Opportunities of Federated Learning in Connected, Cooperative, and Automated Industrial Systems

Stefano Savazzi, Monica Nicoli, Mehdi Bennis, Sanaz Kianoush, and Luca Barbieri

The authors explore emerging opportunities of federated learning for the next-generation networked industrial systems. Open problems are discussed, focusing on cooperative driving in connected automated vehicles and collaborative robotics in smart manufacturing.

Abstract

Next-generation autonomous and networked industrial systems (i.e., robots, vehicles, drones) have driven advances in ultra-reliable low-latency communications (URLLC) and computing. These networked multi-agent systems require fast, communication-efficient, and distributed machine learning (ML) to provide mission-critical control functionalities. Distributed ML techniques, including federated learning (FL), represent a mushrooming multidisciplinary research area weaving together sensing, communication, and learning. FL enables continual model training in distributed wireless systems: rather than fusing raw data samples at a centralized server, FL leverages a cooperative fusion approach where networked agents, connected via URLLC, act as distributed learners that periodically exchange their locally trained model parameters. This article explores emerging opportunities of FL for the next-generation networked industrial systems. Open problems are discussed, focusing on cooperative driving in connected automated vehicles and collaborative robotics in smart manufacturing.

Introduction

The rapid transformation of industrial systems, driven by digitization and 5G communication evolution, has led to extensive research initiatives on manufacturing and automotive verticals. These include, for example, Industry 4.0 (I4.0), at the European level, the European Factories of the Future Research Association (EFFRA, effra.eu) the 5G Alliance for Connected Industries and Automation (5g-acia.org), and the 5G Automotive Association (5gaa.org). The envisioned smart industrial systems rely on networked machines with increasing levels of intelligence and autonomy, moving far beyond traditional low-cost low-power sensors. According to industrial requirements, such machines are required to support:

1. Autonomous and adaptive decision making in dynamic situations with mobile operators/equipment, device-less human-machine interfaces, and time-varying environments
2. Big-data-driven training of large-size machine learning (ML) models for decision making
3. Ultra-reliable low-latency communications (URLLC) [1] for mission-critical device-to-device (D2D) operations

Networked and cooperative intelligent machines have recently opened new research opportunities that target the integration of distributed ML tools with sensing, communication, and decision operations. Cross-fertilization of these components is crucial to enable challenging collaborative tasks in terms of safety, reliability, scalability, and latency.

Among distributed ML techniques, federated learning (FL) [2, 3] has been emerging for model training in decentralized wireless systems. Model parameters, namely weights and biases in deep neural network (DNN) layers, are optimized collectively by cooperation of interconnected devices acting as distributed learners. In contrast to conventional edge-cloud ML, FL does not require local training data to be sent to the server, which may be infeasible in mission-critical settings with extremely low latency and data privacy constraints.

The most popular FL implementation, namely federated averaging [2], alternates between the computation of a *local model* at each device and a round of communication with the server for learning of a *global model*. Local models are typically obtained by minimizing a local loss function via stochastic gradient descent (SGD) steps [3], using local training examples and target values.

Federated averaging is privacy-preserving by design, as it keeps the training data on-device. However, it still leverages the server-client architecture, which might not be robust to data poisoning attacks and scalability needs.

Overcoming this issue mandates moving toward fully decentralized FL solutions relying *solely* on local processing and cooperation among end machines. As shown in Fig. 1, the device sends its local ML model parameters to neighbors and receives in return the corresponding updates. Next, it improves its local parameters by fusing the received contributions. This procedure continues until convergence.

The article addresses the opportunities of emerging distributed FL tools specifically tailored for systems characterized by autonomous industrial components (vehicles, robots). FL is first proposed as an integral part of the sensing-decision-action loop. Next, novel decentralized FL

