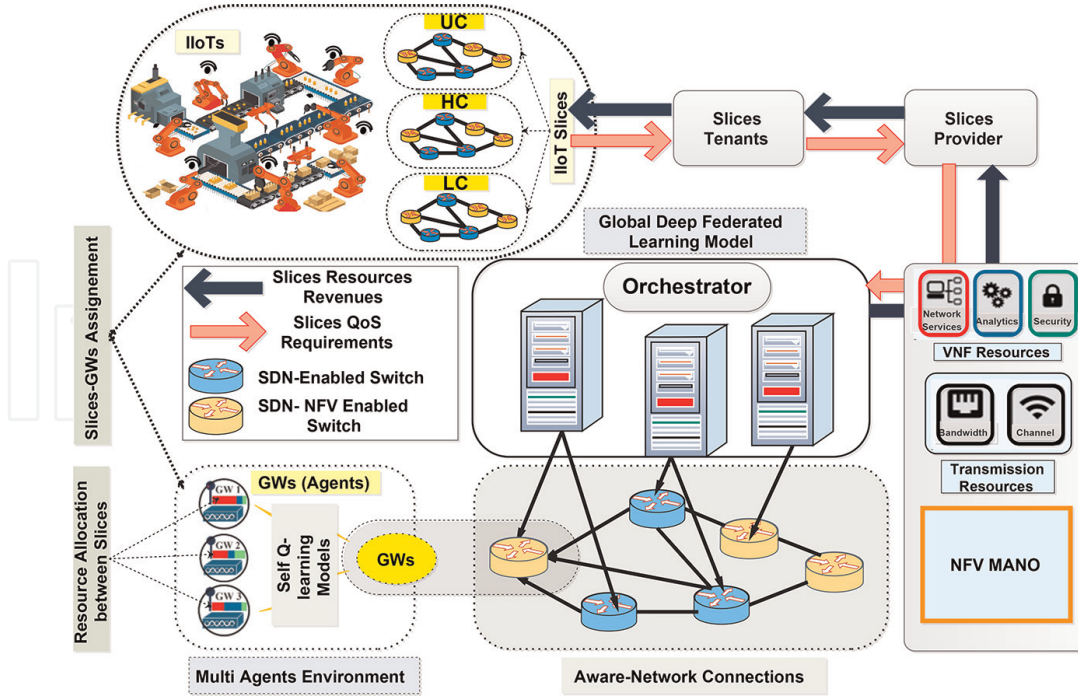
## 5. Network slicing architecture and system model

### 5.1 Network slicing architecture

The 5G network infrastructure design should focus on attentive consideration of software control, hardware infrastructure, and interconnection between them. In this context, we consider a network slicing architecture consisting of a set of IIoT slices  $J = \{1, \dots, j\}$ , where  $j$  represents the slice number. These slices are built on a unified physical infrastructure and share the same network resources. The proposed architecture, which is denoted in **Figure 4**, consists of three virtual slices. The urgent slice is the UCLE which yields more significance to the QoS and the efficiency. Thereafter, the HCLE slice that donates less importance to the latency. The last one is defined as the LCLE, which has the lowest slice priorities with unsecured QoS. The table denoted in the work [7] represents the slice's QoS requirements adopted for our architecture. This architecture consists of a set of  $K = \{1, \dots, k\}$  gateways, where  $k$  is the number of gateways. Then, gateways take over the task of providing radio resources to the substrate network layer, which contains a set of  $I = \{1, \dots, i\}$  IIoT devices, affected to the slice that meets its QoS demands.

### 5.2 Slicing system model

This slice set  $j \in J$  is integrated virtually on a Gateways (GWs) set  $k \in K$ . However, the physical resources of each GW consist of a set of  $C = \{1, \dots, c\}$  channels, in which each one includes a bandwidth  $b \in B = \{1, \dots, b\}$ . The goal of this work is to provide, for slices member, dynamic channel management based on TP and SF tuning. IN this



**Figure 4.**  
 Deep federated RL-based network slicing architecture.

context,  $\alpha_i \in \{0, 1\}$  is denoted as the binary value that indicate the admission success of device  $i \in I$  to slice  $j$  on GW  $k$ . Therefore, we define the Throughput  $\phi_i$  and Delay  $d_i$  models, based on SF and TP parameters for each device  $i_{j,k}$ , as in (1), (2), respectively [6].

