Edge Cloud as an Enabler for Distributed AI in Industrial IoT Applications: the Experience of the IoTwins Project*

Paolo Bellavista[1][0000-0003-0992-79488] and Alessio Mora[1][0000-0001-8161-1070]

[1] Alma Mater Studiorum – University of Bologna, Bologna 40136, Italy
{paolo.bellavista, alessio.mora}@unibo.it

Abstract. Emerging Industrial Internet of Things (IIoT) applications are pushing the academic and industrial research towards novel solutions for, on the one hand, frameworks to facilitate the rapid and cost-effective exploitation of general-purpose machine learning mechanisms and tools, and, on the other hand, hw/sw infrastructures capable of guaranteeing the desired and challenging quality of service indicators in industrial scenarios, e.g., latency and reliability. We claim that these directions can be effectively and efficiently addressed through the adoption of innovative quality-aware edge cloud computing platforms for the design, implementation, and runtime support of distributed AI solutions that execute on both global cloud resources and edge nodes in industrial plant premises. In particular, the paper presents the first experiences that we are doing within the framework of the H2020 Innovation Action IoTwins, for the implementation and optimization of distributed hybrid twins in the IIoT application domains of predictive maintenance and manufacturing optimization. IoTwins exploits distributed hybrid twins, partly executing at edge cloud nodes in industrial plant localities, to perform process/fault predictions and manufacturing line reconfigurations under time constraints, also by enabling some forms of sovereignty on industrial monitoring data. In addition, the paper overviews our original taxonomy of the state-of-the-art research literature about distributed AI for decentralized learning, with specific focus on federated settings and on emerging trends for the IIoT domain.

Keywords: Industrial Internet of Things, Edge Cloud Computing, Distributed Digital Twins, Decentralized Learning, IoTwins.

1 Introduction

One of the major challenges of the Industrial Internet of Things (IIoT) is to take advantage of the IoT technology in industrial decisions. IoT today generates a myriad of data by the billions of connected devices, including sensors and actuators, that are usually aggregated and stored on cloud platforms [1, 2]. Mainly for manufacturing industries, the interaction and the management of IoT devices become enablers for new

service opportunities including predictive maintenance, continuous monitoring of the parts that are more subjected to degradation and ageing, scheduling and remote running of maintenance interventions, and simulation of operations to predict product quality and to evaluate process optimizations through digital twin implementation. However, it is recognized that the IIoT exhibits several characteristics that are peculiar and significantly different from the general-purpose IoT that we deal with in smart cities or smart buildings, e.g., in terms of communication bandwidth needed for machine-to-machine big data transmission in real-time. Other IIoT-specific and quality-related challenges refer to latency improvements and robust connectivity, together with limited costs on large scale deployment scenarios; the overall goal is that real-time decisions are enabled, under the quality constraints of the specific application domain and deployment environment, and result to efficiency, safety, and stability of large scale IIoT.

In more traditional IIoT solutions, an integrated infrastructure is deployed to collect information from heterogeneous sensors, to transmit it to the cloud, and to update the related parameters in the form of a closed-loop system [3]. The growth of edge/fog computing (we will use the term edge computing below to indicate generically both kinds of distributed cloud) is enabling the potential to move computing functionality from centralized and globally available datacenters to the edge of the network [4]. Generally speaking, edge computing (either statically or dynamically) migrates core capabilities such as networking, computing, storage, and applications closer to devices and in particular to IIoT endpoints. There are already interesting examples in the related literature about intelligent services close to manufacturing units, e.g., to meet key requirements such as agile connection, data analytics via edge nodes, highly responsive cloud services, and personalized enforcement of privacy policy strategies [5].

We claim that these directions can be effectively and efficiently addressed via the adoption of innovative quality-aware edge cloud computing platforms for the design, implementation, and runtime support of distributed AI solutions that execute on both global cloud resources and edge nodes in industrial plant premises. In particular, the paper presents the first research activities and the first development experiences that we are doing, within the framework of the H2020 Innovation Action IoTwins [6]. Within this large project, better and more extensively described in Section 2, we have the ambition to design, implement, evaluate, and optimize distributed hybrid twins with specific features that are suitable for the IIoT application domains of predictive maintenance and manufacturing optimization. In particular, our distributed hybrid twins are designed to partly execute at edge cloud nodes in industrial plant localities, to perform process/fault predictions and manufacturing line reconfigurations under strict time constraints and under the respect of reliability guarantees, also by enabling some forms of sovereignty on industrial monitoring data.

In addition to presenting the general guidelines of solution and the primary technical challenges that we are investigating within the framework of the IoTwins project, this paper aims at providing a significant contribution to the community of researchers in the field by offering an original taxonomy of the state-of-the-art research literature about distributed AI for decentralized learning, with the specific focus on federated settings and on emerging trends for the IIoT domain. In fact, this application domain is strongly stimulating research on the possibility to exploit machine learning techniques

to feed hybrid twin models and whose learning/refinement processes are distributed and uncoordinated as much as possible, in order to improve scalability and locality-aware specific optimizations. This is pushing for innovative models and original efficient platforms for decentralized learning where *i)* initial learning can be done centrally at the global cloud, once and in a uniform way for all involved industrial plants and facilities; *ii)* learned models are moved at the target edge nodes for running efficiently at the desired locality (e.g., for quality control purposes); *iii)* learned models can be refined at edge nodes in a differentiated way depending on the functioning history at each locality; and iv) local refinements of learned models may feed the next generation of cloud-based uniform learned models through proper collection and harmonization of the contribution from the distributed and federated network of edge nodes.

