A framework for energy and carbon footprint analysis of distributed and federated edge learning

Stefano Savazzi, Sanaz Kianoush, Vittorio Rampa
Consiglio Nazionale delle Ricerche (CNR)
IEIIT institute, Milano
Email: {name.surname}@ieiit.cnr.it

Mehdi Bennis
Centre for Wireless Communications
University of Oulu, Finland
Email: mehdi.bennis@oulu.fi

Abstract—Recent advances in distributed learning raise environmental concerns due to the large energy needed to train and move data from data centers. Novel paradigms, such as federated learning (FL), are suitable for decentralized model training across devices or silos that simultaneously act as both data producers and learners. Unlike centralized learning (CL) techniques, relying on big-data fusion and analytics located in energy-hungry data centers, in FL scenarios devices collaboratively train their models without sharing their private data. This article breaks down and analyzes the main factors that influence the environmental footprint of FL policies compared with classical CL/Big-Data algorithms running in data centers. The proposed analytical framework takes into account both learning and communication energy costs, as well as the carbon equivalent emissions; in addition, it models both vanilla and decentralized FL policies driven by consensus. The framework is evaluated in an industrial setting assuming a real-world robotized workplace. Results show that FL allows remarkable end-to-end energy savings (30% to 40%) for wireless systems characterized by low bit/Joule efficiency (50 kbit/Joule or lower). Consensus-driven FL does not require the parameter server and further reduces emissions in mesh networks (200 kbit/Joule). On the other hand, all FL policies are slower to converge when local data are unevenly distributed (often 2x slower than CL). Energy footprint and learning loss can be traded off to optimize efficiency.

I. INTRODUCTION

Recent advances in machine learning (ML) have revolutionized many domains and industrial scenarios. However, such improvements have been achieved at the cost of large computational and communication resources, resulting in significant energy and CO2 (carbon) footprints. Traditional centralized learning (CL) requires all training procedures to be conducted inside data centers [1] that are in charge of collecting training data from data producers (e.g., sensors, machines and personal devices), fusing large datasets, and continuously learning from them [2]. Data centers are thus energy-hungry and responsible for significant carbon emissions that amount to about 15% of the global emissions of the entire Information and Communication Technology (ICT) ecosystem [3].

An emerging alternative to centralized architectures is federated learning (FL) [4, 5]. Under FL, ML model parameters, e.g., weights and biases \( W \) of Deep Neural Networks (DNN), are collectively optimized across several resource-constrained edge/fog devices, that act as both data producers and local learners. FL distributes the computing task across many devices characterized by low-power consumption profiles, compared with data centers, and owning small datasets [2].

As shown in Fig. 1 using FL policies, such as federated averaging [5], allows devices to learn a local model under the orchestration of a centralized parameter server (PS). The PS fuses the received local models to obtain a global model that is fed back to the devices. PS functions are substantially less energy-hungry compared to CL and can be implemented at the network edge. This suggests that FL could bring significant reduction in the energy footprints, as the consumption is distributed across devices obviating the need for a large infrastructure for cooling or power delivery. However, vanilla FL architectures still leverage the server-client architecture which not only represents a single-point of failure, but also lacks scalability and, if not optimized, can further increase the energy footprint. To tackle these drawbacks, recent developments in FL architectures target fully decentralized solutions relying solely on in-network processing, thus replacing PS functions with a consensus-based federation model. In consensus-based
FL (CFL), the participating devices mutually exchange their local ML model parameters, possibly via mesh, or device-to-device (D2D) communication links [2], and implement distributed weighted averaging [6, 7, 8]. Devices might be either co-located in the same geographic area or distributed.

Contributions: the paper develops a novel framework for the analysis of energy and carbon footprints in distributed ML, including, for the first time, comparisons and trade-off considerations about vanilla FL, consensus-based (CFL) and data center based centralized learning. Despite an initial attempt to assess the carbon footprint for FL [3], the problem of quantifying an end-to-end analysis of the energy footprint still remains unexplored. To fill this void, we develop an end-to-end framework and validate it using real world data.

The paper is organized as follows: Sections II and III describe the framework for energy consumption and carbon footprint evaluation of different FL strategies, and the impact of energy efficiency in terms of communication and computing costs. In Section IV we consider a case study in a real-world industrial workplace targeting the learning of a ML model to localize human operators in a human-robot cooperative manufacturing plant. Carbon emissions are quantified and discussed in continuous industrial workflow applications requiring periodic model training updates.

