Federated Learning for Indoor Localization via Model Reliability With Dropout

Junha Park, Jiseon Moon, Taekyoon Kim, Peng Wu, Tales Imbiriba, Pau Closas, *Senior Member, IEEE*, and Sunwoo Kim, *Senior Member, IEEE*

***Abstract*— In this letter, we propose a novel model weight update method that accounts for the reliability of the local clients in FL-based indoor localization. FL shows degraded localization performance than centralized learning because of the non-independent and identically distributed (non-IID) data configuration. Thus, we aim to improve the localization performance by applying the reliability of the local clients, which is quantified by the model uncertainty of the local models. Bayesian models provide a framework for capturing model uncertainty but usually requires a substantial computational cost as well, particularly for high-dimensional learning problems. In order to resolve this computational issue, the proposed scheme applies Monte Carlo (MC) dropout to approximate the Bayesian uncertainty quantification with enhanced computational efficiency. Our simulation results show that the proposed learning method improves localization performance compared to the existing model, federated averaging (FedAvg), and close to the centralized learning performance.**

***Index Terms*— Federated learning (FL), indoor localization, model uncertainty, Bayesian approximation.**

I. Introduction

FINGERPRINTING localization is a conventional indoor localization scheme for estimating a location using a wireless signal feature (such as its received signal strength, RSS) [1], [2]. In the context of RF fingerprinting, neural network (NN) models have been widely considered to deal with large-scale datasets. One of the challenges is to acquire the fingerprinting database, which can be a time-consuming and costly process. In parallel, as the size of the dataset increases, the localization system typically demands a more complex model with increased computational burden [3], [4]. In order to address this problem, researchers have focused on developing parallel computation techniques with divided datasets and crowd-based solutions [5].

As an alternative, federated learning (FL) has recently emerged as distributed learning scheme, which addresses privacy aspects in distributed processing and power efficiency by potentially leveraging edge computing [6]. In the context of fingerprint-based localization and FL, deep models are employed to exploit the relevant features from the RF signal. The use of FL approaches to train NN models has been considered over non-independent and identically distributed (non-IID) datasets and unbalanced local datasets [7]. Thanks to the aforementioned advantages, researchers considered the adoption of FL into advanced wireless communications and indoor navigation [8]–[10].

As a consequence of each client collecting RSS fingerprints through their own devices, the resulting large-scale dataset features heterogeneous characteristics caused by a large number of clients and limited communication environments. Such client heterogeneity characteristics might lead to performance degradation if not properly addressed, with the current work addressing this problem through client selection approaches [11]. The limitation of other solutions to this problem is that those do not consider the heterogeneous characteristics and that the collected dataset might not be employed due to the exclusion of some clients, ultimately resulting in unexplored areas as well. In our approach, we aggregate the local model weights with the reliability of the local client through a FL scheme, where such reliability is quantified by the uncertainty of the local model in predicting the measurements collected by the corresponding client [12].

In this letter, we propose the first FL-based indoor localization technique that considers the model uncertainty as a measure of reliability. The reliability of the local client is reciprocal with the model uncertainty, which is related to the variance of the prediction error. The computation of such model uncertainty can be obtained by using Bayesian deep learning models, which turns to be impractical in some situations due to its computational burden, thus requiring substantial run-time and computational complexity. We verify the proposed approach through simulation experiments using real-world WiFi RSS fingerprint dataset [13]. Furthermore, we present improvements in localization error of the proposed method compared to existing methods.

The main contributions of this letter are the following: 1) the first FL-based indoor localization system with consideration of the uncertainty of the local client, 2) the reliability of the local client is quantified by the uncertainty of the local model, which is implemented by using MC dropout to mitigate the computation of implementing a Bayesian NN, and 3) the performance of the proposed FL system is investigated through experiments with a real-world dataset.

Manuscript received 3 March 2022; revised 10 April 2022; accepted 21 April 2022. Date of publication 28 April 2022; date of current version 12 July 2022. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(NFA) (No.2019-0-01325, Developed of wireless communication tracking based location information system in disaster scene for fire-fighters and person who requested rescue), and the NSF under Award ECCS-1845833. The associate editor coordinating the review of this letter and approving it for publication was Z. Yang. *(Corresponding author: Sunwoo Kim.)*

Junha Park, Jiseon Moon, Taekyoon Kim, and Sunwoo Kim are with the Department of Electronic Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: eric0725@hanyang.ac.kr; jiseonmoon@hanyang.ac.kr; kty0264@hanyang.ac.kr; remero@hanyang.ac.kr).

Peng Wu, Tales Imbiriba, and Pau Closas are with the Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115 USA (e-mail: wu.p@northeastern.edu; talesim@northeastern.edu; closas@northeastern.edu).

