Compressed Bayesian Federated Learning for Reliable Passive Radio Sensing in Industrial IoT

Luca Barbieri[1,2], Stefano Savazzi[2], Monica Nicoli[1]
[1] *Politecnico di Milano, Milan, Italy*, [2] *Consiglio Nazionale delle Ricerche, Milan, Italy*

***Abstract*—Bayesian Federated Learning (FL) has been recently introduced to provide well-calibrated Machine Learning (ML) models quantifying the uncertainty of their predictions. Despite their advantages compared to frequentist FL setups, Bayesian FL tools implemented over decentralized networks are subject to high communication costs due to the iterated exchange of local posterior distributions among cooperating devices. Therefore, this paper proposes a communication-efficient decentralized Bayesian FL policy to reduce the communication overhead without sacrificing final learning accuracy and calibration. The proposed method integrates compression policies and allows devices to perform multiple optimization steps before sending the local posterior distributions. We integrate the developed tool in an Industrial Internet of Things (IIoT) use case where collaborating nodes equipped with autonomous radar sensors are tasked to reliably localize human operators in a workplace shared with robots. Numerical results show that the developed approach obtains highly accurate yet well-calibrated ML models compatible with the ones provided by conventional (uncompressed) Bayesian FL tools while substantially decreasing the communication overhead (i.e., up to 99%). Furthermore, the proposed approach is advantageous when compared with state-of-the-art compressed frequentist FL setups in terms of calibration, especially when the statistical distribution of the testing dataset changes.**

***Index Terms*—Federated Learning, Bayesian deep learning, Compression, Decentralized networks, Consensus**

I. Introduction

Nowadays, Federated Learning (FL) is used to obtain high-quality Machine Learning (ML) models that are trained from decentralized data sources without disclosing any private information [1]–[4]. Given the privacy-preserving features of FL, several industrial applications have integrated it to provide enhanced learning functionalities compared to conventional data center-based training strategies. Examples of such applications include Industrial Internet of Things (IIoT) services [5], [6], autonomous driving use cases [7], [8], and healthcare [9], [10]. Still, most of the studied FL implementations focus on standard (frequentist) strategies, where the goal is to find a single (optimized) set of ML model parameters that best fit the training data. Following such a strategy has been shown to produce models that output overconfident predictions regardless of their correctness, especially when the local datasets of the devices are limited in size [11], [12]. This raises major safety concerns for industrial services as downstream tasks may rely on the overconfident and often incorrect output provided by the Neural Networks (NNs). To overcome this limitation, Bayesian FL tools [13] target the learning of the posterior distribution in the model parameter space. By doing so, a reliable uncertainty measure can be obtained and used to make more informed decisions, consequently improving safety.

Bayesian FL tools are typically implemented considering approximate methods as obtaining the full posterior is often intractable [11]. The first class of techniques relies on Variational Inference (VI) methodologies [14], [15] where a surrogate distribution is learned in place of the true posterior, while the second uses Markov Chain Monte Carlo (MCMC) approaches [16], [17] that approximate the posterior density via random samples. Focusing on MCMC approaches, some distributed implementations have been developed for centralized [17], [18] or fully distributed network topologies [12], [16], while assuming uncompressed communications. More advanced designs have been introduced recently to reduce the communication footprint in Bayesian FL setups. For example, in [19], a communication-efficient Bayesian FL policy is developed where devices compress their local gradients and apply variance reduction techniques to improve learning performances. Similarly, authors in [20] propose several compression operators for Langevin-based FL strategies that rely on primal, dual, and bidirectional compression. Other approaches instead quantize the gradients exchanged during the learning process with 1-bit compression [21] or propose new Lagevin schemes that support compressed communications [22]. Despite this recent progress, all the aforementioned techniques focus on centralized strategies while completely overlooking decentralized Bayesian FL tools.

Contributions. The paper proposes a novel communication-efficient Bayesian FL strategy suitable for fully decentralized industrial setups, referred to as Compressed Decentralized Bayesian Federated Learning (CD-BFL). In industrial contexts where safety is paramount, Bayesian FL tools are needed to obtain reliable ML models that support trustworthy predictions. Compared to previous works that rely on centralized architectures, this is the first work that develops a fully decentralized method integrating compressed communications among devices. The proposed tool draws inspiration from [23] and extends it to support Bayesian FL strategies based on Langevin dynamics approaches. Specifically, CD-BFL integrates compression policies and allows devices to perform multiple optimization steps before sending updates to their neighbors, massively reducing the communication cost. Numerical results focus on an IIoT learning problem where devices aim at

This paper is partially funded by the EU in the call HORIZON-HLTH-2022-STAYHLTH-01-two-stage under grant agreement No 101080564 and by the project HORIZON-EIC-2022-PATHFINDEROPEN-01 under grant Agreement No 101099491.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00071