Stefano Savazzi and Sanaz Kianoush are with Consiglio Nazionale delle Ricerche;
Monica Nicoli and Luca Barbieri are with Politecnico di Milano; Mehdi Bennis is with Oulu University.
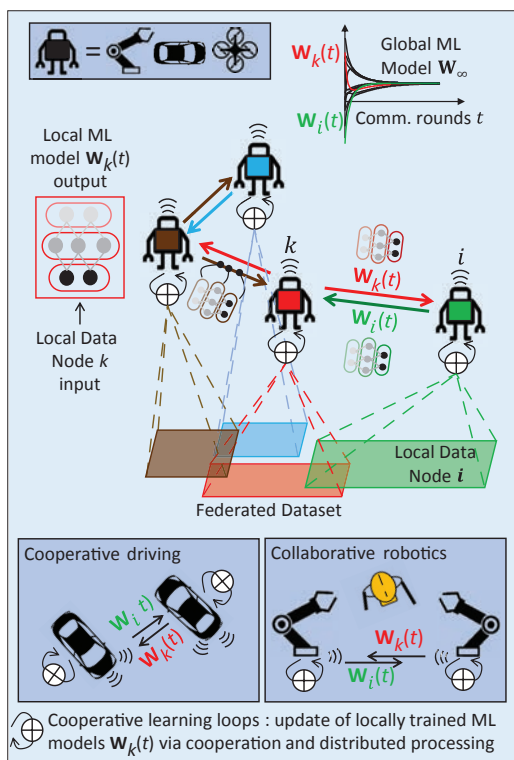
FIGURE 1. Decentralized learning with federated datasets. Examples in mission-critical control applications: collaborative industrial robotics and cooperative automated vehicles.

FIGURE 2. Decentralized FL in the sensing-decision-action loop.

tools and emerging research challenges are highlighted. The potential of FL is further elaborated with consideration primarily given to mission-critical control operations in the field of cooperative automated vehicles and densely interconnected robots. Analysis with real data on a practical usage scenario reveals FL as a promising tool underpinned by URLLC communications.

SENSING, DECENTRALIZED LEARNING, AND COMMUNICATION CO-DESIGN

Decentralized FL solutions have great potential for industrial 5G and beyond 5G (B5G) verticals. In automated industrial processes, decentralized FL imbues intelligence directly into the end machines, which become smart cooperative agents. Figure 2 depicts a schematic of a cooperative and automated multi-agent industrial system. It consists of connected machines performing collaborative tasks, and integrating ML model training within the sensing-decision-action loop. The ML model outputs might be scenario-dependent predictions of a physical process, or rather value functions to be used for policy improvements (i.e., reinforcement learning). Outputs are fed to the machine controller for local decisions or actuations. Training of the ML model calls for highly efficient knowledge discovery operations based on the overall training data collected by *all* the machines performing the same task (federated dataset) rather than local data only. Recent advancements of FL constitute a first but significant step toward *collaborative learning* and particularly the understanding of how ML could be distributed over networked devices without centralized orchestration. Collaborative learning
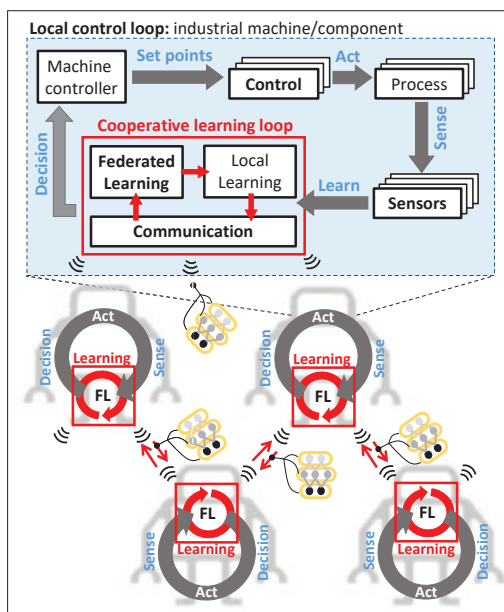
underpins local decisions and allows the networked machines to augment their ML model by sharing ego knowledge. As part of the sensing-decision-action loop, emerging FL tools are expected to target three fundamental challenges:
1. Get over the restrictions of server-client architectures and movements of large, unstructured, raw datasets over D2D wireless links, in favor of (typically) sparse ML model parameters exchange.
2. Optimally balance opportunistic (ego) learning based on local training data and collaborative learning (leveraging neighbors' experience), with the goal of steering convergence.
3. Learn and re-train continuously the models over URLLC links to adapt to changes in the data distribution, environment, process, or situation.