Digital Object Identifier 10.1109/LCOMM.2022.3170878

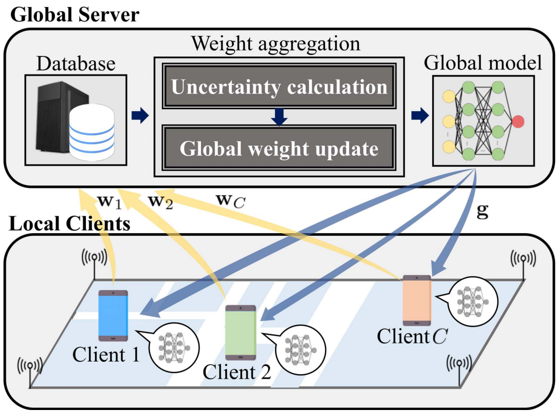

Fig. 1. The local model is trained with given train dataset. As a consequence by using MC dropout, distribution of the predicted output can be approximated.

II. Federated Learning for Indoor Localization

In this section we describe fingerprint data collection with crowdsourcing clients and then iterative model weight update in local clients and global server side for FL-based indoor localization system.

A. Data Collection

We assume that there are K fixed WiFi APs and C crowdsourcing clients during RSS fingerprint collection. The available localization area covered by the fingerprint localization system includes N reference locations $\mathcal{L} = \{\boldsymbol{\ell}_i | i = 1, \ldots, N\}$ where $\boldsymbol{\ell}_i$ is 2×1 dimensional vector composed of x and y coordinates of the i-th reference location. The c-th client collects the reference fingerprint $\mathbf{x}_{c,j} = [r^1_{c,j}, \ldots, r^K_{c,j}]^{\mathrm{T}} \in \mathbb{R}^{K \times 1}$ from the WiFi APs where $r^k_{c,j}$ is the RSS value collected by the c-th client from the k-th AP at the j-th reference location. Here, the c-th client visits a subset of the reference locations $\mathcal{L}_c = \{\mathbf{y}_{c,j} | j = 1, \ldots, N^r_c\}$, so that $\mathcal{L}_c \subseteq \mathcal{L}$ and $\mathcal{L} = \bigcup_{c=1}^{C} \mathcal{L}_c$, where N^r_c is the number of reference locations visited by the c-th clients [1]. The dataset of the c-th client $\mathcal{D}_c = \{\mathbf{X}_c, \mathbf{Y}_c\}$ includes the RSS fingerprint as input $\mathbf{X}_c = [\mathbf{x}_{c,1}, \ldots, \mathbf{x}_{c,N^r_c}] \in \mathbb{R}^{K \times N^r_c}$, and the location as output $\mathbf{Y}_c = [\mathbf{y}_{c,1}, \ldots, \mathbf{y}_{c,N^r_c}] \in \mathbb{R}^{2 \times N^r_c}$.

B. Iterative Model Weight Update

In the FL framework, there are two weight updates: local weight update and global model weight update. The c-th client trains its local model $\mathbf{w}_c$ using its own database $\mathcal{D}_c$. These trained local models are then send to the server, which assembles the various local model weights and updates the global model weight $\mathbf{g}$. Here, all clients without exclusion, send their model weights trained with a database collected from the localization areas they have visited. Other schemes where only a set of users transmit their weights or do it through some duty-cycling are possible, but these considerations are considered out of the scope of this letter which is to investigate the potential performance gains.

1) Local Client: The local client trains each model with the non-IID fingerprint and reference dataset. The training of the c-th local client is formulated by way of minimizing the local objective function of the c-th client $F(\mathbf{w}_c)$, which is presented as follows:

$$\min_{\mathbf{w}_c} F(\mathbf{w}_c), \quad F(\mathbf{w}_c) := \frac{1}{N^r_c} \sum_{j=1}^{N^r_c} f(\mathbf{w}_c; \mathbf{x}_{c,j}, \mathbf{y}_{c,j}), \tag{1}$$

where $\mathbf{w}_c$ is the local model weight of the c-th client and $f(\cdot)$ is the loss function for the local model [14]. The local model weights are updated repeatedly in the direction of decreasing the value of the local objective function, depending on the optimizer and the loss function designed by the localization system. After each local client completes its local training with several iterations, they transmit local model weights to the server.

2) Global Server: As the server receives the optimized local model weights $\widetilde{\mathbf{w}}_c$, the global model weight $\mathbf{g}$ is updated as follows

$$\mathbf{g} = \sum_{c=1}^{C} u_c \widetilde{\mathbf{w}}_c, \tag{2}$$

where u_c is the linear combination coefficient of the c-th client with condition of $u_c \geq 0$ and $\sum_{c=1}^{C} u_c = 1$. By repeating the rounds, the global model is updated with each of the local client's characteristics. In further sections, we describe the novel approach to setting the u_c term, a crucial parameter for the FL-based localization performance.