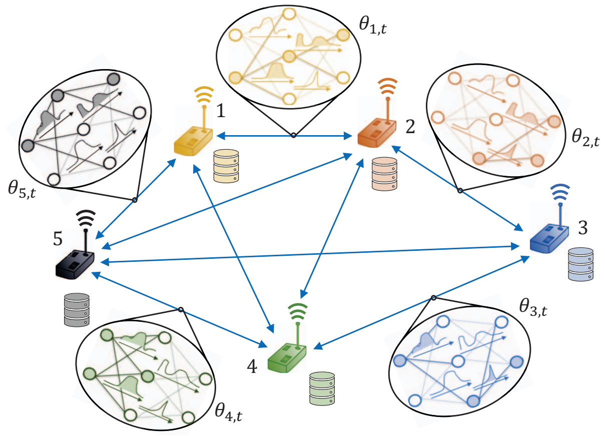


Fig. 1. Bayesian FL setup: collaborating nodes ($N = 5$) iteratively exchange compressed samples via D2D communications and perform local computations to obtain a close approximation of the global posterior distribution.

collaboratively localizing human operators inside a human-robot shared workspace. They show that CD-BFL substantially reduces the communication overhead (i.e., roughly by 99%) compared to a conventional uncompressed decentralized Bayesian FL strategy without impacting final learning performances or calibration. The proposed strategy is also advantageous at reliably quantifying the uncertainty when the statistical distribution of the testing data changes, while a state-of-the-art (compressed) frequentist FL policy fails at producing well-calibrated models.

The paper is organized as follows. Sec. II introduces the system model and reviews Langevin-based sampling algorithms and their federated variants, while Sec. III presents the developed decentralized Bayesian FL policy. Sec. IV introduces the industrial case study used for evaluating the performances of CD-BFL (Sec. V). Finally, Sec. VI draws some conclusions.

II. System Model and Bayesian Learning Preliminaries

We focus on the decentralized FL setup sketched in Fig. 1 where a set of devices $\mathcal{K} = \{1, \ldots, K\}$ collaborate for solving a supervised learning problem. Devices are connected according to the undirected graph $\mathcal{G} = (\mathcal{K}, \mathcal{E})$, with $\mathcal{E}$ being the set of directed edges. The set of neighbors of device k including k is denoted with $\mathcal{N}_k$, while the same set that excludes k is indicated with $\mathcal{N}_{\overline{k}}$. Each device $k \in \mathcal{K}$ is able to share information with its neighbors via D2D communications and holds a local dataset $\mathcal{D}_k = \{(\mathbf{x}_h, y_h)\}_{h=1}^{E_k}$ of E_k examples pairs $(\mathbf{x}_h, y_h)$, with $\mathbf{x}_h$ and y_h being the input data and desired prediction, respectively. The global dataset is denoted with $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^{N}$ and comprises $E = \sum_{k=1}^{K} E_k$ training examples. As commonly observed in FL setups, the local datasets may vary in size across different learners or under-represent the global dataset as only a limited number of classes is available at each device. The devices' goal is to obtain a set of global model parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ approximately sampled from the true global posterior $p(\boldsymbol{\theta}|\mathcal{D})$ of the model $\boldsymbol{\theta}$ which takes into account all the data available at the learners. In what follows, the proposed Bayesian learning tools rely on gradient-based MCMC approaches. Specifically, we review Stochastic Gradient Langevin Dynamics (SGLD), and then discuss its extension to fully decentralized networks.

A. Centralized Stochastic Gradient Langevin Dynamics

SGLD [24] aims at learning the posterior distribution in the model parameter space. To do so, each device shares its local dataset with a centralized unit in charge of carrying out the learning process. Specifically, the goal is to obtain a close approximation of the global posterior distribution

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{k=1}^{K} p(\mathcal{D}_k|\boldsymbol{\theta}) \,, \tag{1}$$

where $p(\mathcal{D}_k|\boldsymbol{\theta}) = \prod_{h=1}^{E_k} p(y_h|\mathbf{x}_h, \boldsymbol{\theta})$ denotes the likelihood function describing the shared ML model adopted by the learners, while $p(\boldsymbol{\theta})$ is the prior distribution. Starting from the initial samples $\boldsymbol{\theta}_0 \in \mathbb{R}^p$ at $t = 0$, SGLD produces new samples iteratively by adding Gaussian noise $\boldsymbol{\xi}_t$ to standard gradient descent updates as [24]