DECENTRALIZED FL: EMERGING TRENDS

Decentralized FL alternates the mutual D2D exchange of local ML models $\mathbf{W}_k(t)$, at device k and time t, with the on-device minimization of a local loss function. The goal is to promote convergence to a global model $\mathbf{W}_\infty$ that minimizes a global loss, decomposed into the sum of local losses. Fully distributed FL implementations run on top of D2D networks characterized by arbitrary connectivity graphs. The collaborative learning is based on decentralized SGD (DSGD), which guarantees convergence under strong convexity assumptions of local loss functions [4] and networks with doubly stochastic adjacency matrix [3]. DSGD alternates on-device SGD steps to obtain $\mathbf{W}_k(t)$ with the mutual exchange of model parameters to steer convergence. Mutual exchange is regulated by gossip [5], consensus, or diffusion algorithms [6, 7].

CONSENSUS, DIFFUSION, AND GRADIENT NEGOTIATIONS

In consensus-based approaches [5, 6], federated nodes exchange local ML model parameters and update them sequentially by distributed averag-
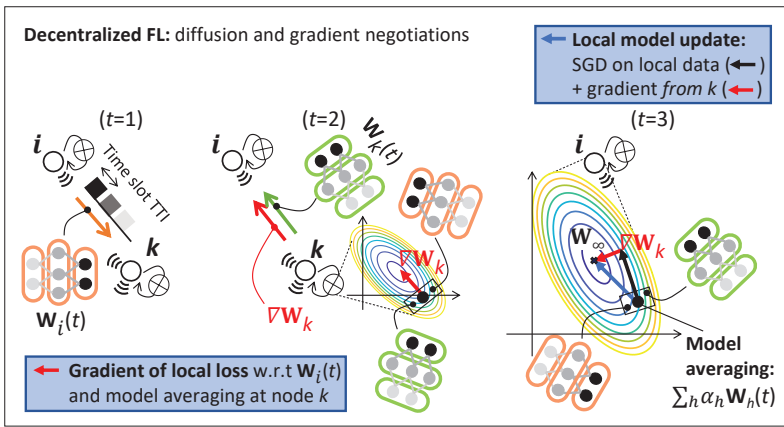
FIGURE 3. Decentralized FL with diffusion: communication, gradient negotiations ($t = 1, 2$), and computing rounds ($t = 3$).

ing. For the mutual exchange of models, nodes might select random, time-varying, or optimized [8–12] partners. In gossipgrad [5], the selection policy creates a virtual network where each cooperating agent is connected to, typically, two other nodes. Convergence time and loss are ruled by the specific choice of model update operations. Some updating strategies favor fast convergence [6], while others target model accuracy [12]. Considering DNN model training, the neural network layers are also typically learned separately and independently. Therefore, on-device optimization and networking phases can run in parallel. Communication overhead scales linearly with the model size and the number of cooperating agents. However, non-independent identically distributed (non-IID) federated datasets and large model size penalize the rate of convergence [3].

Diffusion strategies incorporate a gradient negotiation phase, described in Fig. 3, where devices exchange the information about how neighbor models should be adjusted considering local data. Diffusion virtually expands the local training dataset, and boosts the convergence compared to gossip. Gradient exchange strategies are often regulated by request-reply negotiation stages between neighbors [7]. In the example of Fig. 3, local model i, $\mathbf{W}_i(t = 1)$, is sent to neighbor k at time $t = 1$ to start a negotiation. The received model is used by the neighbor k to compute a gradient vector $\nabla\mathbf{W}$ using the local loss. Both the gradient vector and the local model are then fed back to device i, which started the negotiation ($t = 2$). On-device optimization ($t = 3$) finally adjusts the local model by combining the gradients obtained from local loss with those received from the neighbor (combine-and-adapt [7]). In [6] the negotiation resorts to a two-stage asynchronous procedure: convergence speed improves at the cost of larger communication overhead. Exploring this trade-off is currently an open challenge.

In addition to the above tools, distributed ledger technologies can also be applied to decentralized training [11] by validating clients' model updates via a series of validation steps (proof of work or others). As a result, decentralized FL is transformed into a market of expert model training nodes and validators. The development of robust FL designs against data poisoning and adversarial manipulations is still an open problem.

IMPROVING COMMUNICATION EFFICIENCY

Reliable and low-latency D2D communications serve as the backbone for distributed FL computations. Transmission of model parameters must be extremely reliable with packet error rate down to 10^{-8} to prevent frequent retransmissions penalizing the convergence rate [6]. Transmission time intervals (TTIs) should be aligned with the data dynamics and the computation times required to perform SGD and model adaptation, targeting 5 ms and below [1]. The model size also affects communication design choices: DNN models often used in industrial applications contain >15,000 parameters per layer, usually extremely sparse.