$$\phi_i = SF \cdot \frac{R_c}{2^{SF}} \cdot CR = SF \cdot \frac{b_{j,k}}{2^{SF}} \cdot CR, \quad \forall i \in I_{j,k}, \quad (1)$$

$$d_i = \frac{L_i}{\phi_i}, \quad \forall i \in I_{j,k}, \quad (2)$$

where  $R_c$  is the chip rate,  $b_{i,j}$  denotes the bandwidth assigned for slice  $j$  on LoRa GW  $k$ ,  $CR$  represents the coding rate, and  $L_i$  is denoted as the packet size. Following the ultimate goal that seek to manage slice's QoS demands, energy efficiency (EE), given in (3), is considered as the second objective that should be maximized for IIoT devices assigned to each slice on each GW.

$$\max u_{EE}^{j,k} = \sum_{i \in I_{j,k}} \alpha_i \frac{\bar{\phi}_i}{P_j^T + P_c}, \quad \forall i \in I_{j,k}, \quad (3)$$

however,  $p_i^t$  denotes the allocation power for each IIoT device.  $u_{EE}^{j,k}$  is the EE metric that provides the efficiency of energy efficiency of each slice. While,  $P_c$  denotes the power consumption of the circuit and  $P_j^T = \sum_{i \in I_{j,k}} p_i^t$  is the TP. Finally, we define the multi-objective problem, as in (4), aiming to maximize the slice utility revenues  $U_{rm}^{j,k}$ .

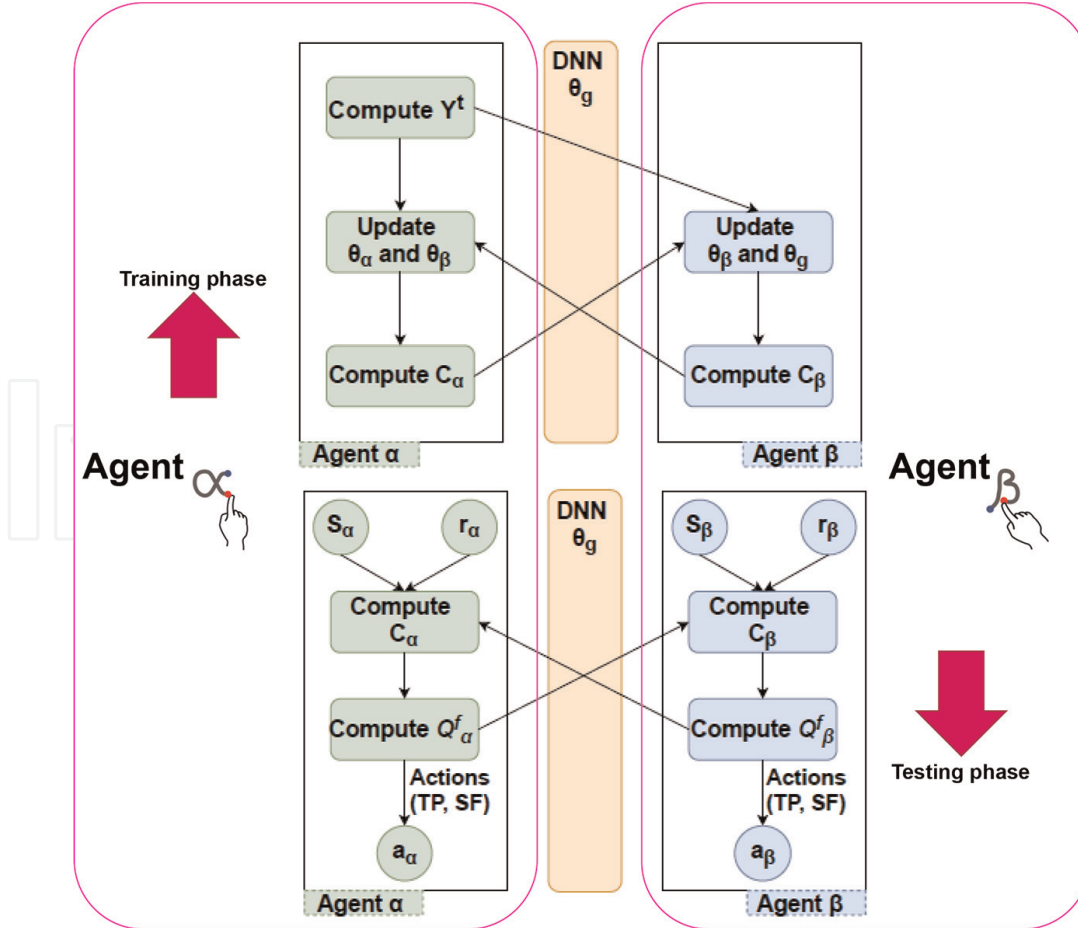
$$\max U_{rm}^{j,k} = \sum_{j,k} \left( u_{QoS}^{j,k} + u_{EE}^{j,k} + u_{REL}^{j,k} \right), \quad \forall k \in K, \forall j \in J, \quad (4)$$



## 6. The proposed DFRL for network slicing framework

We assume that agents, by sharing its self models based on the Q-network experiences, collaborate to receive global rewards from the federated orchestrator. While the orchestrator collects these models to builds a global network model that provides an optimal actions, on LoRa parameters, that maximize QoS revenue [43, 44].