III. Uncertainty in Neural Networks

This section describes the computation of model uncertainty using MC dropout as an approximation for Bayesian NN model. Then we analyze the relationship between the variance of prediction error and the produced uncertainty.

A. Monte Carlo Dropout as a Bayesian approximation

Bayesian deep learning makes it possible to measure the uncertainty of a NN with a probabilistic approach [12]. The probability of the predicted output $\mathbf{y}^*$, given train dataset $\mathbf{X}, \mathbf{Y}$ and the new input $\mathbf{x}^*$, is characterized by

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \mathbf{Y}) \, d\mathbf{w}, \tag{3}$$

where $\mathbf{x}^*$ and $\mathbf{y}^*$ are the input and output that are not included in the training dataset $\mathbf{X}$, respectively. The probability $p(\mathbf{y}^* | \mathbf{x}^*, \mathbf{X}, \mathbf{Y})$ can be solved with variational inference and MC integration, as in the approach in [15] and the uncertainty can be quantified as the variance of the probability distribution.

Even if Bayesian deep learning can optimally measure the uncertainty in NN, it is difficult to apply in FL system because of substantial computational resources. MC dropout, which efficiently approximates Bayesian NNs in terms of computation, makes it possible to measure the uncertainty by using a simpler NN model [15]. Thus we apply the MC dropout method to the proposed FL system to calculate model uncertainty with lower computational cost. The dropout nodes in NN model are chosen over a Bernoulli distribution with the probability of p_d, where the dropout rate is $1 - p_d$. As shown in Fig. 1, MC dropout generates different predicted output from the local model. The randomness of dropout enables to produce a distribution for the predicted output.

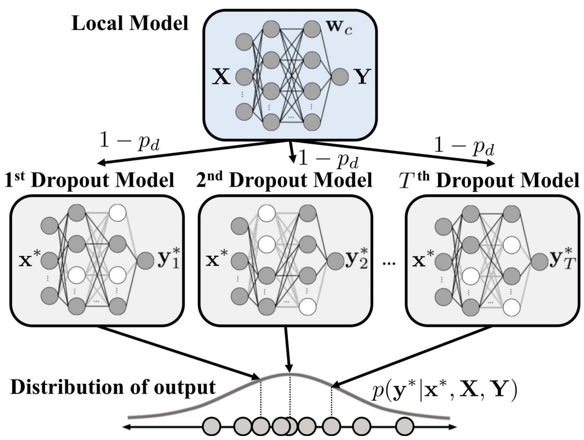

Fig. 2. An illustration of the federated learning based localization system with reliability.

B. Variance Decomposition of Uncertainty

The predicted output of the NN is affected by two factors: noise and model structure. Due to these two factors, uncertainty in NN is described by aleatoric uncertainty and epistemic uncertainty [16]. In deep learning, the observed output $\mathbf{y}$ is represented by a sum of a single noiseless ideal predictions $o(x)$ which is a deterministic value of input x and a random additive noise term distributed as $\mathcal{N}(0, \sigma_\alpha^2)$. $\mathbf{y}^*$ is the predicted output vector of NN model. Both output $\mathbf{y}$ and $\mathbf{y}^*$ are sampled from random variables Y and Y^*, respectively. Since the Y and Y^* are independent variables generated from different datasets, the variance of the prediction error can be decomposed and quantified with two uncertainties as follows

$$\begin{aligned}\mathrm{Var}\left(Y-Y^*\right) &= \mathbb{E}\left[\left(Y-o(x)+o(x)-Y^*\right)^2\right] \\ &= \mathbb{E}\left[\left(Y-o(x)\right)^2\right]+\mathbb{E}\left[\left(o(x)-Y^*\right)^2\right] \\ &= \sigma_\alpha^2+\sigma_e^2, \end{aligned} \tag{4}$$

where σ_α^2 and σ_e^2 represent aleatoric and epistemic uncertainty respectively. From this decomposition, we show that the variance of the predicted output can be decomposed by the summation of two uncertainties. The variance represents the reliability of each client and is used to update the global model in the FL localization system, as described next.

IV. FL Localization With Reliability

In this section, we describe the proposed algorithm which consists of system initialization, reliability calculation, and global model weight update.

A. System Initialization

In FL localization system, every round operates sequentially in local client-side and global server-side side until the global model weight converges. Fig. 2 shows that clients send local model weights trained with their own collected data and sever updates global model weights by the uncertainty of each client.