Although vanilla FL assumes noiseless or rate-limited communications [3], recently proposed *digital* and *analog* designs quantify the effects of intermittent communications, time-varying fading, and interference. Digital implementations of FL require each device to communicate with peers over half-duplex links via time scheduled wireless access. Popular solutions to limit the model size, and thus the communication overhead, are quantization, sparsification, and distillation [8]. Sparsifying operators select a subset of informative ML parameters, often also improving the model generalization.

More recently, analog FL and hybrid analog-digital implementations have been proposed for fast and synchronous model averaging [4]. Analog FL gets over the restrictions of time scheduled access as it exploits the superposition property of dense wireless transmissions when averaging the neighbor models. Each device receives the superposition of the ML models simultaneously transmitted by the neighborhood. When ML parameters are sparse and sent as uncoded/uncompressed, Lasso type recovery can be adopted for decoding and reconstruction [9].

Besides low-level communications, optimization of resource allocation has been considered to find the best trade-off between on-device computation (SGD updates) and wireless channel use for parameters exchange [10]. Optimized medium access control can also substantially reduce the FL loss [12]: for example, scheduling should reward devices having high-quality data compared to those possessing few or redundant samples [11].

EMERGING RESEARCH CHALLENGES

The rollout of new decentralized communication paradigms in B5G is expected to bridge the gap between deep learning and wireless networking research, at the same time raising unprecedented challenges. For example, current FL designs generally ignore the underlying dynamics of the network graph, the presence of intermittent communication links, or weakly connected components [3]. Research in this direction should focus on balancing local data collection, model adaptations, and cooperation from selected agents. Digital, analog, or hybrid implementations should be considered based on devices' mobility, model size, computational and bandwidth resources, as well as connectivity quality. Learning an optimal policy for graph (and/or neighbor) selection while training the ML model, namely *simultaneously* learning the network graph and the model, is also a promising direction.

Automated systems are often characterized by heterogeneous devices performing distinct but related tasks. Collaborative learning of multiple functions, namely *multitask learning* [13], will have a relevant impact on several industrial applications. Understanding when cooperation, in the form of consensus, or any decentralized agreement protocol, can better steer convergence compared to opportunistic or ego behaviors is an open problem of wide interest. The analysis must also take into consideration the cost of distributed computations, the federated data distribution, and the network designs for URLLC.

Using the I4.0 vision and emerging B5G verticals as guidelines, in what follows, opportunities of FL are highlighted for the needs of selected mission-critical control applications: cooperative automated driving and collaborative robotics.

DISTRIBUTED INTELLIGENCE FOR COOPERATIVE AUTOMATED DRIVING

Vehicular URLLC leverages B5G connectivity to enable flexible vehicle-to-everything (V2X) interactions with road infrastructure (V2I) and other vehicles (V2V). Distributed ML over V2X networks will play a central role in the cooperative intelligent transportation system (C-ITS), enabling level 4–5 cooperative automated driving functionalities [14].

Cooperative automated vehicles share maps of the driving environment (Fig. 4) using V2X URLLC to extend the range and resolution of their ego imaging sensors (radar, camera, lidar). Besides improving detection and localization of safety-related events, vehicles can also negotiate the maneuvering and synchronize to a common mobility pattern, forming tight autonomous driving convoys and increasing traffic efficiency. All the above scenarios are characterized by time-critical functions that must be implemented on a closer-to-the-ground cloud, with part of the cooperative computational tasks performed locally, by pushing intelligence into the vehicles rather than on the mobile edge cloud (MEC) [14].