The considered network consists of two agents, called as agent  $\alpha$  and agent  $\beta$ , that play in Markov environment. In this context, we denote the reply memory for agent  $\alpha$  by  $\mathcal{D}_\alpha = \{s_\alpha, a_\alpha, s'_\alpha, r_\alpha\}$  and the reply memory for agent  $\beta$  by  $\mathcal{D}_\beta = \{s_\beta, a_\beta, s'_\beta, r_\beta\}$ . These memories are used to store transitions parameters which will be collected, during interaction, to build an optimal policy ( $\pi_\alpha^*$  and  $\pi_\beta^*$ ). The notations of Q-functions, states, actions, and policy are denoted, respecting to agents  $\alpha$  and  $\beta$ , respectively, as  $\{Q_\alpha, s_\alpha \in \mathcal{S}, a_\alpha \in \mathcal{A}_\alpha, \pi_\alpha^*\}$  and  $\{Q_\beta, s_\beta \in \mathcal{S}, a_\beta \in \mathcal{A}_\beta, \pi_\beta^*\}$ . Thereby, assuming that states spaces ( $s_\alpha$  and  $s_\beta$ ), transitions parameters ( $\mathcal{D}_\alpha$  and  $\mathcal{D}_\beta$ ), and the Q-network functions ( $Q_\alpha$  and  $Q_\beta$ ) are different for the defined agents  $\alpha$  and  $\beta$ . Each agent builds its own Q-network ( $Q_\alpha$  or  $Q_\beta$ ), and  $\theta$  ( $\theta_\alpha$  or  $\theta_\beta$ ) parameters. These agents interacts with the DFL model with the aim is to build a global federated model that satisfy dynamic slice's QoS demands exploiting local agents experiences  $Q_\alpha$  and  $Q_\beta$ .



**Figure 5.**  
Training and testing phases.

Therefore, based on the Q-networks models, (5) represents the DFL (based on DNN) Q-network output as  $Q_f(\theta_\alpha, \theta_\beta; \theta_g)$ .

$$Q_f(\theta_\alpha, \theta_\beta; \theta_g) = DNN([Q_\alpha(s_\alpha, a_\alpha; \theta_\alpha) | Q_\beta(s_\beta, a_\beta; \theta_\beta)]; \theta_g), \quad (5)$$

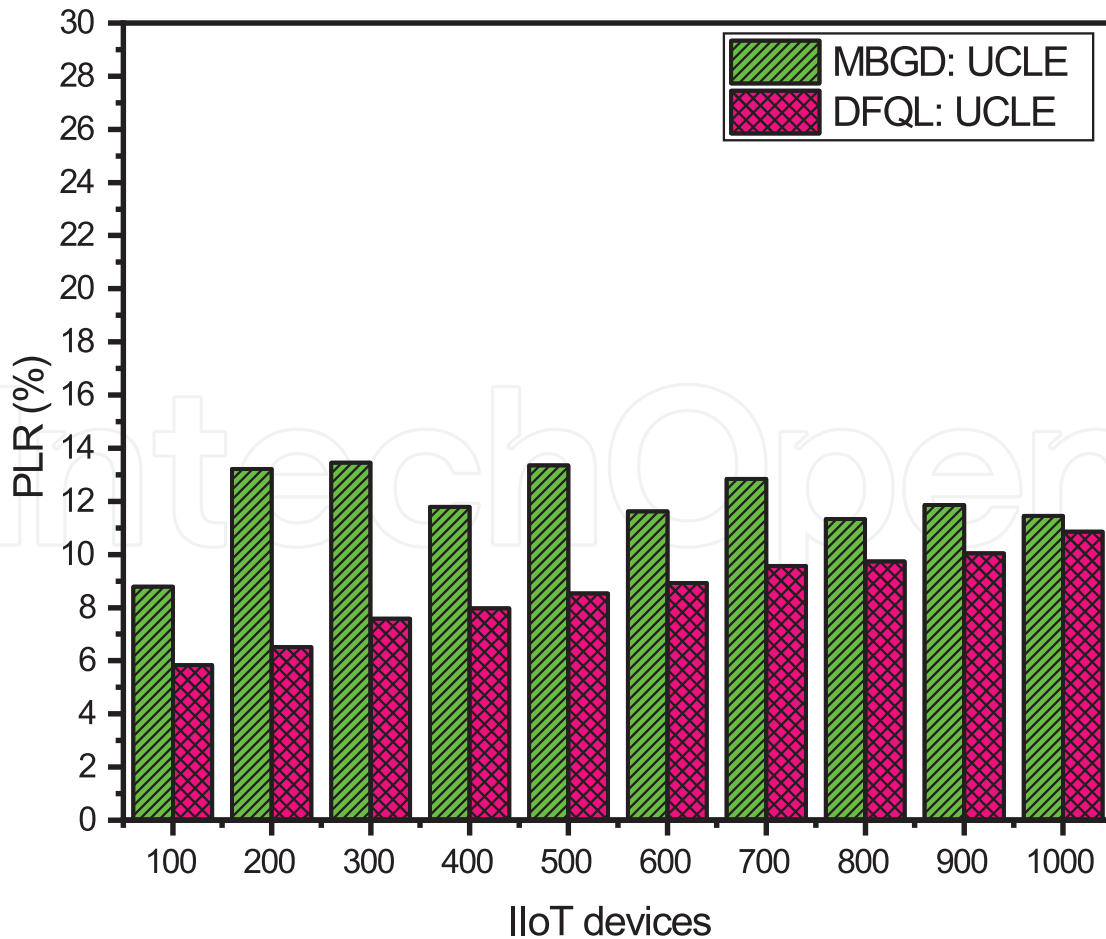
where  $[.]$  denotes the concatenation symbol and  $\theta_g$  denotes the DNN (DFL) parameter shared between agents.