The remainder of this paper is organized as follows. Section 2 rapidly overviews the primary objectives and solution directions adopted in the IoTwins project, while Section 3 sketches the main features of our distributed hybrid twins. An original taxonomy of the first decentralized learning solutions based on edge computing that have been recently appeared in the related literature is presented in Section 4. Primary directions of most open technical challenges and most promising research activities in the field of decentralized learning, together with some brief concluding remarks, end the paper.

2 The IoTwins Project

The original results presented in the following parts of this paper have been achieved within the context of the just started H2020 IoTwins Innovation Action project, scientifically coordinated by our research group. IoTwins is a large (3 years, 20.1M€ budget) industry-driven project that puts together 23 partners from 8 countries; it has the ambition to lower the barriers, in particular for SMEs, to building edge-enabled and cloud-assisted intelligent systems and services based on big data for the domains of manufacturing and facility management. To this purpose, IoTwins is working to design a reference architecture for distributed and edge-enabled twins and is experimenting its implementation, deployment, integration, and in-the-field evaluation in several industrial testbeds.

IoTwins claims that IoT, edge computing, and industrial cloud technologies together are the cornerstones for the creation of distributed twin infrastructures that, after testbed experimentation, refinement, and maturity improvements, can be easily adopted by SMEs: *i)* industrial cloud, also based on HPC resources, enables the creation of accurate predictive models based on advanced ML for end-to-end deep networks, which require huge computing power for training; *ii)* elastic cloud resource availability creates the opportunity to boost model accuracy by fitting and complementing data produced by industrial IoT sensors with data produced by large-scale parallel simulation; *iii)* edge computing makes it possible to close the loop between accurate models and optimal decisions by enabling very responsive on-line local management of operational parameters in the targeted plants and filtered/fused reporting to the cloud side of only significant monitoring data (e.g., anomalies and deviations); and iv) edge computing can leverage and accelerate the adoption of digital twin techniques by exploiting its industry-

perceived advantages in terms of increased reliability/autonomy (e.g., independently of continuous connectivity to the global remote cloud) and of improved locality preservation of critical production data that can be maintained and used directly at the plant premises (data sovereignty).

In particular, the IoTwins distributed hybrid twins are essentially models that accurately represent a system (either infrastructure or process or machine) along with its performance, see Figure 1. These models enable the description of the system itself and its dynamics (descriptive or interpretative models), the prediction of its evolution (predictive models), and the optimization of its operation, management and maintenance (prescriptive models). They may be hybrid, i.e., by exploiting mixed and heterogeneous types of input from in-the-field experimental measurements (online/offline monitoring) and from analytical models as well as simulations/emulations. Of course, this is not the first case of digital twins in the literature: more traditionally, digital twins are meant as virtual representations of real-world objects (typically mobile and/or temporarily disconnected devices), e.g., in smart city scenarios [7] or in commercial applications to make IoT products remotely monitorable and controllable [8].

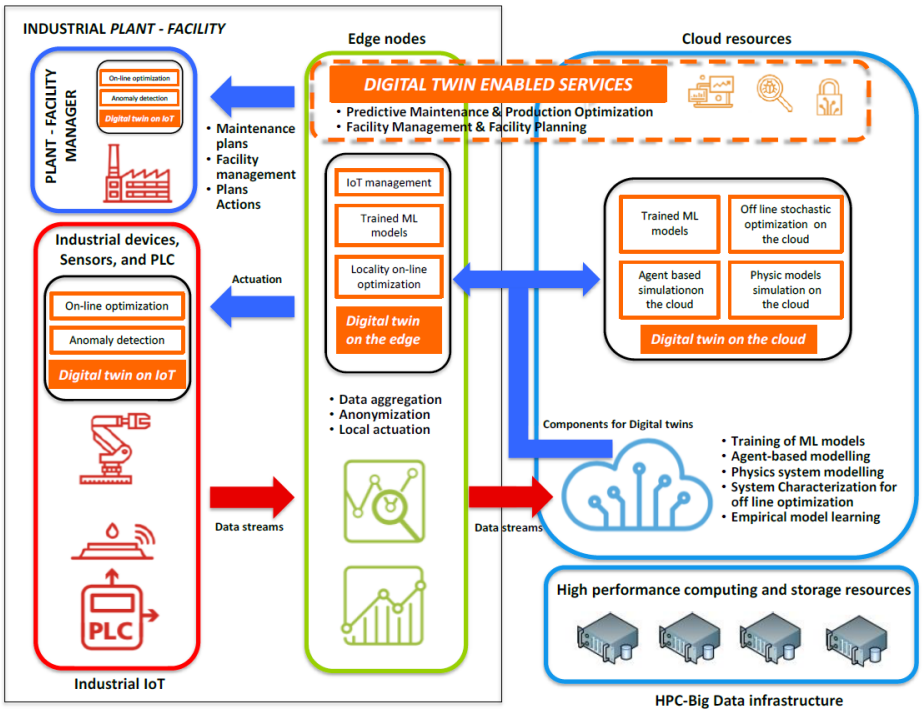

Figure 1. Conceptual vision of IoTwins distributed hybrid twins.