II. ENERGY FOOTPRINT MODELING FRAMEWORK

The proposed framework provides insights into how different components of the FL architecture, i.e., the local learners, the core network and the PS, contribute to the energy bill. The learning system consists of $K$ devices and one data center ($k = 0$). Each device $k > 0$ has a dataset $\mathcal{E}_k$ of (labeled) examples $(x_h, y_h)$ that are typically collected independently. The objective of the learning system is to train a DNN model $\hat{y}(W; x)$ that transforms the input data $x$ into the desired outputs $\hat{y} \in \{y_h\}_C^{\mathcal{E}_k}$, where $C$ is the number of the output classes. Model parameters are specified by the matrix $W$ [3]. The training system uses the examples in $\bigcup_{k=1}^{K} \mathcal{E}_k$ to minimize the loss function $\xi(x_h, y_h|W)$ iteratively, over a pre-defined number $n$ of learning rounds.

Considering a device $k$, the total amount of energy consumed by the learning process can be broken down into computing and communication components. The energy cost is thus modelled as a function of the energy $E_k^{(C)}$ due to computing per learning round, and the energy $E_{k,0}^{(T)}$ per correctly received/transmitted bit over the wireless link $(k, h)$. In particular, the latter can be further broken down into uplink (UL) communication ($E_{0,k}^{(T)}$) with the data center (or the PS), and downlink (DL) communication ($E_{n,k}^{(T)}$), from the PS to the device. The energy cost for communication includes the power dissipated in the RF front-end, in the conversion, baseband processing and transceiver stages. We neglect the cost of on-off radio switching. In addition, communication energy costs are quantified on average, as routing through the radio access and the core network can vary (but might be assumed as stationary apart from failures or replacements). Finally, the energy $E_k^{(C)}$ for computing includes the cost of the learning round, namely the local gradient-based optimizer and data transfer.

energy $E_k^{(C)}$ depends on the GPU/CPU power consumption $P_0$ [3], the time span $T_0$ required for processing an individual batch of data, i.e., minimizing the loss $\xi(x_h|W)$, and the number $B$ of batches per round. We neglect here the cost of initial dataset loading since it is a one-step process. For $n = n(\xi)$ rounds, and a target loss $\xi$, the total, end-to-end, energy in Joule [J] is given by:

$$E_{CL}(\xi) = \gamma \cdot n \cdot E_0^{(C)} + \sum_{k=1}^{K} b(\mathcal{E}_k) \cdot E_{k,0}^{(T)},$$

where $\gamma$ is the Power Usage Effectiveness (PUE) of the considered data center [10], [11]. The cost for UL communication for data fusion, $\sum_{k=1}^{K} b(\mathcal{E}_k) \cdot E_{0,k}^{(T)}$, scales with the data size $b(\mathcal{E}_k)$ of the $k$-th local database $\mathcal{E}_k$ and the number of devices $K$. PUE $\gamma > 1$ accounts for the additional power consumed by the data center infrastructure for data storage, power delivery and cooling; values are typically $\gamma = 1.1 \div 1.8$ [11].

B. Federated Learning (FL)

Unlike CL, FL distributes the learning process across a selected subset $N_t$ of $K_a < K$ active devices as shown in Fig. 1. At each round $t$, the local dataset $\mathcal{E}_k$ is used to train a local model $W_{k,t}$, in order to minimize the local loss $\xi_k$ as $W_{k,t} = \text{argmin}_W \xi_k(W)$. The local model is then

| Parameters | Data center/PS ($k = 0$) | Devices ($k \geq 1$) |
|-----------|--------------------------|----------------------|
| Comp. $P_k$: | 140 W(CPU) + 42 W(GPU) | 5.1 W (CPU) |
| Batch time $T_k$: | 20 ms | 190 ms |
| Batches $B$: | 3 | 3 |
| Raw data size: | $K \cdot b(\xi_k)$ MB | $b(\xi_k) \simeq 30$ MB |
| Model size: | $b(W) = 290$ KB | $b(W) = 290$ KB |
| PUE $\gamma$: | 1.67 | 1 |
| Utilization $\beta$: | 0.1 (model averaging) |
| ML model: | DeepMind [2], 5 layers, $C = 6$. Optimizer: Adam |