At this stage, the Mean Square Error (MSE), is defined for agents  $\alpha$  and  $\beta$ , as a Loss function denoted in formulas (6), (7) [7]. These formulas are used to train the proposed framework, by updating the parameters  $(\theta_\alpha, \theta_\beta, \theta_g)$ , to build federated model that will be able to find, then, an optimal action decision on TP and SF that maximize slice's QoS rewards.

$$MSE_\alpha^t(\theta_\alpha, \theta_g) = \mathbb{E} \left[ \left( Y^t - Q_f^\alpha(s_\alpha^t, a_\alpha^t, C_\beta; \theta_\alpha, \theta_g) \right)^2 \right] \quad (6)$$

$$MSE_\beta^t(\theta_\beta, \theta_g) = \mathbb{E} \left[ \left( Y^t - Q_f^\beta(s_\beta^t, a_\beta^t, C_\alpha; \theta_\beta, \theta_g) \right)^2 \right], \quad (7)$$

while  $Y^t = r^t(s) + \gamma \max_{a \in \mathcal{A}} Q_f^\alpha(s_\alpha^t, a_\alpha^t, C_\beta; \theta_\alpha, \theta_g)$  is attributed for agent  $\alpha$  only, as a condition to start training. **Figure 5** depicts the DFRL framework process.



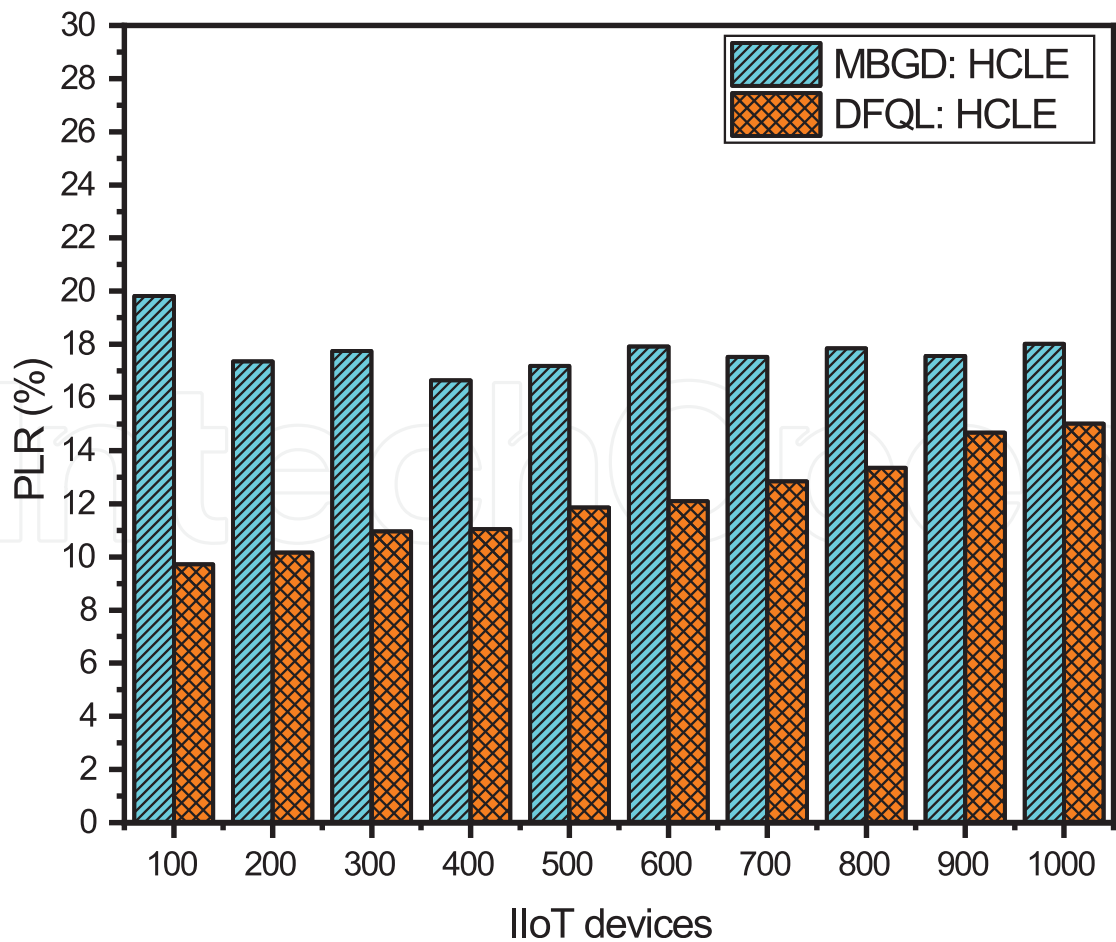
**Figure 6.**  
 PLR of UCLE.