IoTwins distributed twins are used to detect and diagnose anomalies, to determine an optimal set of actions that maximize key performance metrics, to effectively and efficiently enforce on-line quality management of production processes under latency and reliability constraints, and to provide predictions for strategic planning to help

companies, especially SMEs, to significantly improve their profitability through digitalization, as well as to open up new opportunities for them for the creation of new services and business models. The IoTwins hybrid twins, among the others, enable: *i)* the description of systems; *ii)* the prediction of systems evolution; *iii)* the management and maintenance of systems. They are going to be used to detect and diagnose anomalies, to determine an optimal set of actions that maximize key performance metrics, to effectively and efficiently enforce on-line quality management of production processes under latency and reliability constraints, and to provide predictions for strategic planning to help companies to significantly improve their profitability through digitalization, as well as to open up new opportunities for them for the creation of new services and business models.

A crucial focus and primary activity of the project will be to deliver twelve industrial testbeds, of significant interest for SMEs, by sharing the same underlying methodology. The IoTwins testbeds are grouped into three classes: *i)* testbeds in the manufacturing sector with the goal of optimizing production quality and plant maintenance, *ii)* testbeds for the optimization of facility/infrastructure management, and *iii)* testbeds for the in-the-field verification of the replicability, scalability, and standardization of the proposed approach, as well as the generation of new business models. In particular, in the manufacturing sector, four industrial pilots are aimed at providing predictive maintenance services that exploit sensors data to forecast the time to failure and produce maintenance plans that optimize maintenance costs; this will permit to reduce the risk of unplanned downtime of around 25%, that is estimated to affect up from 5% to 20% of the overall manufacturing productivity. In the service sector, the three IoTwins testbeds concern facility management, by covering online monitoring and operation optimization in IT facilities and smart grids, as well as intervention planning and infrastructure maintenance/renovation on sport facilities on the basis of the data collected by sophisticated and heterogeneous monitoring infrastructures. These three pilots are aimed at improving the environmental footprint of ICT facilities, by increasing the efficiency and resiliency of large critical ICT infrastructures, and at maximizing people safety via online adaptation of evacuation plans (and mobility flows in general) in sport facilities. The five last testbeds have the original goal of showcasing the replicability of the proposed IoTwins methodology in different sectors, the scalability of the adopted solutions, and their capability to help SMEs to generate new business models. For example, some industrial partners are interested to customize and apply the solutions developed in the first set of testbeds in other more articulated deployment environments (larger multi-site production plants in the case of Guala Closures or larger stadium facilities in the case of Barcelona Football Club).

3 Edge Cloud for Distributed Hybrid Twins in IoTwins

IoTwins distributed hybrid twins foster the distribution of trained models and of control intelligence at the cloud, at the topological edges of the interested network localities (edge twins), and possibly also at the IoT network leaves (IoT twins running directly at sensors/actuators/production machinery). The attempt to use the cloud as the only host

for the execution of trained models stopped in the stochastic nature of the Internet, in the need to transfer massive amounts of data towards the cloud, and in the inability to achieve responsiveness in some application fields that demand rapid reaction to events. The edge computing paradigm fills that gap, by bringing computing power and storage to the surrounds of targeted devices, while keeping the advantages of dynamic deployment, resource virtualization, and possible elasticity of the cloud.

The edge of a network operator typically includes several heterogeneous devices that can be used to execute services. These include, on the one hand, (resource-constrained) home gateways, which are able to host only lightweight services, namely network applications executed in lightweight execution environments (e.g., Docker containers, or even processes executed on the bare metal); on the other hand, (possibly fat) servers, up to micro datacenters, located either in a Point of Presence or on Radio Access Network nodes, which can host services with larger resource requirements (CPU, memory). From a software perspective, IoTwins aims to design and realize an edge automation platform able to tackle different aspects: *i)* to enable interoperability so to glue together different edge platforms proposed/employed by IoTwins partners (TTT Nerve, Siemens, etc.); *ii)* to support the dynamic migration of trained models (application/control logic and data) at IoT and edge nodes and between these nodes (if needed); *iii)* to dynamically refine trained models at edge nodes based on local observations (local optimization vs global optimization at the cloud); iv) to enforce soft real-time quality requirements at edge nodes in terms of latency and reliability by enabling fast locally-computed interactions and feedbacks; and v) to support distributed orchestration of Virtualized Network Functions (VNFs), by overcoming the limitations of current orchestrators that are typically centralized and unable to coordinate multiple and highly heterogeneous edge nodes, via extensions of standard-compliant orchestrators, e.g., the ETSI Management and Orchestration (MANO)-compliant OpenBaton.