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla f(\boldsymbol{\theta}_t) + \sqrt{2\eta}\boldsymbol{\xi}_{t+1} \,, \tag{2}$$

where $t = 0, \ldots, T$ denotes the iteration number, $\boldsymbol{\xi}_{t+1}$ is a random vector drawn from an independent and identically (i.i.d.) Gaussian distribution $\mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$, η is the learning rate, while $f(\boldsymbol{\theta}_t) = \sum_{k=1}^{N} f_k(\boldsymbol{\theta}_t, \mathcal{M}_k)$ with

$$f_k(\boldsymbol{\theta}_t, \mathcal{M}_k) = -\log p(\mathcal{M}_k|\boldsymbol{\theta}_t) - \frac{1}{N} \log p(\boldsymbol{\theta}_t) \,, \tag{3}$$

where $p(\mathcal{M}_k|\boldsymbol{\theta}_t)$ is the likelihood evaluated over a mini-batch $\mathcal{M}_k = \{(\mathbf{x}_b, y_b)\}_{m=1}^{M}$ comprising M training examples. In practice, a burn-in phase is typically utilized in gradient-based MCMC methods where the first T_b samples are discarded while the remaining $T - T_b$ ones are used for uncertainty quantification.

B. Decentralized Stochastic Gradient Langevin Dynamics

SGLD is not directly applicable to FL setups as all the data collected by the learners need to be sent to a centralized location, thereby raising privacy concerns. To overcome this limitation, Decentralized Stochastic Gradient Langevin Dynamics (DSGLD) [16] has been introduced, which enables to implement SGLD over decentralized wireless networks without sharing any privacy-sensitive data. Under DSGLD, each device $k \in \mathcal{K}$ updates its local samples as [16]

$$\boldsymbol{\theta}_{k.t+1} = \sum_{j \in \mathcal{N}_k} \omega_{kj} \boldsymbol{\theta}_{j,t} - \eta \nabla f_k(\boldsymbol{\theta}_{k,t}, \mathcal{M}_k) + \sqrt{2\eta}\boldsymbol{\xi}_{k,t} \,, \tag{4}$$

where $\omega_{k,j}$ is the (k, j)-th entry of a symmetric, doubly stochastic $K \times K$ matrix $\boldsymbol{\Omega}$ which can be chosen according to the choices presented in [25]. Still, DSGLD entails excessive communication overhead to exchange the samples among neighboring devices. To address this shortcoming, we propose a communication-efficient DSGLD implementation, as detailed in the next section.

Algorithm 1 CD-BFL

Input: initial samples $\boldsymbol{\theta}_{k,0}\, \forall k \in \mathcal{N}$, graph $\mathcal{G}$, matrix $\boldsymbol{\Omega}$, mixing weight ζ, initialize $\mathbf{v}_{k,0} = \mathbf{0}_p$ and $\bar{\mathbf{v}}_{j,0} = \mathbf{0}_p\ \forall j \in \mathcal{N}$

```
for each round $t = 0, 1, \ldots T$ do
    for each learner $k \in \mathcal{N}$ do
        $\boldsymbol{\theta}_{k,t}^{(0)} \leftarrow \boldsymbol{\theta}_{k,t}$
        for each $\ell = 1, \ldots, L$ do
            $\boldsymbol{\theta}_{k,t}^{(\ell)} = \boldsymbol{\theta}_{k,t}^{(\ell-1)} - \eta \nabla f_k\left(\boldsymbol{\theta}_{k,t}^{(\ell-1)}, \mathcal{M}_k^{(\ell)}\right)$
        end for
        $\Delta\boldsymbol{\theta}_{k,t} = \mathcal{Q}\left(\boldsymbol{\theta}_{k,t}^{(L)} - \mathbf{v}_{k,t}\right)$
        send $(\Delta\boldsymbol{\theta}_{k,t})$ and receive $\{\Delta\boldsymbol{\theta}_{j,t}\}_{j\in\mathcal{N}_k^-}$
        $\mathbf{v}_{k,t+1} = \mathbf{v}_{k,t} + \Delta\boldsymbol{\theta}_{k,t}$
        $\bar{\mathbf{v}}_{k,t+1} = \bar{\mathbf{v}}_{k,t} + \sum_{j\in\mathcal{N}_i} \omega_{kj}\, \Delta\boldsymbol{\theta}_{j,t}$
        $\boldsymbol{\theta}_{k,t+1} = \boldsymbol{\theta}_{k,t}^{(L)} + \zeta(\bar{\mathbf{v}}_{k,t+1} - \mathbf{v}_{k,t+1}) + \sqrt{2\eta}\xi_{k,t+1}$
    end for
end for
```

III. Compressed Bayesian FL strategy

This section presents the proposed communication-efficient decentralized Bayesian FL strategy, namely CD-BFL, which is summarized in Algorithm 1. CD-BFL draws inspiration from [24] and introduces compression strategies to reduce the communication overhead required to implement DSGLD over wireless networks. Besides, it allows devices to perform multiple gradient descent steps before each communication phase, improving reliability as shown in Sec. V. The proposed scenario is critical in industrial setups, such as the one discussed in Sec. IV, where networked devices have limited resources.