### 7. Experiment results

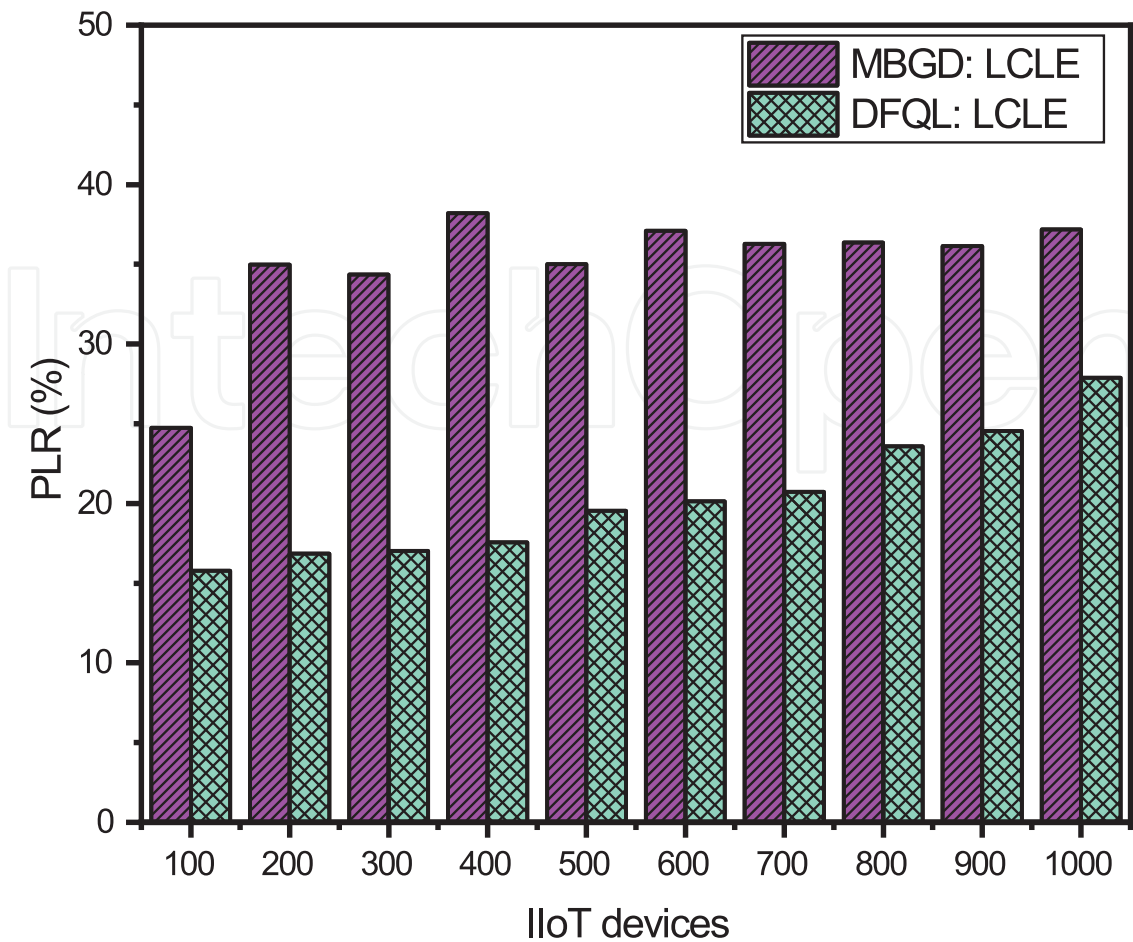
The proposed framework has been implemented in Python language using TensorFlow-gpu package on Intel Xeon E5-2620 v4 2x 8-Core with 64 GB RAM. Also, the NVIDIA GK110BGL [Tesla K40c] is used to improve speed during the training phase.