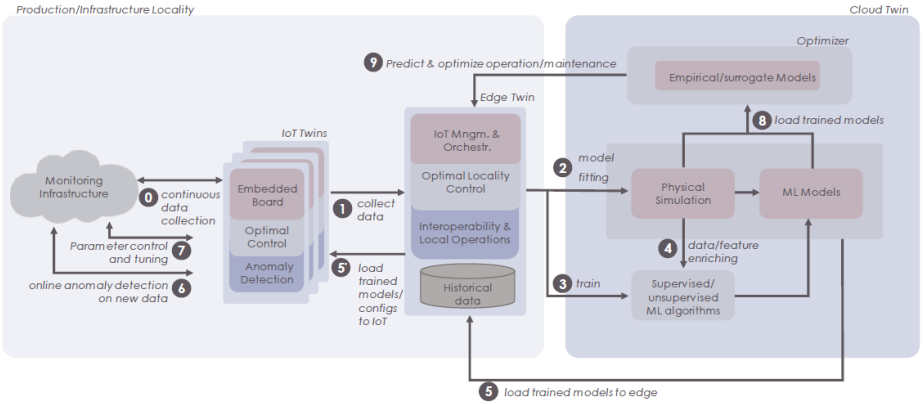

Figure 2. The workflow for big data management and processing by IoTwins hybrid twins, generating a decentralized setting for model learning and refinement.

To remark that IoTwins works on an integrated cloud+edge infrastructure that is easily replicable and highly interoperable, we concisely report here the most relevant standardization efforts in the domain of edge/fog computing that IoTwins is carefully

taking into account. On the one hand, ETSI Multi-access Edge Computing (MEC), emerged in 2014, is often seen as the key enabler for offering ultra-low latency and high bandwidth in edge solutions. The MEC system exposes a standardized and open system that shall be capable of supporting various virtualization techniques as well as the capability to provide a mechanism to edge-based services to discover applications available on other edge hosts. IoTwins edge twins will be based on different technologies, such as TTT Nerve and Open Baton, which is an ETSI MANO-compliant Network Function Virtualization (NFV) orchestrator; Open Baton is part of the OpenSDNCore project driven by Fraunhofer FOKUS and TUB with the objective of providing a compliant implementation of the ETSI NFV MANO specification; recently new extensions for the OpenBaton exploitation over industrial gateway edge nodes have been introduced by Fraunhofer, TUB, and UNIBO. Moreover, the OpenFog Consortium, as one of its first specifications, released the OpenFog Reference Architecture (RA) that clarifies the characteristics and requirements for fog nodes. The OpenFog RA mentions orchestration functionality only superficially so far, and it is expected that new specifications in this direction will be released soon. IoTwins will take into careful consideration these specifications when available. Finally, the Industrial Internet Consortium (IIC) is an international consortium acting as a driver for the development of next generation technology for industrial applications. It has proposed the interesting Industrial Internet RA (IIRA). Fraunhofer FOKUS is a member of the consortium, actively contributing to the discussion, thorough the IIC Task Groups, especially around Smart Factories and Edge Computing. Within IoTwins, UNIBO and ETXE will provide solutions fully compliant with IIRA in the domains of edge deployment, dynamic management, functionality migration, and orchestration.

4 An Original Taxonomy for Decentralized Learning based on Edge Nodes

The unprecedented amount of rich data being generated at the edge of the network — and expected to steadily grow [9] — not only by mobile phones and IoT devices, but also by IIoT sensors and actuators, represents the perfect ingredient to build accurate Machine Learning (ML) and in particular Deep Learning (DL) models for a wide range of applications, from improving the usability of personal devices [10–12] to smartifying the manufacturing process (e.g., defeat reduction, automated self-regulation of prediction processes, and predictive maintenance). However, the sensitive nature of IoT data in general, and particularly of IIoT-generated data through which it is possible to make inferences on characteristics of industrial production processes, implies that there are privacy concerns and responsibilities when managing, storing, and processing those data in centralized locations. Furthermore, the data tsunami produced by edge devices risks to overwhelm the network backbone with unnecessary raw data headed to the cloud, hence a part of these data should instead be consumed or processed in proximity to their sources, as suggested in [9].

Decentralized Learning has recently gained momentum exactly to meet these needs and to become a promising alternative solution to the more traditional cloud-based ML.

Decentralized Learning leverages on the primary idea of leaving training data distributed on the devices that have generated them, by working to enable the learning of joint models via local computation and periodic communications. Similarly to the frameworks designed for distributed settings, i.e., datacenter-oriented deployment environments such as in [13, 14], most Decentralized Learning approaches leverage data-parallel variants of (sequential) iterative optimization algorithms, e.g., Gradient Descent-based algorithms [15]. The current global model is usually replicated on multiple nodes (e.g., edge devices), with each replica independently training on its private dataset, which can be considered as a subset of a global training dataset; in these solutions, each replica also works to produce updates (e.g., locally computed gradients or updated parameters) for the global model; the updates are periodically aggregated (e.g., averaged in the simplest type of merging of local updates towards a next-generation global model) until the model verifies a convergence condition.

The ephemeral nature[1] of these updates — they are meaningful only with respect to the current global model — and their typically lower informative content — compared to the raw data (data processing inequality) — pave the way for upgrading the data owner's privacy. It is also worth noting that the size of a single update is independent of the size of the local training data corpus, thus considerably reducing the necessary network bandwidth if compared with the trivial solution of uploading the whole raw data to a global datacenter.