Let us denote the current model iterate $\boldsymbol{\theta}_{k,t}$ available at device k at iteration t, each device updates it recursively using stochastic gradient descent for L local steps. Specifically, starting from $\boldsymbol{\theta}_{k,t}^{(\ell)} = \boldsymbol{\theta}_{k,t}$ when $\ell = 0$, each step $\ell \geq 1$ with $\ell = 1, \ldots, L$ is evaluated as

$$\boldsymbol{\theta}_{k,t}^{(\ell)} = \boldsymbol{\theta}_{k,t}^{(\ell-1)} - \eta \nabla f_k\left(\boldsymbol{\theta}_{k,t}^{(\ell-1)}, \mathcal{M}_k\right) \tag{5}$$

where $\mathcal{M}_k$ is a mini-batch containing M training examples (drawn randomly from the local dataset $\mathcal{D}_k$) while η is the same learning rate defined in (2). Then, instead of exchanging $\boldsymbol{\theta}_{k,t}$ as it is, the devices use an additional control sequence $\mathbf{v}_{k,t}$, with $\mathbf{v}_{k,0} = \mathbf{0}_p$ for $t = 0$, to preserve the average of the local iterates over consecutive iterations while also enforcing the noise introduced by the compression stage to vanish for a sufficiently high number of iterations t [23]. Given the current control sequence $\mathbf{v}_{k,t}$, each device computes the following (compressed) quantity

$$\Delta\boldsymbol{\theta}_{k,t} = \mathcal{Q}\left(\boldsymbol{\theta}_{k,t}^{(L)} - \mathbf{v}_{k,t}\right) \tag{6}$$

where $\mathcal{Q}(.)$ denotes a compression operator (see e.g., [26] or [27] for widely-used compression policies). The compressed representation $\Delta\boldsymbol{\theta}_{k,t}$ is then exchanged over the D2D links.

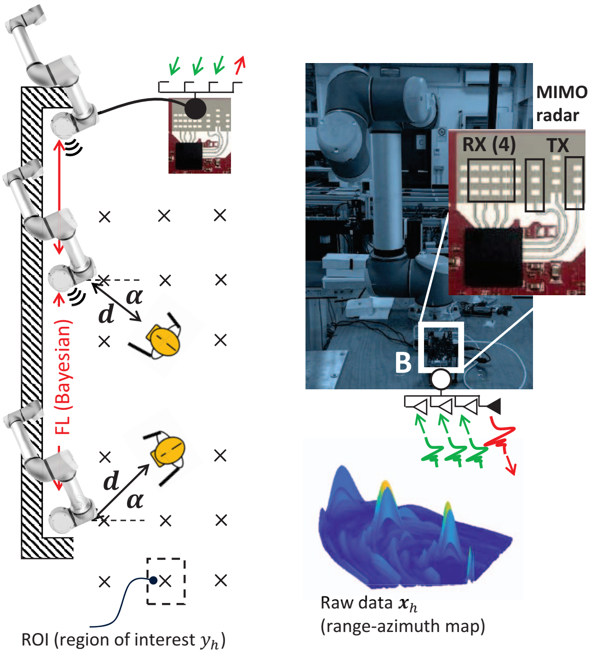

MIMO radar
RX (4)
TX
B
FL (Bayesian)
d
α
d
α
Raw data $\boldsymbol{x}_h$ (range-azimuth map)
ROI (region of interest y_h)

Fig. 2. Human-Robot-Cooperative workspace scenario and FL setup.

Devices receiving the compressed samples from their neighbors first update their control sequences as

$$\mathbf{v}_{k,t+1} = \mathbf{v}_{k,t} + \Delta\boldsymbol{\theta}_{k,t}. \tag{7}$$

Next, they update an additional variable $\bar{\mathbf{v}}_{k,t}$, with $\bar{\mathbf{v}}_{k,0} = \mathbf{0}_p$ for $t = 0$, that stores the compressed representation of the control sequences received from their neighbors as

$$\bar{\mathbf{v}}_{k,t+1} = \bar{\mathbf{v}}_{k,t} + \sum_{j\in\mathcal{N}_k} \omega_{kj}\Delta\boldsymbol{\theta}_{j,t} \tag{8}$$

where ω_{kj} is again the (k,j)-th entry of a symmetric, doubly stochastic matrix $\boldsymbol{\Omega}$. Finally, devices update their local model iterates via a consensus-based aggregation strategy while also adding the Gaussian noise as in standard DSGLD

$$\boldsymbol{\theta}_{k,t+1} = \boldsymbol{\theta}_{k,t}^{(L)} + \zeta(\bar{\mathbf{v}}_{k,t+1} - \mathbf{v}_{k,t+1}) + \sqrt{2\eta}\boldsymbol{\xi}_{k,t+1} \tag{9}$$

where $0 < \zeta \leq 1$ is a mixing parameter that can be optimized to improve learning performance or calibration. As in gradient-based MCMC methods, the first T_b samples $\{\boldsymbol{\theta}_{k,t}\}_{t=0}^{T_b}$ are discarded while the remaining $T - T_b$ ones, namely $\{\boldsymbol{\theta}_{k,t}\}_{t=T_b+1}^{T}$, are used for evaluating the performances.