Decentralized FL techniques [6] are promising solutions for these time-sensitive applications. They require the vehicles to transmit smaller models that can be aggregated at the roadside unit (RSU) or by vehicles via ultra-low-latency (3 ms) and highly reliable (10^{-5}) V2X connectivity. The exchange of local ML model parameters, rather than raw data, is expected to decrease the learning time, allowing quick reaction to unexpected events and making safety-critical decisions. Domain-specific FL designs must also account for intermittent communications, time-varying network graphs, and non-IID training datasets changing quickly over time, thus evolving according to vehicle motions, with speeds up to 250 km/h (3rd Generation Partnership Project, 3GPP, TR 22.886), and the surrounding environment. In such cases, a transitory phase is expected where vehicles with outdated or partially trained models will coexist with highly automated ones and benefit from their cooperation. The federation with fully equipped vehicles will assist lower-level vehicles to get an augmented vision of the driving environment, even if equipped with less accurate sensors. Balancing between centralized and decentralized FL implementations for energy opti-
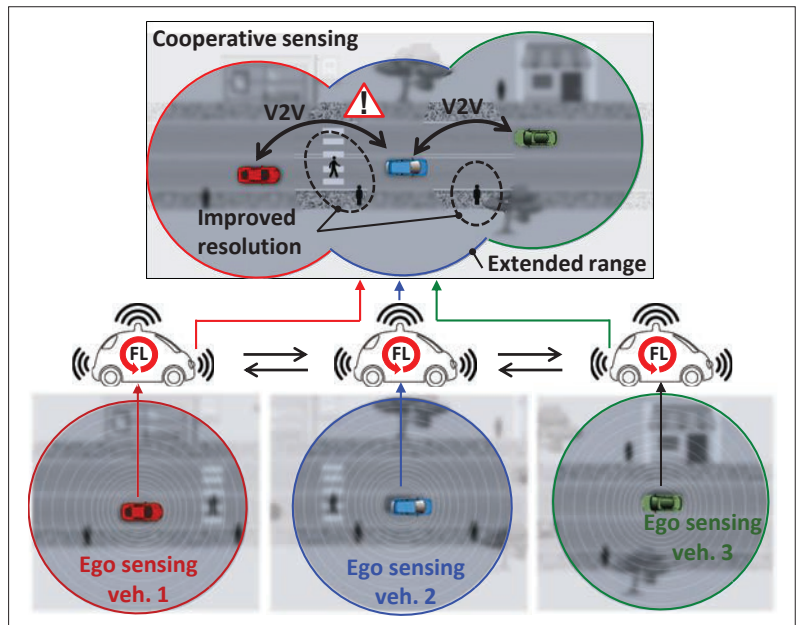

FIGURE 4. Distributed FL for cooperative perception by real-time fusion of imaging data at different vehicles.

mization is also a main issue. Besides cooperative driving, to improve safety and reliability, the vehicles also need to learn the network latency in a distributed manner [1], whereby decentralized FL and its variants are instrumental. Open problems further relate to the tension between local and global models, and the impact of large numbers of vehicles.

Figure 4 highlights a distributed ML-assisted cooperative perception task where vehicles fuse their sensor data in a decentralized implementation to get an extended vision of the environment. Fusing six-dimensional dynamics (i.e., position and orientation, from global navigation satellite systems, GNSSs, and inertial measurement units, IMUs) and imaging sensor data (i.e., lidar, camera, radar) from different vehicles improves the location sensing accuracy making real-time responses feasible. A main challenge is the association of unlabeled imaging measurements at different vehicles to jointly sensed features for cooperative simultaneous localization and mapping (SLAM). Large data volumes and computational complexity are also critical challenges. Decentralized FL is a promising candidate, as it is able to learn a common model for data association and fusion from local raw data, limiting the V2X exchange to model parameters.