### Table I

| Comp. EE: | $EE_{ID} = 0.02 \div 1$ Mb/J | $EE_{DL} = 0.02 \div 1$ Mb/J |
|-----------|----------------|----------------|
| Downlink (DL): | Mesh or D2D (M): | Uplink (UL): |
| EE: | $EE_{ID} = 0.02 \div 1$ Mb/J | $EE_{DL} = 0.02 \div 1$ Mb/J |
| Comp. EE: | $EE_{ID} = 0.02 \div 1$ Mb/J | $EE_{DL} = 0.02 \div 1$ Mb/J |
| Utilization $\beta$: | 0.1 (model averaging) |
| ML model: | DeepMind [2], 5 layers, $C = 6$. Optimizer: Adam |
C. Consensus-driven Federated Learning (CFL)

As shown in [3], the PUE for all devices is set to control resource allocation and frame aggregation operations. Data regularly in small batches simplifies medium access lossy/lossless compression [12][13] is implemented. Sending the parameters size is roughly the same for each device, unless compared with the raw data [5]:

\[
\text{compared with the raw data [5]:}
\]

\[
\text{compared with the raw data [5]:}
\]

while the communication interface is powered on to decode and PS consumption, namely:

\[
E_{\text{PS}} = \gamma \cdot \beta \cdot \sum_{k=1}^{K} \left( E_{0,k}^{C} + a \cdot \sum_{k=1}^{K} \left( E_{0,k}^{T} + b(W) \cdot E_{k,t}^{(T)} \right) \right).
\]

PS energy is given by \( E_{0,k}^{C} \) and depends on the time, \( \beta T_0 \), needed for model averaging. The new model is sent to the PS [5] over the UL. The PS is in charge of updating the global model \( W_{t+1} \) for the following round \( t + 1 \) through the aggregation of the \( K_a \) received models [4]:

\[
W_{t+1} = \frac{1}{K_a} \sum_{k \in N_t} \Gamma_k \cdot W_{k,t}, \quad \text{with } \Gamma_k = \frac{Q_k}{Q} \quad \text{and } (Q_k, Q) \text{ being the number of local and global examples, respectively.}
\]

The new model \( W_{t+1} \) is finally sent back to the devices over the DL. Other strategies are discussed in [5]. Notice that, while \( K_a \) active devices run the local optimizer and share the local model with the PS on the assigned round, the remaining \( K - K_a \) devices have their computing interface turned off, while the communication interface is powered on to decode the updated global model.

For \( n \) rounds, now consisting of learning and communication tasks, the total end-to-end energy includes both devices and PS consumption, namely:

\[
E_{\text{FL}}(\xi) = \gamma \cdot n \cdot \beta \cdot E_{0}^{(C)} + \frac{1}{K_a} \sum_{k=1}^{K} \left( E_{0,k}^{(C)} + a \cdot \sum_{k=1}^{K} \left( E_{0,k}^{(T)} + b(W) \cdot E_{k,t}^{(T)} \right) \right).
\]

The sum \( \sum_{k \in N_t} b(W) \cdot E_{k,t}^{(T)} \) models the total energy spent by the device \( k \) to diffuse the local model parameters to \( N \) selected neighbors at round \( t \).

### III. Carbon Footprint Assessment

The carbon footprint evaluation assumes that each device \( k \), including the server, is located in a specific geographical region characterized by a known carbon intensity (\( CI_k \)) of electricity generation [15]. CI is measured in kg CO2-equivalent emissions per kWh (kgCO2eq/kWh) which quantifies how much carbon emissions are produced per kilowatt hour of generated electricity. In the following, we consider the CI figures reported in EU back in 2019 [16]. Considering the energy models [1]-[3], carbon emission is evaluated by multiplying each individual energy contribution, namely \( E_{0,k}^{C} \) and \( E_{k,t}^{(C)} \) by the corresponding intensity values \( CI_k \). Carbon footprints and the proposed framework are summarized in Table II for CL (\( C_{CL} \)) and FL policies (\( C_{FL} \) and \( C_{CFL} \)).