However, the challenge of learning from decentralized data requires to consider a different optimization setting with respect to the traditional distributed training performed in datacenters, where the data is evenly distributed among different datacenter nodes (often, further assuming that the number of nodes is much less than the ratio between the amount of training examples and the number of nodes), and each machine is supposed to have a representative sample of the underlying data distribution. Furthermore, Decentralized Learning communication costs dominate even more than in datacenter optimizations (e.g., edge devices may have limited connectivity, for example for battery and/or cost motivations, if compared with tightly connected distributed systems such as clusters of machines in datacenters). These considerations have led to the development of new algorithms tailored for the so-called Decentralized Learning federated setting [16], where the assumptions made for the traditional distributed setting do not hold. In federated settings, training examples are massively distributed among a large number of nodes (in particular, the number of nodes can be much higher than the average number of samples stored on a specific node) and unbalanced, i.e., different nodes may have very different amounts (orders of magnitude) of training examples. Furthermore, the data points on each participant may be non-IID[2], i.e., training data available locally are not representative samples of the overall distribution.

Between these two extremes of distributed and federated settings, in IoTwins we claim the suitability of an intermediate setting, where the learning participants are limited in number (e.g., less than 100), and with relaxed connectivity constraints with

[1] As, for example, indicated in Article 5 of the GDPR [53] by the European Parliament.

[2] In this field, the IID acronym is recognized and stands for Independent and Identically Distributed.

respect to the federated setting. However, their data can still be non-IID and unbalanced among different participants. We call this Decentralized Learning setting as ***geographically distributed*** to indicate that learners are not physically located in the same premises. The participants are trusted entities (e.g., manufacturing industries or facility management organizations), which want to collaboratively learn a shared knowledge without disclosing their sensitive data.

A Decentralized Learning framework able to address specific setting peculiarities can be designed by considering different degrees of freedom. The coordination among the learners can be facilitated by a star-shaped network topology that leverages a central entity, namely a parameter server, to distribute the current state of the global model at the beginning of each local iteration, and maintain the state updated during the training task. Participants can directly exchange their locally computed updates as well, in a peer-to-peer fashion, hence not requiring any infrastructure at the price of possible increased communication cost. For example, there could be different models, with each one of them taking random walks in the network and being updated when visiting a new device. As traditional distributed training algorithms, also Decentralized Learning approaches can exploit asynchronous updates to optimize on speed by using potentially stale parameters for local training or wait for the slowest participant to synchronously aggregate all the produced updates without risks to use outdated parameters.

In [17], the authors proposed their pioneering Distributed Selective Stochastic Gradient Descent (DSSGD), where participants asynchronously download and locally replace a fraction of their neural network parameters, run local training, and asynchronously upload a tiny fraction (e.g., 1%) of the computed gradients to a parameter server. The asynchronicity is determined by the absence of coordination among participants; since model updates may occur during local computations, stale gradients [18] could be used for local training. The privacy improvement resulting from the approach in [17] is threefold: *i)* training data remain stored locally, *ii)* participants are aware of the learning objective (and control how much to reveal about their individual models), and *iii)* they can infer the joint model locally without sharing their raw data. Furthermore, to address indirect leakage of sensitive information about any individual point of the training dataset, differentially private mechanisms are employed [19–21].

Among the various Decentralized Learning algorithms inspired by [17], Federated Learning (FL) builds a global model by iteratively aggregating (e.g., averaging) in a synchronous manner the locally computed updates (gradients or model parameters), by leveraging on a parameter server that provides the current model parameters to the selected learning participants at the beginning of each round, i.e., local training iteration [22, 23]. To balance the communication costs, learners might take several steps of the local iterative optimization method (e.g., several steps of mini-batch gradient descent) during a single round.

A plethora of works have tried to address the diverse issues within the context of FL. To prevent the possible leakage of privacy-sensitive information from the updates [24, 25], various techniques have been proposed, such as participant-level differential privacy, i.e., hiding the presence or absence of any specific participant's private dataset in the training [26, 27], secure multi-party aggregation [28], and homomorphic encryption [29]. To cope with the inherent non-IIDness of data in federated settings, which can

cause model divergence or significantly degrade the model accuracy [30], data sharing [30] among participants have been empirically proved to be effective at the cost of less decentralization; moreover, in [31] and [32] the original FedAvg [22] framework is extended by providing both theoretical analysis and empirical evaluation about the improved robustness to data heterogeneity. The latter works also tolerate, respectively, inexact updates and model poisoning [33–35], i.e., malicious participants that manipulate the training process through voluntarily malicious model updates. In addition, several efforts have been made to enhance communication efficiency, targeting the upload link [36–38] or both upload and download links [39–41]. A communication-efficient variation of FL is designed in [42], namely Federated Distillation (FD), a distributed knowledge distillation where learners exchange not the model parameters but the model output, i.e., the communication payload size only depends on the output dimension.

A peer-to-peer fashioned alternative to FL, namely Gossip Learning (GL), has been proposed in [43], although it was already explored when considering the more traditional distributed setting (e.g., [44, 45]). After having initialized a local model, each node sends it to another node, which firstly merges the received model with its current parameters, then updates the resulting model by exploiting its private dataset, and the process repeats. These cycles are not synchronized; hence a node may merge its fresher model with an outdated one — albeit with limited impact thanks to an age parameter associated with models.