DECENTRALIZED FL FOR COBOTS

The development of collaborative robotics in advanced manufacturing environments (cobots) can be interpreted as parallel to autonomous driving. Standardized in International Standards Organization (ISO)/TS 15066, collaborative robot operations allow the machines (industrial manipulators, vehicles) installed in a shared workspace to move concurrently with human operators inside fenceless environments. Connected and collaborative robots are transforming industrial workspaces, such as assembly lines in the example of Fig. 5a, and represent a challenging ground for the development of decentralized FL tools. First, robots operate in increasingly complex
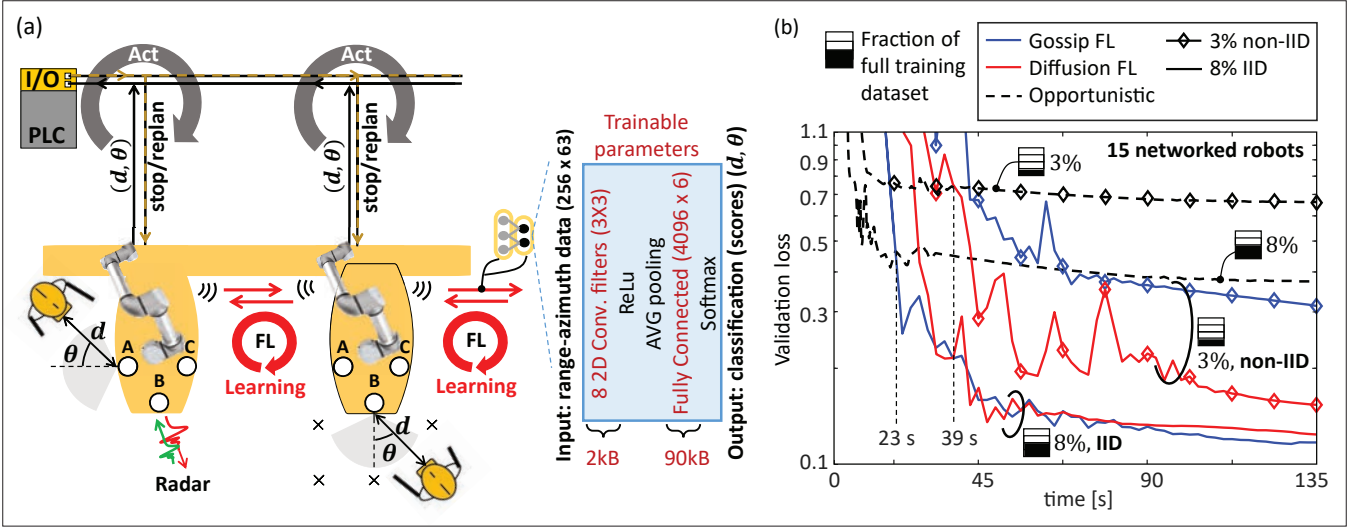
FIGURE 5. a) Decentralized FL setup in a cobot environment: frequency-modulated continual wave (FMCW), multiple-input multiple-output (MIMO) radar locations (A,B,C), and ML model-based classification of subject positions; b) FL loss for gossip (blue) and diffusion (red) vs. time and over a D2D network of 15 robots. IID (no markers) and non-IID (diamond markers) federated datasets using 8 percent and 3 percent of the full training dataset, respectively. D2D network with TTI = 3 ms, BLER = 10^{-9}. Dashed lines refer to the opportunistic learning case.

and time-varying environments. Network design, paired with distributed learning tools, must consider the problem of continual learning (and periodic re-training) over URLLC communication links to track variations of data dynamics caused by changes of the workflow process. Second, robots might perform distinct tasks, although these can be considered as strongly related, targeting a common manufacturing process (assembly or disassembly tasks) and efficient workspace sharing. In such cases, opportunistic (ego) and collaborative approaches should be carefully assessed. For example, FL can boost the model training for some tasks, such as vision functions, in common with many robots.

To shed light on some of the above challenges, we resort to a typical mission-critical control problem in I4.0 workspaces. The goal is to plan the motions of a robotic manipulator based on the information about the positions of human operators sharing the workspace. Operator locations are classified using an ML model, described in Fig. 5a, that computes the human-robot distance d and the direction of arrival (DOA) θ in real time, using six regions of interest as target classes. During the online workflow, the ML model can be trained/updated continuously by decentralized FL and consensus (cooperative learning loop). The sensing-decision-action loop is supervised by an industry-standard programmable logic controller (PLC) that makes decisions about robot emergency stop or trajectory replanning [15], based on the information about subject positions. Decentralized FL uses D2D links, runs in parallel to the sensing-decision-action loop, and thus takes some load off of the PLC network, whose resources must be reserved for robot motion control. All the robots support vision functions implemented by three radars working in the 77–81 GHz band with a field of view of 120° each. Radars produce the raw data [15] that are fed into the ML model.

RESULTS AND DISCUSSIONS

The example of Fig. 5b analyzes the performance of decentralized FL approaches compared to ego (i.e., opportunistic learning) considering the

application case previously described. In particular, the robots might choose to combine in groups of 15 agents and implement FL over a D2D network, or instead act opportunistically, thus learning from local data only, disabling the D2D radio interface. The examples also highlight the impact of federated data distribution on convergence time and validation loss. When federated data is IID partitioned, each device uses 8 percent of the full training dataset of 900 samples. In the more challenging non-IID scenario, only 3 percent of the data is used: batches contain examples for only 5 of the 6 output target classes, chosen randomly.