To analyze the main factors that impact the estimated carbon emissions, a few simplifications to the energy models [1]-[3] are introduced in the following. Communication and computing costs are quantified on average, in terms of the corresponding energy efficiencies (EE). Communication EE for DL (\( EE_{\text{D}} = [E_{0,k}^{(T)}]^{-1} \)), UL (\( EE_{\text{U}} = [E_{k,t}^{(T)}]^{-1} \)) and mesh networking (\( EE_{\text{M}} = [E_{k,t}^{(T)}]^{-1} \)) are measured in bit/Joule [bit/J] and describe how much energy is consumed per correctly received information bit [17]. Efficiencies depend on device/server consumption for communication \( P_T \) and net
UL/DL or mesh throughput $R$. Depending on network implementations, we consider different choices of EE_D, EE_U and EE_M. The computing efficiency, $EE_C = \left[ E_o^{(C)} \right]^{-1}$, quantifies the number of rounds per Joule [round/J], namely how much energy per learning round is consumed at the data center (or PS). Devices equipped with embedded low-consumption CPUs typically experience a larger time span $T_h > T_0$ to process an individual batch of data; on the other hand, they use much lower power ($P_k$). Device computing EE is typically larger and modeled here as $EE_C = \varphi = E_o^{(C)} / E_i^{(C)} < 1$. Typical values for communication and computing EE are in Table 1.

In the proposed FL implementation, the set of $K_a$ active FL devices changes according to a round robin scheduling, other options are proposed in [19]. Considering typical FL implementations, such as gossip [6], we let the devices choose up to $N = 1$ neighbors per round. When ad-hoc mesh, or D2D, communication interfaces are not available, the energy cost to implement the generic peer-to-peer link $(k, h)$ roughly corresponds to an UL transmission from the source $k$ to the core network access point (i.e., router), followed by a DL communication from the router$(s)$ to the destination device $h$, namely $E_{k, h}^{(T)} \approx E_{k, 0}^{(T)} + E_{0, h}^{(T)}$, or equivalently $[EE_M]^{-1} \approx [EE_D]^{-1} + [EE_U]^{-1}$. Router can be a host or base-station. In mesh networks, further optimization via power control [18] may be also possible depending on the node deployment. Since devices do not need the router to relay information to the PS, which may be located in a different country, substantial energy savings are expected.

IV. INDUSTRY 4.0 ROBOTIZED ENVIRONMENT

According to [20], in 2019 industry was responsible for about 30% of the world greenhouse gas emissions. To counter this impact, Industry 4.0 (I4.0) and other mitigation policies have been recently introduced [21]. In line with the I4.0 paradigm, we resort to a common Industrial Internet of Things (IoT) scenario where AI-based sensors and machines are interconnected and co-located in the same plant [22]. These sensors interact within an industrial workspace where human workers are co-present. Devices are served by a WiFi (IEEE 802.11ac) network and a router ($P_t = 6$ W [23]) is in charge of orchestrating the mesh communication or forwarding to the data center, or PS.

A. Case study: scenario-dependent setup

The goal of the training task is to learn a ML model for the detection (classification) of the position of the human operators sharing the workspace, namely the human-robot distance $d$ and the direction of arrival (DOA) $\theta$. Further details about the robotic manipulators, the industrial environment and the deployed sensors are given in [2, 18]. Input data $x_n$, available online [24], are range-azimuth maps obtained from 3 time-division multiple-input-multiple output (TD-MIMO) frequency modulated continuous wave (FMCW) radars working in the 77 GHz band [24]. During the on-line workflow, position $(d, \theta)$ information are obtained from the trained ML model and sent to a programmable logic controller for robot safety control (e.g., emergency stop or replanning tasks). The ML model adopted for the classification of the operator location is a simplified version of the DeepMind [2]. It consists of 5 trainable layers and 3M parameters, of which 170k are compressed, encoded by 16 bits and exchanged during FL. Model outputs are reduced to $C = 6$ for the detection of 6 subject locations around the robot, detailed in [24]. Batch times and size of exchanged model parameters $b(W)$ (kB) are

![Diagram](image-url)
reported in Table I. Adam optimizer is used with a Huber loss [5]. The number of devices (K) is in the range 30 ≤ K ≤ 60, data can be identically distributed (IID) or non-IID. Moreover, 20 ≤ Kα ≤ 40 and N = 1 are assumed.