We provide, in this section, the mean percentage of Packet Loss Rate (PLR) for IIoT devices, as denoted in **Figures 6–8**, and compare it to the PLR within MBGD scheme [7].

However, by increasing the devices, PLR will increase subsequently. This return to the data rate, that when increase, the number of successful transmitted packet increase accordingly. While it is not the case when throughput is low. We remark also, in **Figures 6–8**, that UCLE and HCLE slices have a reduced PLR compared to LCLE. However, this due to the reliability and efficiency constraints dedicated for this slice which is not the case in LCLE slice that consider only the load. Compared with the slice results using the MBGD technique, we could obviously note the efficiency of the proposed federated scheme in supporting dynamic slicing strategy by reducing PLR over than 9%. This improvement return to the shared experience between agents that can improve the action decision on TP and SF to slices.



**Figure 7.**  
*PLR of HCLE.*



**Figure 8.**  
*PLR of LCLE.*

## 8. Conclusion

This prospective chapter presents a future outlook on low-end nodes in the IIoT and IWSN eras. Following a detailed discussion of the trends and challenges posed by the IIoT and IWSN paradigm to low-end devices, it discusses how modern reconfigurable platforms are the perfect candidate to meet the ever-evolving industrial environments. Indeed, in this chapter, we proposed a federated network slicing based on deep reinforcement learning techniques for channels and bandwidth management based LoRa promising technology that meet IIoT and IWSN network service requirements based on the SDN, NFV, network slicing, and deep reinforcement learning techniques. Each LoRa GW plays an agent role, in the environment, and profits from the learning experience provided by the other coexist agents via the global federated model.

In the case of future studies, this chapter introduced comprehensive review and several research lines, especially one attractive future line is related to the integration of FPGA SoC at the edge to build a smart factory as well as IIoT and IWSN environments with environmentally friendly capabilities and functionalities. In addition, future research is needed to fully embrace cloud services and new ways of connectivity in order to get the full benefits of the new Edge FPGA SoC technology.



## Nomenclature

$J = \{1, \dots, j\}$	IIoT network slices set
$K = \{1, \dots, k\}$	LoRa gateways (agents) set
$I = \{1, \dots, i\}$	IIoT devices set associated to each slice
$B = \{1, \dots, b\}$	channel bandwidth set
$C = \{1, \dots, c\}$	LoRa-GW's channels set
$\alpha_i \in \{0, 1\}, \forall i \in I_{j,k}$	device's admission and association index to slice
$\forall i \in I_{j,k}$	device $i$ assigned to slice $j$ on gateway $k$
TP	transmission power
SF	spreading factor
$\phi_i, \forall i \in I_{j,k}$	throughput of device $i$
$d_i, \forall i \in I_{j,k}$	delay of device $i$
$u_{QoS}^{j,k}$	quality of service metric for slice $j$ on GW $k$
$p_i^t, \forall i \in I_{j,k}$	the power allocated for each device $i$
$u_{EE}^{j,k}$	energy efficiency metric for slice $j$ on GW $k$
$p_i^r$	the received power
$u_{REL}^{j,k}$	reliability metric for slice $j$ on GW $k$
$U_{rm}^{j,k}, \forall k \in K, \forall j \in J$	the global slice utility revenues metric
$\{S, A, T, R\}$	state, action, transition function, reward
$\alpha$ and $\beta$	agent $\alpha$ and agent $\beta$
$\gamma$	discount factor
$\theta_\alpha, \theta_\beta$	DQL network parameters (weights)
$\theta_g$	DNN network parameters (weights)
$\mathcal{D}_\alpha, \mathcal{D}_\beta$	reply memories to store transitions

## Abbreviations

IoT	internet of things
IIoT	industrial IoT
IWSN	industrial wireless sensor network
AI	artificial intelligence
ML	machine learning
DL	deep learning
QoS	quality of service
RL	reinforcement learning
DRL	deep reinforcement learning
FL	federated learning
DFL	deep federated learning
DFRL	deep federated reinforcement learning
CPIToS	cyber-physical internet of thing systems
5G	fifth generation network
SDN	software defined network
NFV	network function virtualization
NS	network slicing
DQL	deep Q-learning
GD	gradient descent