To complete the picture of Decentralized Learning strategies, summarized in **Table 1**, we introduce a differently designed method to decouple the training of neural network models from the need for directly accessing the raw data. This technique, sometimes referred as Split Learning [46] or splitNN, horizontally partitions the neural network among the training participant, which holds the shallower layers, and a central entity, which holds the deeper layers. Inter-layer values, i.e., activations and gradients, are communicated in place of raw data. Differently from the previously presented approaches, where the global model is fully replicated on each participant, in Split Learning, all the learners share the neural network deeper layers hosted by the central entity, hence the training process is sequential, albeit distributed. In fact, each participant retrieves the current state of the model either in a peer-to-peer mode, downloading it from the last training participant, or in a centralized mode, downloading it from the central entity, and runs the distributed training using her private dataset. Then, the process is repeated with a different participant, collectively learning a joint model without sharing private raw data.

Although splitNN has demonstrated to reduce computation burden and bandwidth utilization with respect to baseline FL (considering 100 and 500 learners), it has been explicitly designed to allow entities to train deep learning models without sharing patient's raw data in the health domain [47], hence considering a less populous federation of learners with respect to our previously defined federated setting. Furthermore, FL and GL allow on-device inference of the model by design, while this is not true for splitNN that requires a distributed inference unless the complete trained model is provided to the participants. It is worth noting that a less responsive inference determined by the partitioning of the neural network can be considered acceptable in offline health management applications, but not in interactive applications (such as emoji prediction).

To conclude this concise overview of the Decentralized Learning approaches that we are considering as the basis for the IoTwins activities, we emphasize that, as far as we know, there are no examples of Decentralized Learning frameworks explicitly designed to cope with IIoT. In this sense, we highlight the suitability — and the growing appeal — of Agent-Based Computing (ABC) to enable cooperation inside IIoT ecosystems as well as to model and simulate those federations of edge devices [48].

Table 1. Our original taxonomy for Decentralized Learning approaches.

	Optimization Setting	Data Parallel	Model Partition	Network Topology	Update Mode	Exchanged Parameters	
						download	upload
DSSGD [17]	Geodistrib.	YES	NO	Star-shaped	Asynch	model params*	gradients*
FL [22]	Federated	YES	NO	Star-shaped	Synch	model parameters	
FD [42]	Federated	YES	NO	Star-shaped	Synch	model output/ labels	model output/ labels
GL [43]	Federated	YES	NO	Peer-to-peer	Asynch	model parameters	
splitNN [46]	Geodistrib.	NO	YES	Star-shaped Peer-to-peer**		model parameters*	

* means a fraction of (e.g., a fraction of the model parameters).

** In Split Learning there is a centralization entity by design, but we emphasize how the current global state is distributed among participants (i.e., star-shaped topology vs peer-to-peer).

5 Conclusive Remarks and Open Challenges for Future Research on Decentralized Learning

This paper had the ambition to present, through the notable example of the research and development activities planned in the just started IoTwins project, how and why Decentralized Learning based on edge cloud computing could be a suitable solution for IIoT applications where there is the need to consider central requirements such as wide decentralization, high scalability, limited latency, locality-specific optimizations, and sovereignty on manufacturing process data. In addition to presenting the general solution guidelines and high-level architecture adopted uniformly in all the IoTwins testbeds, from predictive maintenance applications to latency-critical quality control for production processes, the paper provided the community with an original contribution in terms of categorization of the emerging Decentralized Learning approaches, in particular for the solutions that target medium-scale deployment environments for the federated setting, such as in most usual industrial scenarios nowadays. This taxonomy is guiding our architectural and design choices in IoTwins and we hope that could be useful to the whole community of researchers in the field by shedding new light on the possible liberty degrees available (and their associated differentiated suitability to achieve different application domain or deployment requirements) in the development of Decentralized Learning solutions for the IIoT.

In addition, this initial promising work has already highlighted some primary directions of major interest and associated technical challenges that short/medium-term research activities in the field will have to deal with.

First, we claim that fog-aware and edge-aware Decentralized Learning solutions have already demonstrated to be very promising to scale the training cooperation, e.g.,

to further reduce the traffic headed to the cloud taking advantage of intermediate update-aggregator entities, but their industrial exploitation call for significant additional efforts towards more mature and standard-compliant platforms for edge/fog-based distributed AI for IIoT. For example, the hierarchical FL presented in [49] adopts the MEC standard specification [50]: edge servers aggregate updates from their localities and forward partial aggregations to the cloud to contribute to the global model; to increase openness and interoperability, these edge servers expose a MEC-compliant API that should help in integrating them in full 5G infrastructures. But several aspects ae still uncovered, as mobility management support and container-oriented management and orchestration, just to name a few. A similar solution [51], specifically targeting IoT devices, leverages the fog layer [52] to lighten resource-constrained devices and to prevent computation/communication bottlenecks. Fog nodes gather transformed data (i.e., the original data projected to a lower-dimensional space) from IoT devices and compute differentially private updates (i.e., clipped gradients perturbed with Gaussian noise), before heading them to the cloud. This communication-efficient solution also offers enhanced privacy (lowering the dimensionality of raw data contributes to limit possible information leakage, still remaining useful for learning), as well as it enables computationally constrained IoT devices to participate to learning tasks. Furthermore, fog nodes can be queried in place of the cloud, resulting in a trade-off between the low on-device inference time and the high inference time of cloud-based ML.