Energy and carbon footprints are influenced by data center and device hardware configurations. The data center hardware consumption is reported in Table I and uses CPU (Intel i7 8700K, 3.7 GHz, 64 GB) and GPU (Nvidia Geforce GTX 1060, 1.5 GHz, 3 GB). For FL devices, we use Raspberry Pi 4 boards based on a low-power CPU (ARM-Cortex-A72 SoC type BCM2711, 1.5 GHz, 8 GB). These devices can be viewed as a realistic pool of FL learners embedded in various IIoT applications. FL is implemented using Tensorflow v2.3 backend (sample code available also in [24]). In what follows, rather than choosing a specific communication protocol, we follow a what-if analysis approach, and thus we quantify the estimated carbon emissions under the assumption of different DL/UL communication efficiencies (EE). Since actual emissions may be larger than the estimated ones depending on the specific protocol overhead and implementation, we will highlight relative comparisons.

B. Case study: carbon footprint analysis

Fig. 2 provides an estimate of the carbon footprint under varying settings as detailed in Table I. Fig. 2(a) shows the carbon footprint for varying number of learning rounds (n), comparing CL with K = 60 devices and FL with Kα = 40. For CL (dashed line), an initial energy cost shall be paid for UL raw data transmission, which depends on the data size b(Ei) and the communication EE; in this example, EEU = EED = 200 kbit/J. Next, the energy cost is only due to computing (40 J/round), unless new labelled data are produced by devices before the learning process ends on the data center. In contrast to CL, FL footprint depends on communication and computing energy costs per round. CFL (cross markers) has a cost of 224 J/round, smaller than FL, namely 287 J/round (circle markers) as PS is not required. Notice that mesh communication is replaced by UL and DL WiFi transmissions to/from a router.

Energy and accuracy loss ξ can be traded off to optimize efficiency. For example, CL needs n = 25 rounds at the data center to achieve a loss of ξ = 0.08 and a carbon footprint of 2.9 gCO2-eq. Model training should be typically repeated every 3 hours to track modifications of the robotic cell layout, which corresponds to a total carbon emission of 8.4 equivalent kgCO2-eq per year. CFL trains for more rounds (here n = 27) to achieve a slightly larger loss (ξ = 0.2), but reduces the emissions down to 1.7 gCO2-eq, or 4.9 kgCO2-eq per year, if training is repeated every 3 hours. Finally, FL achieves a similar footprint, however this comes in exchange for a larger validation loss (ξ = 0.3) due to communication with the PS. Although not considered here, tuning of model as well as changing the aggregation strategy at the PS [5] would reduce the training time and thus emissions.

The end-to-end energy cost is investigated in Figs. 2(b) and 2(c). Energy vs. loss trade-off is first analyzed in Fig. 2(b). We consider 3 setups where the data center and the devices are placed in different geographical areas featuring different carbon indexes (CI). In particular, the first scenario (max EU, red) is characterized by devices located in a region that produces considerable emissions as CI = 0.97 kgCO2-eq/kWh. This corresponds to the max emission rate in EU [16]. In the second (IT, black) and third (FI, blue) scenarios, devices and data center are located in Italy, CI = 0.28 kgCO2-eq/kWh, and Finland, CI = 0.11 kgCO2-eq/kWh, respectively. When the availability of green energy is small (i.e., max EU scenario, CI = 0.97), the learning loss and accuracy must be traded with carbon emissions. For example, for an amount of gas emission equal, or lower, than CL, the learning loss of CFL should be increased to ξ = 0.1, corresponding to an average accuracy of 90%. Considering FL, this should be increased to ξ = 0.25. For smaller carbon indexes, i.e. IT and FI scenarios, the cost per round reduces. Therefore, FL can train for all the required rounds and experience the same loss as in CL with considerable emission savings (30% + 40% for Finland). A promising roadmap for FL optimization is to let local learners contribute to the training process if, or when, green energy, namely small CI, is made available.