GMM	gaussian mixture model
SP	service provide
MEC	mobile edge computing
GPU	graphic processor unit
FPGA	field programmable gate array
UCLE	ultra critical of latency and efficiency
HCLE	high critical of latency and efficiency
LCLE	low critical of latency and efficiency
GW	gateway

### Author details

Seifeddine Messaoud<sup>1\*</sup>, Soulef Bouaafia<sup>1</sup>, Abbas Bradai<sup>2</sup>, Mohamed Ali Hajjaji<sup>3</sup>, Abdellatif Mtibaa<sup>4</sup> and Mohamed Atri<sup>5</sup>

1 Faculty of Sciences, Electronics and Microelectronics Lab., University of Monastir, Monastir, Tunisia

2 XLIM Lab., University of Poitiers, Poitiers, France


3 Electronics and Microelectronics Lab., ISAAT Sousse, University of Sousse, Sousse, Tunisia

4 Electronics and Microelectronics Lab., ENIM Monastir, University of Monastir, Monastir, Tunisia

5 College of Computer Science, King Khalid University, Abha, Saudi Arabia

\*Address all correspondence to: [seifeddine.messaoud@fsm.rnu.tn](mailto:seifeddine.messaoud@fsm.rnu.tn)

### IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Givehchi O, Landsdorf K, Simoens P, Colombo AW. Interoperability for industrial cyber-physical systems: An approach for legacy systems. *IEEE Transactions on Industrial Informatics*. 2017;**13**(6):3370-3378
- [2] Nordrum, A.. Popular Internet of Things. *IEEE Spectrum's Technology Blog* [Online]. 2016. Available from: <http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>
- [3] Messaoud S, Bradai A, Bukhari SHR, Qung PTA, Ahmed OB, Atri M. A survey on machine learning in internet of things: Algorithms, strategies, and applications. *Internet of Things*. 2020;**12**:100314
- [4] Khan LU, Yaqoob I, Tran NH, Kazmi SM, Dang TN, Hong CS. Edge computing enabled smart cities: A comprehensive survey. *arXiv preprint arXiv:1909.08747*. 2019
- [5] Kazmi SMA, Khan LU, Tran NH, Hong CS. *Network Slicing for 5G and Beyond Networks*. Berlin/Heidelberg, Germany: Springer. 2019;**1**
- [6] Dawaliby S, Bradai A, Pousset Y. Adaptive dynamic network slicing in LoRa networks. *Future Generation Computer Systems*. 2019;**98**:697-707
- [7] Messaoud S, Bradai A, Moulay E. Online GMM clustering and mini-batch gradient descent based optimization for industrial IoT 4.0. *IEEE Transactions on Industrial Informatics*. 2020;**16**(2):1427-1435
- [8] Messaoud S, Bradai A, Atri M. Distributed Q-learning based-decentralized resource allocation for future wireless networks. In: 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD). IEEE; 2020. pp. 892-896
- [9] Liang L, Wu Y, Feng G, Jian X, Jia Y. Online auction-based resource allocation for service-oriented network slicing. *IEEE Transactions on Vehicular Technology*. 2019;**68**(8):8063-8074
- [10] Leconte M, Paschos GS, Mertikopoulos P, Kozat UC. A resource allocation framework for network slicing. In: *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. Honolulu, HI, USA: IEEE; 2018. pp. 2177-2185
- [11] Caballero P, Banchs A, De Veciana G, Costa-Pérez X, Azcorra A. Network slicing for guaranteed rate services: Admission control and resource allocation games. *IEEE Transactions on Wireless Communications*. 2018;**17**(10):6419-6432
- [12] Guan W, Wen X, Wang L, Lu Z, Shen Y. A service-oriented deployment policy of end-to-end network slicing based on complex network theory. *IEEE Access*. 2018;**6**:19691-19701
- [13] Li D, Hong P, Wang W, Pei J. Virtual network function placement with function decomposition for virtual network slice. In: *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. Paris, France: IEEE; 2018. pp. 1-4
- [14] Bagaa M, Taleb T, Laghrissi A, Ksentini A, Flinck H. Coalitional game for the creation of efficient virtual core network slices in 5G mobile systems. *IEEE Journal on Selected Areas in Communications*. 2018;**36**(3):469-484