D2D communications are organized into consecutive frames and use time-division multiple access (TDMA) scheduling. Frames have payload 1 kB and TTI of 3 ms, in line with URLLC [1]. Notice that a frame drop (block error rate, BLER, is 10^{-9}) causes the loss of a layer update and an increase of the convergence time. FL and networking have been simulated on a virtual environment but using real data from the plant. The environment allows deployment of networked agents acting as virtual robots and learning over a configurable federated dataset (IID and non-IID).

The simulations of Fig. 5b quantify the average validation loss [3] experienced by the deployed agents vs. time. This allows assessment of the time required by the consensus rounds, as relevant for real-time implementation. Gossip and diffusion FL approaches, described previously, are evaluated over a k-regular D2D network consisting of robots with two neighbors each. Implementation of gossip is based on [5]: on each round, the agents randomly choose a single neighbor from the neighborhood, exchange the local ML models, and adapt them by averaging, followed by SGD on local data. Diffusion FL implements the gradient negotiations described in Fig. 3: In each round, the agents now exchange the local ML models *and* the gradients $\nabla \mathbf{W}$ with a single neighbor, again chosen randomly. For both cases, the time span of an individual round consists of the transmission of the ML parameters in the assigned

frames, the model adaptation, and the SGD steps (30 ms per round). Before D2D transmission, the ML model parameters are sparsified [9] to limit the communication overhead, measured in kilobytes per round.

Diffusion FL boosts the model training for robots possessing little data (3 percent) and insufficient examples (non-IID). Compared to ego learning, cooperation gives smaller loss after a training period of 39 s, which is enough to track workflow changes. Diffusion needs considerable communication overhead (220 kB per round) due to exchange of gradients. Convergence time thus increases with network size. Notice that small but sudden increases of the loss are observed on some rounds when averaging the neighbor models. These effects are more visible when the local models are trained with non-IID data and can be mitigated by learning rate optimization [6]. Gossip requires twice lower number of frames (92 kB per round), but it is less effective over non-IID data. However, it is still useful for refining models trained with IID data as it improves ego approaches after 23 s. Opportunistic learning converges fast as it does not utilize D2D communications, but experiences a large loss: it is a viable solution only for agents possessing enough data and performing specialized tasks that do not require much retraining.

Conclusions and Future Directions

In this article, we explore opportunities and applications of FL in networked and automated industrial systems, underpinned by D2D wireless communications. Open problems and challenges have been discussed, focusing on manufacturing and automotive B5G verticals. Decentralized FL enables the cooperative learning of ML models. It can be seamlessly integrated into the application-dependent sensing-decision-action loop within each automated entity to improve knowledge discovery operations. Learning and retraining continuously to follow the changing dynamics of the environment has a profound impact on the networking co-design, capitalizing on model sparsity and superposition properties of the wireless links.

Future research is expected to target increasingly complex mobile environments characterized by heterogeneous devices cooperating to learn distinct, but related, functions. The choice between opportunistic and cooperative behaviors will largely depend on the URLLC design. Emerging FL tools are promising in cooperative automated driving, leveraging V2X interactions, and in collaborative robotics for distributed learning in complex and dynamic workflows.