IV. Case study: HRC workspace

This section discusses the proposed Bayesian federation model CD-BFL based on experimental data. As depicted in Fig. 2, the proposed FL scenario resorts to a network of radar devices [28] each equipped with a Time-Division Multiple-Input-Multiple-Output (TD-MIMO) Frequency Modulated Continuous Wave (FMCW) radar working in the 77–81 GHz band [29]. The above-cited devices are employed to

TABLE I
LABELS (HR COLLABORATIVE SITUATIONS) FOR $R = 10$ ROIS.

Labels	ROI: range d	ROI: DOA α
$y_h = 0$	$d \geq 2$m	$-60 \leq \alpha \leq 60$ deg
$y_h = 1$	0.5m $\leq d \leq 0.7$m	$40 \leq \alpha \leq 60$ deg
$y_h = 2$	0.3m $\leq d \leq 0.5$m	$-10 \leq \alpha \leq 10$ deg
$y_h = 3$	0.5m $\leq d \leq 0.7$m	$-60 \leq \alpha \leq -40$ deg
$y_h = 4$	1m $\leq d \leq 1.2$m	$20 \leq \alpha \leq 40$ deg
$y_h = 5$	0.9m $\leq d \leq 1.1$m	$-10 \leq \alpha \leq 10$ deg
$y_h = 6$	1m $\leq d \leq 1.2$m	$-40 \leq \alpha \leq -20$ deg
$y_h = 7$	1.2m $\leq d \leq 1.6$m	$10 \leq \alpha \leq 20$ deg
$y_h = 8$	1.1m $\leq d \leq 1.5$m	$-5 \leq \alpha \leq 5$ deg
$y_h = 9$	1.2m $\leq d \leq 1.6$m	$-20 \leq \alpha \leq -10$ deg

monitor a shared industrial workspace during Human-Robot Collaboration (HRC) tasks to detect and track the position of the human operators (the range distance d and the Direction Of Arrival (DOA) α) relative to a robotic manipulator inside a fenceless space [30]. In industrial shared workplaces, measuring positions and distance is mandatory to enforce a worker protection policy and to implement collision avoidance [31].

Each radar features an array of 3 TX and 4 uniformly spaced RX antennas, with an azimuth Field Of View (FOV) of -/+ 60°, with angle and range resolution of $25°$ and 4.2cm, and 3.9 GHz band (sweep time 36μs). Radars autonomously compute the beat signals on each receiving antenna as the result of the radar echoes reflected by moving objects. Beat signals are then converted in frequency domain via Fast Fourier Transform (FFT) to obtain range-azimuth map measurements $\mathbf{x}_h$. Maps $\mathbf{x}_h$ serve as training data and have size 256 x 63 samples. Radars use a trained ML model to obtain the relative position (d, α) information. In addition, the subject position can be sent to a programmable logic controller for robot safety control, for emergency stop or replanning tasks.

The ML model is here trained to detect the human subject in $R = 10$ Region Of Interests (ROIs), namely potential collaborative situations characterized by different human-robot (HR) distances or DOAs. In particular, the ROI with label $y_h = 0$ corresponds to the robot and the corresponding human worker cooperating at a safe distance (distance $\geq$ 2 m), the remaining 9 labels ($y_h > 0$) identify the human operator as working close by the robot, at variable distances and DOAs as depicted in Table I. The range-azimuth measurements $\mathbf{x}_h$ are collected independently by the individual devices using a dedicated radar DSP (C674x). The corresponding labels y_h are associated manually during training stages and stored locally. Federated model training is then implemented to replace data fusion and using an ARM-Cortex-A57 SoC (Jetson Nano device model). Notice that the radars collect a large amount of data, that cannot be shipped back to the server for training and inference, due to the latency constraints imposed by the localization service and safety policies [30].

The CD-BFL tool has been simulated on a virtual environment which creates an arbitrary number of virtual radar devices configured to process an assigned batch of range-azimuth data $\mathbf{x}_h$ and exchanging compressed samples as detailed in Sec. III. The model adopted for localization is based on a Lenet architecture [32] with $p = 2.7$ million trainable parameters and is also trained continuously to track the variations of data dynamics caused by changes in the workflow processes (typically, on a daily basis). The initial training of the ML model is obtained at day $i = 1$ using a large dataset $\mathcal{D}^{(1)} = \{(\mathbf{x}_h, y_h)\}_{h=1}^{E^{(1)}}$ of raw range-azimuth data manually labeled. Possible re-training stages occur daily, $i > 1$ and are based on new data. Datasets for initial model training and two subsequent re-training stages, $\mathcal{D}^{(i)}$ with $i = 2, 3$, i.e., two consecutive days, are available online in [33].