In Fig 2(c) we now quantify the carbon emissions of CL, FL and CFL for varying communication EE, ranging from EEU = EED = 50 kbit/J to 400 kbit/J, and number of devices, K = 30 (Kα = 20), and K = 60 (Kα = 40). An increase of the network size or a decrease of the network kb/J efficiency cause

| TABLE III |
| -------------------------------------------- |
| NUMBER OF ROUNDS (MIN-MAX), COMMUNICATION/COMPUTING ENERGY COSTS AND CORRESPONDING CARBON FOOTPRINTS FOR SELECTED CASES, VARYING LOSSES ξ AND IID VS. NON-IID DATA DISTRIBUTIONS. EEU = EED = 100 kbit/J |
|                                | Comm. Energy (kJ) | Comp. Energy (kJ) | Footprint (g-CO2-eq) |
|---------------------------------|------------------|--------------------|----------------------|
|                                | ξ = 0.1 | ξ = 0.2 | ξ = 0.1 | ξ = 0.2 | ξ = 0.1 | ξ = 0.2 | ξ = 0.1 | ξ = 0.2 |
| CL K = 60, Kα = 40 | 14.2 | 16.1 | 16.1 | 18.1 | 17.0 | 19.0 | 17.0 | 19.0 |
| CFL non-IID K = 60, Kα = 40 | 14.2 | 16.1 | 16.1 | 18.1 | 17.0 | 19.0 | 17.0 | 19.0 |
| CFL IID K = 60, Kα = 40 | 14.2 | 16.1 | 16.1 | 18.1 | 17.0 | 19.0 | 17.0 | 19.0 |
| FL non-IID K = 60, Kα = 40 | 14.2 | 16.1 | 16.1 | 18.1 | 17.0 | 19.0 | 17.0 | 19.0 |
| FL IID K = 60, Kα = 40 | 14.2 | 16.1 | 16.1 | 18.1 | 17.0 | 19.0 | 17.0 | 19.0 |
communication to emit much more CO2 than training. Since FL is more communication efficient as (compressed) model parameters are exchanged, in line with [3], the best operational condition of FL is under limited communication EE regimes. For the considered scenario, the optimal EE below which FL leaves a smaller carbon footprint than CL is in the range 50%±100 kbit/J for FL (ξ = 0.2) and 250% ±300 kbit/J for CFL (ξ = 0.1). Finally, notice that for all cases FL can efficiently operate under EE = 50 kbit/J, typically observed in low power communications [23], and 4G/5G NB-IoT [26].

Table [11] compares the energy and carbon footprints for IID and non-IID data distributions. Computing, communication energy costs and corresponding carbon emissions for different target losses are evaluated with respect to the max EU scenario. Considering FL and CFL, federated computations are now distributed across $K_3$ devices, therefore larger computing costs are needed. Non-IID data generally penalizes both FL and CFL as energy consumption increases up to 40% in some cases. For example, while CFL with IID data limits the number of required epochs (targeting $\xi = 0.1$) to a maximum of $n = 43$, it is less effective for non-IID distributions as the required rounds now increase up to $n = 62$ for some devices. CFL and FL thus experience an increase in energy costs, but CFL still emits lower carbon emissions. More advanced gradient-based CFL methods [17] might be considered when data distributions across devices are extremely unbalanced.

V. Conclusions

This work developed a framework for the analysis of energy and carbon footprints in distributed and federated learning (FL). It provides, for the first time, a trade-off analysis between vanilla and consensus CL on local datasets, and centralized learning inside the data center. A simulation framework has been developed for the performance analysis over arbitrarily complex wireless network structures. Carbon equivalent emissions are quantified and discussed for a continual industrial workflow monitoring application that tracks the movements of workers inside human-robot shared workspaces. The ML model is periodically (re)trained to track changes in data distributions. In many cases, energy and accuracy should be traded to optimize FL energy efficiency. Furthermore, by eliminating the parameter server, as made possible by emerging decentralized FL architectures, further reducing the energy footprint is a viable solution. Novel opportunities for energy-aware optimizations are also highlighted. These will target the migration of on-device computations where the availability of green energy is larger. Finally, FL requires a frequent and intensive use of the communication interfaces. This mandates a co-design of the federation policy and the communication architecture, rooted in the novel 6G paradigms.

REFERENCES

[1] M. Dayarathna, et al., “Data Center Energy Consumption Modeling: A Survey,” IEEE Communications Surveys & Tutorials, vol. 18, no. 1, pp. 732-794, First quarter 2016.