Second, it is worth noting that the Decentralized Learning frameworks introduced so far have been developed and validated with the goal of supervised learning in mind, i.e., assuming that the training examples gathered from edge devices are always labelled. This assumption does not hold in all the different reifications of federated settings in IIoT application domains of practical interest. Indeed, while data generated by the interaction of users with smartphones or IoT devices can be easily labeled (e.g., the choices of users with respect to a range of suggested emojis in intelligent keyboards, as in Google-supported solutions in this research area), the data harvested from the monitoring of industrial manufacturing processes may not be automatically labeled and may require non-trivial classification steps, which could be part of the functionality offered by distributed hybrid twins. Related technical challenges, e.g., how to combine input from in-the-field monitoring with automated labelling and Decentralized Learning in distributed and lazily coordinated edge nodes, are still largely unexplored and will probably gain similar relevance as the other main technical issues currently associated with Decentralized Learning in federated settings (heterogeneity, privacy, communication-efficiency, and scalability).

Acknowledgements

This work has been accomplished with the partial support of the EU H2020 IoTwins Innovation Action project (under grant agreement ID = 857191, Sept. 2019 – Aug. 2022, see https://cordis.europa.eu/project/rcn/223969/factsheet/en).

References

1. Ericsson White Paper: Cellular networks for Massive IoT – enabling low power wide area applications, (2016).
2. Xu, L. Da, He, W., Li, S.: Internet of things in industries: A survey. IEEE Trans. Ind. Informatics. 10, 2233–2243 (2014). https://doi.org/10.1109/TII.2014.2300753.
3. Zhu, R., Zhang, X., Liu, X., Shu, W., Mao, T., Jalaian, B.: ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial lot. IEEE Access. 3, 2366–2378 (2015). https://doi.org/10.1109/ACCESS.2015.2501644.
4. Casadei, R., Fortino, G., Pianini, D., Russo, W., Savaglio, C., Viroli, M.: A development approach for collective opportunistic Edge-of-Things services. Inf. Sci. (Ny). 498, 154–169 (2019). https://doi.org/10.1016/j.ins.2019.05.058.
5. Shi, W., Dustdar, S.: The Promise of Edge Computing. Computer (Long. Beach. Calif). 49, 78–81 (2016). https://doi.org/10.1109/MC.2016.145.
6. IoTwins Project - Distributed Digital Twins for industrial SMEs: a big-data platform, https://cordis.europa.eu/project/rcn/223969/factsheet/en, last accessed 2019/11/13.
7. Vlacheas, P., Giaffreda, R., Stavroulaki, V., Kelaidonis, D., Foteinos, V., Poulios, G., Demestichas, P., Somov, A., Biswas, A., Moessner, K.: Enabling smart cities through a cognitive management framework for the internet of things. IEEE Commun. Mag. 51, 102–111 (2013). https://doi.org/10.1109/MCOM.2013.6525602.
8. Arrayent | The IoT Platform for Trusted Brands, https://www.arrayent.com/, last accessed 2019/11/19.
9. Cisco Public: Cisco Global Cloud Index: Forecast and Methodology, 2016–2021, http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf, last accessed 2019/11/08.
10. Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., Beaufays, F.: Applied Federated Learning: Improving Google Keyboard Query Suggestions. (2018).
11. Chen, M., Mathews, R., Ouyang, T., Beaufays, F.: Federated Learning Of Out-Of-Vocabulary Words. 1–6 (2019).
12. Ramaswamy, S., Mathews, R., Rao, K., Beaufays, F.: Federated Learning for Emoji Prediction in a Mobile Keyboard. (2019).
13. Dean, J., Corrado, G.S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M.Z., Ranzato, M.A., Senior, A., Tucker, P., Yang, K., Ng, A.Y.: Large scale distributed deep networks. Adv. Neural Inf. Process. Syst. 2, 1223–1231 (2012).
14. Zhang, S., Choromanska, A., Lecun, Y.: Deep learning with elastic averaging SGD. Adv. Neural Inf. Process. Syst. 2015-Janua, 685–693 (2015).
15. Ruder, S.: An overview of gradient descent optimization algorithms. 1–14 (2016).
16. Konečný, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated Optimization: Distributed Machine Learning for On-Device Intelligence. 1–38 (2016).
17. Shokri, R., Tech, C.: Privacy-Preserving Deep Learning. 1310–1321.
18. Pan, X., Chen, J., Monga, R., Bengio, S., Jozefowicz, R.: Revisiting Distributed Synchronous SGD. 1–10 (2017).
19. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in

private data analysis. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2006). https://doi.org/10.1007/11681878_14.

20. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 4004 LNCS, 486–503 (2006). https://doi.org/10.1007/11761679_29.
21. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. 9, 211–487 (2013). https://doi.org/10.1561/0400000042.
22. Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., Agüera y Arcas, B.: Communication-efficient learning of deep networks from decentralized data. Proc. 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017. 54, (2017).
23. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H.B., Van Overveldt, T., Petrou, D., Ramage, D., Roselander, J.: Towards Federated Learning at Scale: System Design. (2019).
24. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep Models under the GAN: Information leakage from collaborative deep learning. Proc. ACM Conf. Comput. Commun. Secur. 1, 603–618 (2017). https://doi.org/10.1145/3133956.3134012.
25. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting Unintended Feature Leakage in Collaborative Learning. 691–706 (2019). https://doi.org/10.1109/sp.2019.00029.
26. McMahan, H.B., Ramage, D., Talwar, K., Zhang, L.: Learning Differentially Private Recurrent Language Models. 1–14 (2017).
27. Geyer, R.C., Klein, T., Nabi, M.: Differentially Private Federated Learning: A Client Level Perspective. 1–7 (2017).
28. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. Proc. ACM Conf. Comput. Commun. Secur. 1175–1191 (2017). https://doi.org/10.1145/3133956.3133982.
29. Phong, L.T., Aono, Y., Hayashi, T., Wang, L., Moriai, S.: Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. IEEE Trans. Inf. Forensics Secur. 13, 1333–1345 (2018). https://doi.org/10.1109/TIFS.2017.2787987.
30. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated Learning with Non-IID Data. (2018).
31. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated Optimization in Heterogeneous Networks. (2018).
32. Xie, C., Koyejo, S., Gupta, I.: SLSGD: Secure and Efficient Distributed On-device Machine Learning. (2019).
33. Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S.: Analyzing Federated Learning through an Adversarial Lens. 1–19 (2018).
34. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How To Backdoor Federated Learning. (2018).
35. Fung, C., Yoon, C.J.M., Beschastnikh, I.: Mitigating Sybils in Federated Learning Poisoning. 1–16 (2018).

36. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated Learning: Strategies for Improving Communication Efficiency. 1–10 (2016).
37. Tao, Z., William, C.: eSGD : Communication Efficient Distributed Deep Learning on the Edge. {USENIX} Work. Hot Top. Edge Comput. (HotEdge 18). 1–6 (2018).
38. Agarwal, N., Suresh, A.T., Yu, F., Kumar, S., Brendan McMahan, H.: CPSGD: Communication-efficient and differentially-private distributed SGD. Adv. Neural Inf. Process. Syst. 2018-Decem, 7564–7575 (2018).
39. Chen, Y., Sun, X., Jin, Y.: Communication-Efficient Federated Deep Learning with Asynchronous Model Update and Temporally Weighted Aggregation. 1–10 (2019).
40. Sattler, F., Wiedemann, S., Müller, K.-R., Samek, W.: Robust and Communication-Efficient Federated Learning from Non-IID Data. 1–17 (2019).
41. Caldas, S., Konečny, J., McMahan, H.B., Talwalkar, A.: Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. (2018).
42. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.-L.: Communication-Efficient On-Device Machine Learning: Federated Distillation and Augmentation under Non-IID Private Data. (2018).
43. Hegedűs, I., Danner, G., Jelasity, M.: Gossip learning as a decentralized alternative to federated learning. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 11534 LNCS, 74–90 (2019). https://doi.org/10.1007/978-3-030-22496-7_5.
44. Blot, M., Picard, D., Cord, M., Thome, N.: Gossip training for deep learning. 1–5 (2016).
45. Daily, J., Vishnu, A., Siegel, C., Warfel, T., Amatya, V.: GossipGraD: Scalable Deep Learning using Gossip Communication based Asynchronous Gradient Descent. (2018).
46. Gupta, O., Raskar, R.: Distributed learning of deep neural network over multiple agents. J. Netw. Comput. Appl. 116, 1–8 (2018). https://doi.org/10.1016/j.jnca.2018.05.003.
47. Vepakomma, P.: Split learning for health : Distributed deep learning without sharing raw patient data. (2018).
48. Savaglio, C., Ganzha, M., Paprzycki, M., Bădică, C., Ivanović, M., Fortino, G.: Agent-based Internet of Things: State-of-the-art and research challenges. Futur. Gener. Comput. Syst. 102, 1038–1053 (2020). https://doi.org/10.1016/j.future.2019.09.016.
49. Liu, L., Zhang, J., Song, S.H., Letaief, K.B.: Edge-Assisted Hierarchical Federated Learning with Non-IID Data.
50. Mao, Y., Member, S., You, C., Member, S., Zhang, J., Member, S.: A Survey on Mobile Edge Computing : The Communication Perspective. 19, 2322–2358 (2017).
51. Lyu, L., Bezdek, J.C., He, X., Jin, J.: Fog-Embedded Deep Learning for the Internet of Things. IEEE Trans. Ind. Informatics. 15, 4206–4215 (2019). https://doi.org/10.1109/TII.2019.2912465.
52. Mahmud, R., Kotagiri, R., Buyya, R.: Fog Computing : A Taxonomy, Survey and Future Directions. In: Internet of everything. pp. 103–130. , Singapore (2018).
53. THE EUROPEAN PARLIAMENT: REGULATION (EU) 2016/679, https://eur-lex.europa.eu/eli/reg/2016/679/oj, last accessed 2019/11/09.