V. NUMERICAL RESULTS

In this section, we evaluate CD-BFL over a network of $K = 10$ radars each having 50 independently and identically distributed (iid) range-azimuth maps taken from the dataset $\mathcal{D}^{(1)}$ [33]. Connectivity among radars is simulated and assumes that each radar communicates with all $K-1$ neighbors.

As performance metrics, we utilize the standard measure of validation accuracy and the Expected Calibration Error (ECE) [34], which quantifies the ML model's reliability. Specifically, the ECE measures the disagreement between the accuracy and the confidence scores provided by the NNs and is computed as follows. Given the predictions of the ML model over the validation dataset, they are divided into a set of O non-overlapping bins $\{B_o\}_{o=1}^{O}$ according to their associated confidence scores. These scores are computed by taking the maximum probability assigned by the softmax operation at the last layer of the NN. Then, we compute the average accuracy $\text{acc}(B_o)$ and average confidence $\text{conf}(B_o)$ for each bin separately. Finally, the ECE is evaluated by taking into account all bins as [34]

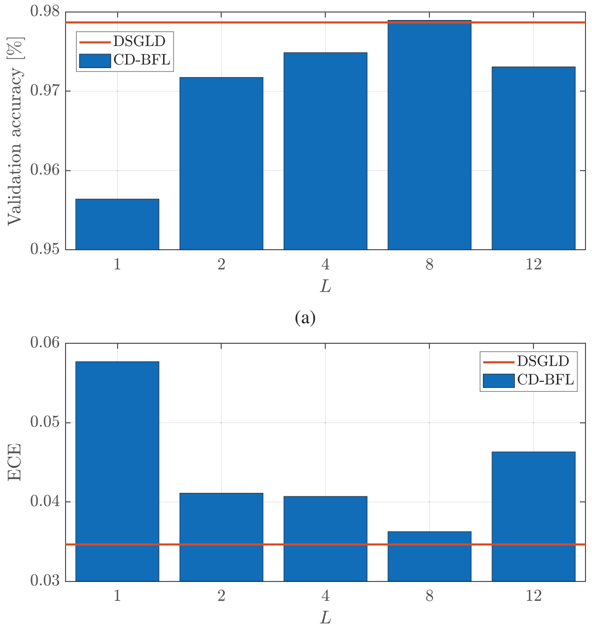
$$\text{ECE} = \sum_{o=1}^{O} \frac{|B_o|}{\sum_{o'=1}^{O} |B_{o'}|} |\text{acc}(B_o) - \text{conf}(B_o)| \,, \tag{10}$$

where $|B_o|$ denotes the number of validation samples falling inside the o-th bin.

The experiments compare the performances of CD-BFL with the ones attained by an uncompressed Bayesian FL method, namely DSGLD, as well as the state-of-the-art frequentist FL approach developed in [23], which we refer to as Compressed Frequentist Federated Learning (CF-FL). All methods use a learning rate of $\eta = 10^{-4}$ and are trained for $T = 800$ iterations. For the Bayesian FL setups, we consider a standard Gaussian as prior, i.e., $p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}_p, \mathbf{I}_p)$ and a burn-in phase comprising the first $T_b = 700$ iterations. Top-k sparsification [27] is used as compression operator by CF-FL and CD-BFL and is configured such that only 1% of the model parameters is exchanged by the devices, allowing to reduce the communication overhead by 99%. Finally, all methods set the entries of $\boldsymbol{\Omega}$ as in [35] and we select $\zeta = 0.03$.

A. Results

The first set of results evaluates the performances of CD-BFL for varying number of local optimization steps L, ranging from 1 up to 12, and compares it with the ones attained by DSGLD. Fig. 3 reports the validation accuracy (Fig. 3a)



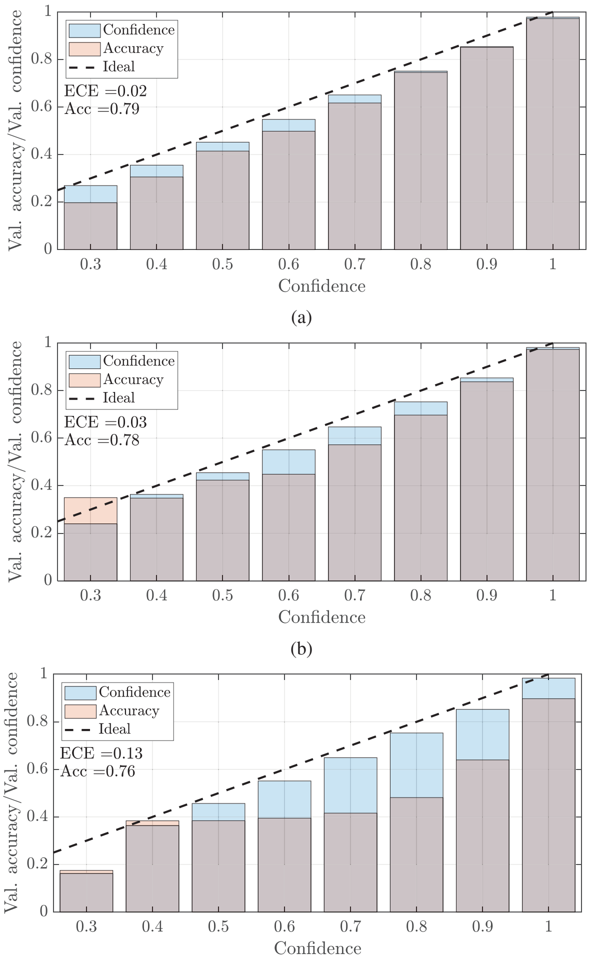
(a)