- [15] Wang G, Feng G, Qin S, Wen R, Sun S. Optimizing network slice dimensioning via resource pricing. *IEEE Access*. 2019;7:30331-30343
- [16] Schneider S, Dräxler S, Karl H. Trade-offs in dynamic resource allocation in network function virtualization. In: 2018 IEEE Globecom Workshops (GC Wkshps). Abu Dhabi, United Arab Emirates: IEEE; 2018. pp. 1-3
- [17] Liu J, Shi Y, Zhao L, Cao Y, Sun W, Kato N. Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network. *IEEE Journal on Selected Areas in Communications*. 2018;36(2):221-232
- [18] Gu B, Feng J, Zhou Z, Guizani M. Time-dependent pricing for on-demand bandwidth slicing in software defined networks. In: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). Limassol, Cyprus: IEEE; 2018. pp. 1024-1029
- [19] Wang G, Feng G, Tan W, Qin S, Wen R, Sun S. Resource allocation for network slices in 5G with network resource pricing. In: GLOBECOM 2017-2017 IEEE Global Communications Conference. Singapore: IEEE; 2017. pp. 1-6
- [20] Yang K, Jiang T, Shi Y, Ding Z. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*. 2020;19(3):2022-2035
- [21] Yu Z, Hu J, Min G, Lu H, Zhao Z, Wang H, et al. Federated learning based proactive content caching in edge computing. In: 2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi, United Arab Emirates: IEEE; 2018. pp. 1-6
- [22] Wang S, Tuor T, Salonidis T, Leung KK, Makaya C, He T, et al. When edge meets learning: Adaptive control for resource-constrained distributed machine learning. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Honolulu, HI, USA: IEEE; 2018. pp. 63-71
- [23] McMahan HB, Moore E, Ramage D, Arcas BA. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*. 2016
- [24] Smith V, Chiang CK, Sanjabi M, Talwalkar A. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*. 2017
- [25] Kim H, Park J, Bennis M, Kim SL. Blockchain on-device federated learning. *IEEE Communications Letters*. 2019;24(6):1279-1283
- [26] Ren J, Wang H, Hou T, Zheng S, Tang C. Federated learning-based computation offloading optimization in edge computing-supported internet of things. *IEEE Access*. 2019;7:69194-69201
- [27] Zhou Y, Fadlullah ZM, Mao B, Kato N. A deep-learning-based radio resource assignment technique for 5G ultra dense networks. *IEEE Network*. 2018;32(6):28-34
- [28] Joung J. Machine learning-based antenna selection in wireless communications. *IEEE Communications Letters*. 2016;20(11):2241-2244
- [29] Li H, Ota K, Dong M. Learning IoT in edge: Deep learning for the internet of things with edge computing. *IEEE Network*. 2018;32(1):96-101
- [30] Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang YC, et al. Applications of deep reinforcement learning in communications and



networking: A survey. *IEEE Communication Surveys and Tutorials*. 2019;21(4):3133-3174

[31] Pham QV, Fang F, Ha VN, Piran MJ, Le M, Le LB, et al. A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art. *IEEE Access*. 2020;8:116974-117017

[32] Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*. 2016

[33] Yang T, Andrew G, Eichner H, Sun H, Li W, Kong N, et al. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*. 2018

[34] Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*. 2021; 5(1):1-19

[35] Jiang JC, Kantarci B, Oktug S, Soyata T. Federated learning in smart city sensing: Challenges and opportunities. *Sensors*. 2020;20(21): 6230

[36] Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*. 2020; 10(1):1-12

[37] Shokri R, Shmatikov V. Privacy-preserving deep learning. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. Denver, CO, USA; 2015. pp. 1310-1321

[38] Deepa N, Pham QV, Nguyen DC, Bhattacharya S, Prabadevi B, Gadekallu TR, et al. A survey on blockchain for big data: Approaches, opportunities, and future directions. *arXiv preprint arXiv:2009.00858*. 2020

[39] Hakak S, Khan WZ, Gilkar GA, Assiri B, Alazab M, Bhattacharya S, Reddy GT. Recent advances in blockchain technology: A survey on applications and challenges. *arXiv preprint arXiv:2009.05718*. 2020

[40] Wang YE, Wei GY, Brooks D. Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701*. 2019

[41] Cho HD, Engineer PDP, Chung K, Kim T. Benefits of the Big. *LITTLE Architecture*. *EETimes*; 2012

[42] Liu L, Han G, Xu Z, Shu L, Martinez-Garcia M, Peng B. Predictive boundary tracking based on motion behavior learning for continuous objects in industrial wireless sensor networks. *IEEE Transactions on Mobile Computing*. 2021. Early Access

[43] Haque ME, Baroudi U. Ambient self-powered cluster-based wireless sensor networks for industry 4.0 applications. *Soft Computing*. 2021;25(3):1859-1884

[44] Messaoud S, Bradai A, Ahmed OB, Quang PTA, Atri M, Hossain MS. Deep federated q-learning-based network slicing for industrial IoT. *IEEE Transactions on Industrial Informatics*. 2020;17(8):5572-5582