References

[1] M. Bennis *et al.*, "Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," *Proc. IEEE*, vol. 106, no. 10, Oct. 2018, pp. 1834–53.
[2] J. Konečný *et al.*, "Federated Optimization: Distributed Machine Learning for On-Device Intelligence," CoRR, 2016; http://arxiv.org/abs/ 1610.02527.
[3] P Kairouz *t al.*, "Advances and Open Problems in Federated Learning"; https://arxiv.org/abs/1912.04977.
[4] M. M. Amiri *et al.*, "Federated Learning Over Wireless Fading Channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, May 2020, pp. 3546–57.
[5] J. A. Daily *et al.*, "Gossipgrad: Scalable Deep Learning Using Gossip Communication Based Asynchronous Gradient Descent"; https://arxiv.org/abs/1803.05880.
[6] S. Savazzi *et al.*, "Federated Learning with Cooperating Devices: A Consensus Approach for Massive IoT Networks," *IEEE Internet of Things J.*, vol. 7, no. 5, May 2020, pp. 4641–54.
[7] A. H. Sayed *et al.*, "Diffusion Strategies for Adaptation and Learning over Networks: An Examination of Distributed Strategies and Network Behavior," *IEEE Signal Processing Mag.*, vol. 30, no. 3, 2013, pp. 155–71.
[8] A. Elgabli *et al.*, "GADMM: Fast and Communication Efficient Framework for Distributed Machine Learning," *J. Machine Learning Research*, vol. 21, 2020, pp. 1–390.
[9] H. Xing *et al.*, "Decentralized Federated Learning via SGD over Wireless D2D Networks," *Proc. IEEE 21st Int'l. Wksp. Signal Processing Advances in Wireless Commun.*, May 2020.
[10] S. Wang *et al.*, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE JSAC*, vol. 37, no. 6, June 2019, pp. 1205–21.
[11] H. Kim *et al.*, "Blockchained On-Device Federated Learning," *IEEE Commun. Letters*, vol. 24, no. 6, June 2020, pp. 1279–83.
[12] M. Chen *et al.*, "Performance Optimization of Federated Learning over Wireless Networks," *Proc. IEEE GLOBECOM*, Waikoloa, HI, 2019, pp. 1–6.
[13] R. Nassif *et al.*, "Multitask Learning Over Graphs: An Approach for Distributed, Streaming Machine Learning," *IEEE Signal Processing Mag.*, vol. 37, no. 3, May 2020, pp. 14–25.
[14] M. Brambilla *et al.*, "Augmenting Vehicle Localization by Cooperative Sensing of the Driving Environment: Insight on Data Association in Urban Traffic Scenarios," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 4, Apr. 2020, pp. 1646–63.
[15] S. Kianoush, *et al.* "A Multisensory Edge-Cloud Platform for Opportunistic Radio Sensing in Cobot Environments," *IEEE Internet of Things J.*, vol. 8, no. 2, Jan. 2021, pp. 1154–68.

Biographies

STEFANO SAVAZZI is a researcher at the Consiglio Nazionale delle Ricerche (CNR) with the Institute of Electronics, Computer and Telecommunication Engineering (IEIIT). He was a visiting researcher at Uppsala University (2005) and the University of California, San Diego (2009). He has co-authored over 100 scientific publications. Research interests include signal processing, learning, and networking design aspects for the Internet of Things and beyond 5G. He was the recipient of the 2008 Dimitris N. Chorafas Foundation Award.

MONICA NICOLI is an associate professor at Politecnico di Milano, with the Department of Management, Economics and Industrial Engineering. She was a visiting researcher at ENI-Agip, Italy, in 1998–1999 and at Uppsala University, Sweden, in 2001. She has co-authored over 100 scientific publications. Her research interests are in the area of statistical signal processing, with focus on wireless communications, localization, and smart mobility. She was a recipient of the 1999 Marisa Bellisario Award and a co-recipient of the 2014 IET Intelligent Transport Systems and the 2018 IEEE Statistical Signal Processing Workshop Best Paper Awards.

MEHDI BENNIS is an associate professor at the Centre for Wireless Communications, University of Oulu, Finland, an Academy of Finland Research Fellow, and head of the intelligent connectivity and networks/systems group (ICON). His main research interests are in radio resource management, heterogeneous networks, game theory, and machine learning in 5G networks and beyond. He was the recipient of several prestigious awards, including the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, the 2016 Best Tutorial Prize from the IEEE Communications Society, the 2017 EURASIP Best Paper Award for the *Journal of Wireless Communications and Networks*, the all-University of Oulu award for research, and the 2019 IEEE ComSoc Radio Communications Committee Early Achievement Award.

SANAZ KIANOUSH has been a postdoctoral researcher at IEIIT of CNR since November 2014. She was a visiting researcher at Aalto University in 2018 and a lecturer at Azad University Sary, Iran, in 2008. Her research interests include statistical signal processing, machine learning in communication systems, and device-free radio localization.

LUCA BARBIERI is currently a Ph.D. candidate in information technology in the DEIB department, Politecnico di Milano. His current research interests focus on machine learning and localization techniques for vehicular and industrial networks.

> The choice between opportunistic and cooperative behaviors will largely depend on the URLLC design. Emerging FL tools are promising in cooperative automated driving, leveraging V2X interactions, and in collaborative robotics for distributed learning in complex and dynamic workflows.