(b)

Fig. 3. Comparison between DSGLD and CD-BFL for varying values of L in terms of validation accuracy (a) and ECE (b).

and the ECE (Fig. 3b) achieved by CD-BFL and DSGLD. Results show a trade-off between the number of local gradient descent updates L and the final learning performances. Indeed, increasing L generally leads to better accuracy and ECE provided $L \leq 8$, while for $L > 8$ the models produced by the proposed strategy tend to be poorly calibrated and with limited generalization ability. This should be expected as too many local optimization steps may introduce overfitting with a consequent reduction of the final accuracy and an increase in the ECE. Therefore, in the considered setting, $L = 8$ should be chosen to maximize the prediction abilities and the reliability of the NNs. By doing so, CD-BFL attains nearly the same accuracy as DSGLD at the cost of a slightly higher ECE.

The second set of results aims at characterizing the ability of CD-BFL to provide reliable models when the data collected by the radar devices have different statistical distributions (e.g., due to different radar configurations and/or slight changes in the HRC workspace). To do so, we resort to the same dataset $\mathcal{D}^{(1)}$ in [33] where we train CD-BFL with $L = 8$, as suggested by the previous analysis, DSGLD, and CF-FL and we evaluate the learned models on the testing datasets of all other days. In particular, the testing datasets are manipulated to contain only samples with labels $1 \leq y_h \leq 6$ as considered highly critical due to the short distance (i.e., $d < 1.5$ m) between the robot and the human operator. Indeed, in those positions, reliable ML models must be obtained so that robot control strategies can confidently use the NN predictions to avoid injuries. In what follows, the results shown are averaged over the testing datasets of days $i = 2, 3$.



(a)

(b)

(c)

Fig. 4. Reliability plots attained by (a) DSGLD, (b) CD-BFL and (c) CF-FL.

Fig. 4 reports the reliability diagrams [34] which provide a visual representation of the reliability of the ML models trained under DSGLD (Fig. 4a), CD-BFL (Fig. 4b) and CF-FL (Fig. 4c) strategies. Comparing the results, even though CF-FL and CD-BFL provide the same savings in terms of communication overhead, they attain rather different reliability diagrams. Indeed, CF-FL provides overconfident models since the confidence scores shown in Fig. 4c are much higher than the accuracy for most bins. This poses major safety concerns as downstream tasks, e.g., robot control strategies, may rely on the (overconfident) predictions of the NN, likely giving rise to injuries. Besides, CF-FL is shown to be the least accurate when compared with DSGLD or CD-BFL. On the other hand, the reliability diagram of CD-BFL shows that the confidence scores closely follow the actual accuracy. Therefore, the proposed strategy can be confidently used under safety-critical conditions as it provides well-calibrated ML models able to reliably quantify the uncertainty of their predictions. Still, some performance loss in terms of ECE and accuracy are exhibited by CD-BFL compared to DSGLD, albeit this difference is very limited.

VI. Conclusions

This paper proposed a communication-efficient decentralized Bayesian FL policy, referred to as CD-BFL, for reliable passive localization in IIoT setups. The developed method introduces compression operators while allowing devices participating in the learning process to perform multiple local optimization steps. The goal is to reduce communication overhead while maximizing the reliability of the ML models. The proposed tool is integrated within a cooperative passive localization task where networked radars aim to accurately detect and classify human motions inside an industrial human-robot collaborative workplace. Numerical results show that CD-BFL can be confidently utilized under safety-critical industrial operations as its ML models reliably quantify the uncertainty of their predictions. Specifically, CD-BFL provides accurate prediction capabilities as well as reliable uncertainty quantification that are in line with conventional uncompressed decentralized Bayesian FL setups but at a much lower (i.e., 99%) communication overhead. Moreover, the models produced by CD-BFL are well-calibrated even when they are used under different testing conditions, while state-of-the-art compressed decentralized FL strategies fail at producing reliable ML models.