[2] S. Savazzi et al., “Opportunities of Federated Learning in Connected, Cooperative and Automated Industrial Systems,” IEEE Communications Magazine, vol. 52, no. 2, February 2021. [Online]. Available: https://arxiv.org/abs/2101.03367.

[3] X. Qiu, et al., “Can Federated Learning Save the Planet?” NeurIPS - Tackling Climate Change with Machine Learning, Dec. 2020, Vancouver, Canada. [Online]. Available: https://arxiv.org/abs/2010.06537.

[4] J. Konečný, et al., “Federated optimization: Distributed machine learning for on-device intelligence,” CoRR, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527.

[5] P. Kairouz, et al., “Advances and open problems in federated learning,” [Online]. Available: https://arxiv.org/abs/1912.04977.

[6] M. Blot, et al., “Gossip training for deep learning,” 50th Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 2016. [Online]. Available: https://arxiv.org/abs/1611.09726.

[7] S. Savazzi, et al., “Federated Learning with Cooperating Devices: A Consensus Approach for Massive IoT Networks,” IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4641-4654, May 2020.

[8] Z. Chen, et al., “Consensus-Based Distributed Computation of Link-Based Network Metrics,” IEEE Signal Processing Letters, vol. 28, pp. 249-253, 2021.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, et al., “Human-level control through deep reinforcement learning.” Nature 518, 529-533, 2015.

[10] A. Capozzoli, et al., “Cooling systems in data centers: state of art and emerging technologies,” Energy Procedia, vol. 83, pp. 484-493, 2015.

[11] H. Xing, et al., “Decentralized Federated Learning via SGD over Wireless D2D Networks,” Proc. IEEE 21st Int. Workshop on Signal Processing Advances in Wireless Comm. (SPAWC), Atlanta, GA, USA, pp. 1-5, 2020.

[12] S. Shlezinger, et al., “Federated Learning with Quantization Constraints,” Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, pp. 8851-8855, 2020.

[13] A. Elgabli, et al., “GADMM: Fast and Communication Efficient Framework for Distributed Machine Learning,” Journal of Machine Learning Research, vol. 21, no. 76, pp. 1-39, 2020.

[14] L.F.W. Anthony, et al., “Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models,” Proc of ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, 2020.

[15] European Environment Agency, Data and maps: “Greenhouse gas emission intensity of electricity generation,” Dec. 2020. [Online]. Available: https://tinyurl.com/56isV5ht.

[16] E. Björnson and E. G. Larsson, “How Energy-Efficient Can a Wireless Communication System Become?” Proc. 52nd Asilomar Conf. on Sig., Syst., and Comp., Pacific Grove, CA, USA, 2018, pp. 1252–1256.

[17] S. Savazzi, et al., “A Joint Decentralized Federated Learning and Communications Framework for Industrial Networks,” Proc. of IEEE CAMAD, Pisa, Italy, pp. 1–7, 2020.

[18] M. M. Amir, et al., “Federated learning over wireless fading channels,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3546–3557, May 2020.

[19] J. G. J. Olivier, et al., “Trend in global CO2 and total greenhouse gas emissions,” 2020 Report, PBL Netherland Environmental Assessment Agency, Dec. 2020. [Online]. Available: https://tinyurl.com/zzb7j67.

[20] H. Fekete, et al. “A review of successful climate change mitigation policies in major emitting economies and the potential of global replication,” Renewable and Sustainable Energy Reviews, vol. 137, art. 110602, 2021.

[21] S. Kianoush, et al., “A Multisensory Edge-Cloud Platform for Opportunistic Radio Sensing in Cobot Environments,” IEEE Internet of Things Journal, vol. 8, no. 2, pp. 1154–1168, 2021.

[22] The Power Consumption Database: WiFi routers. [Online]. Available: http://www.wpcdb.com [Accessed: 08/03/2021].

[23] Dataset: “Federated Learning: mmWave MIMO radar dataset for testing,” IEEE Dataport, 2020. [Online]. Available: http://dx.doi.org/10.21227/0wmc-hq36. Accessed: March. 2021.

[24] X. Vilajosa, et al., “IEIE 6TISCH: A Tutorial,” IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 595–615, Firstquarter 2020.

[25] S. Zhang, et al., “Energy efficiency for NPUSCH in NB-IoT with guard band,” ZTE Commun., vol. 16, no. 4, pp. 46–51, Dec. 2018.