Future research works will target the theoretical characterization of the developed CD-BFL tool and the development of strategies that jointly optimize the communication and the computational phases of the cooperative Bayesian FL strategy.

References

[1] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[2] T. Li *et al.*, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[3] S. Savazzi *et al.*, "Opportunities of federated learning in connected, cooperative, and automated industrial systems," *IEEE Communications Magazine*, vol. 59, no. 2, pp. 16–21, 2021.

[4] L. Barbieri, S. Savazzi, S. Kianoush, M. Nicoli, and L. Serio, "A carbon tracking model for federated learning: Impact of quantization and sparsification," in *2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2023, pp. 213–218.

[5] D. C. Nguyen *et al.*, "Federated learning for industrial internet of things in future industries," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 192–199, 2021.

[6] P. Boobalan *et al.*, "Fusion of federated learning and industrial internet of things: A survey," *Computer Networks*, vol. 212, p. 109048, 2022.

[7] Z. Du *et al.*, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2020.

[8] L. Barbieri *et al.*, "A layer selection optimizer for communication-efficient decentralized federated deep learning," *IEEE Access*, vol. 11, pp. 22 155–22 173, 2023.

[9] D. C. Nguyen *et al.*, "Federated learning for smart healthcare: A survey," *ACM Computing Surveys*, vol. 55, no. 3, feb 2022.

[10] B. Camajori Tedeschini *et al.*, "Decentralized federated learning for healthcare networks: A case study on tumor segmentation," *IEEE Access*, vol. 10, pp. 8693–8708, 2022.

[11] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[12] L. Barbieri *et al.*, "Channel-driven decentralized Bayesian federated learning for trustworthy decision making in D2D networks," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[13] L. Cao *et al.*, "Bayesian federated learning: A survey," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 8 2023, pp. 7233–7242.

[14] M. Ashman *et al.*, "Partitioned Variational Inference: A Framework for Probabilistic Federated Learning," *arXiv e-prints*, p. arXiv:2202.12275, Feb. 2022.

[15] R. Kassab *et al.*, "Federated generalized Bayesian learning via distributed Stein variational gradient descent," *IEEE Transactions on Signal Processing*, vol. 70, pp. 2180–2192, 2022.

[16] M. Garbazbalaban *et al.*, "Decentralized stochastic gradient Langevin dynamics and Hamiltonian monte carlo," *Journal of Machine Learning Research*, vol. 22, no. 239, pp. 1–69, 2021.

[17] S. Ahn *et al.*, "Distributed stochastic gradient MCMC," in *Proceedings of the 31st International Conference on Machine Learning*, 22–24 Jun 2014, pp. 1044–1052.

[18] D. Liu *et al.*, "Wireless federated langevin monte carlo: Repurposing channel noise for bayesian sampling and privacy," *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 2946–2961, 2023.

[19] M. Vono *et al.*, "QLSD: Quantised langevin stochastic dynamics for bayesian federated learning," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, vol. 151, 28–30 Mar 2022, pp. 6459–6500.

[20] A. Karagulyan *et al.*, "ELF: Federated langevin algorithms with primal, dual and bidirectional compression," in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. [Online]. Available: https://openreview.net/forum?id=wxZ3G1LdJj

[21] S. Lee *et al.*, "Bayesian Federated Learning over Wireless Networks," *arXiv e-prints*, p. arXiv:2012.15486, Dec. 2020.

[22] L. Sun *et al.*, "Federated Learning with a Sampling Algorithm under Isoperimetry," *arXiv e-prints*, p. arXiv:2206.00920, Jun. 2022.

[23] A. Koloskova *et al.*, "Decentralized stochastic optimization and gossip algorithms with compressed communication," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 3478–3487.

[24] M. Welling *et al.*, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 681–688.

[25] L. Xiao *et al.*, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[26] D. Alistarh *et al.*, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] S. Shi *et al.*, "Understanding top-k sparsification in distributed deep learning," *arXiv e-prints*, 2019.

[28] S. Savazzi *et al.*, "An energy and carbon footprint analysis of distributed and federated learning," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 1, pp. 248–264, 2023.

[29] B. Vandersmissen *et al.*, "Indoor person identification using a low-power fmcw radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 7, pp. 3941–3952, 2018.

[30] N. Pedrocchi *et al.*, "Safe human-robot cooperation in an industrial environment," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 27, 2013. [Online]. Available: https://doi.org/10.5772/53939

[31] 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.

[32] Y. Lecun *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] S. Savazzi, "Federated learning: mmwave mimo radar dataset for testing," 2020. [Online]. Available: https://dx.doi.org/10.21227/0wmc-hq36

[34] C. Guo *et al.*, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1321–1330.

[35] H. Xing *et al.*, "Federated learning over wireless device-to-device networks: Algorithms and convergence analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3723–3741, 2021.