Before the FL training, the global model in the server initializes the global model weight $\mathbf{g}$. Also, the validation dataset $\mathcal{D}^v = \{\mathbf{X}^v, \mathbf{Y}^v\}$ is prepared in server database,

Algorithm 1 Updating Global Model Weight With Local Model Uncertainty

Input : Client index $c = 1, 2, \ldots, C$; Training data of the c-th local client $\mathcal{D}_c = \{\mathbf{X}_c, \mathbf{Y}_c\}$ where the inputs and outputs are RSS fingerprints and locations; Validation data $\mathcal{D}^v = \{\mathbf{X}^v, \mathbf{Y}^v\}$

Output: Optimized global model weight

```
Initialization: Initialize global model weight g^1
for γ = 1, 2, ... until convergence of global model do
    Initialize the local model weight w_c^{γ−1} with the global model weight g^γ
    for c = 1, 2, ..., C do
        Update the local model weight w_c^γ using D_c
        Compute the client uncertainty U_c^γ and reliability R_c^γ using (7)
    end
    Update the global model weight g^{γ+1} using (8)
end
```

which includes $\mathbf{X}^v = [\mathbf{x}_1^v, \ldots, \mathbf{x}_{N^v}^v] \in \mathbb{R}^{K\times N^v}$ and $\mathbf{Y}^v = [\mathbf{y}_1^v, \ldots, \mathbf{y}_{N^v}^v] \in \mathbb{R}^{2\times N^v}$ for the purpose of calculating local models uncertainty, where N^v is the number of the pair of fingerprint and location in $\mathcal{D}^v$. The i-th fingerprint in $\mathcal{D}^v$ is denoted by $\mathbf{x}_i^v = [r_i^{v,1}, \ldots, r_i^{v,K}]^{\mathrm{T}} \in \mathbb{R}^{K\times 1}$ and the i-th location in $\mathcal{D}^v$ is represented by $\mathbf{y}_i^v$, $i = 1, \ldots, N^v$.

B. Reliability Calculation With Monte Carlo Dropout

When the γ-th round starts, the c-th client initializes local modelweight $\mathbf{w}_c^{\gamma-1}$ with the global model weight $\mathbf{g}^\gamma$ received from the server and starts local training with local training dataset $\mathcal{D}_c$. After the local training is completed as described in Subsection II-B.1, the client transmits the trained local models weight $\mathbf{w}_c^\gamma$ to database in global server. Then the server computes each predicted location $\hat{\mathbf{y}}_{c,i}^v$ of the i-th fingerprint $\mathbf{x}_i^v$ in the validation dataset $\mathcal{D}^v$ with the c-th local model weights $\mathbf{w}_c^\gamma$ which is stored in the server database. By repeatedly predicting validation dataset T times with MC dropout for the c-th client local model weight, T localization errors for one validation fingerprint are obtained.

The localization error for the i-th validation fingerprint of the c-th client is represented by

$$e_{c,i} = \left\|\hat{\mathbf{y}}_{c,i}^v - \mathbf{y}_i^v\right\|_2. \tag{5}$$

Hereafter, the localization error $e_{c,i}$ is expressed as a random variable $\mathcal{E}_{c,i}$. By MC random sampling, we can approximate the distribution of random variable $\mathcal{E}_{c,i}$ with T localization error samples. The variance of prediction error of the i-th validation input $\sigma_{c,i}^2$ is calculated as follows

$$\sigma_{c,i}^2 = \mathbb{E}\left[\left(\mathcal{E}_{c,i} - \bar{\mathcal{E}}_{c,i}\right)^2\right], \tag{6}$$

where $\bar{\mathcal{E}}_{c,i}$ is the mean of the prediction error of the i-th validation input. Each client has N^v variances corresponding to the number of validation dataset, and the uncertainty of the c-th client $\mathcal{U}_c$ is the mean of the variances for the entire

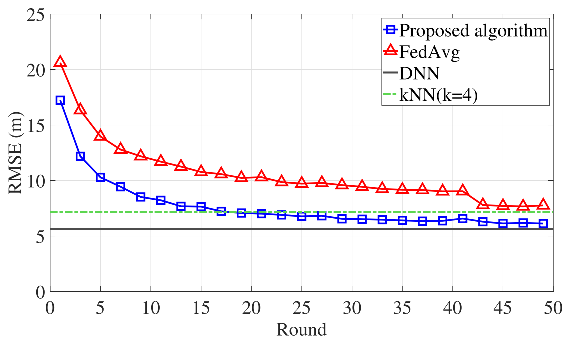

Fig. 3. The localization error of the four localization methods tested.

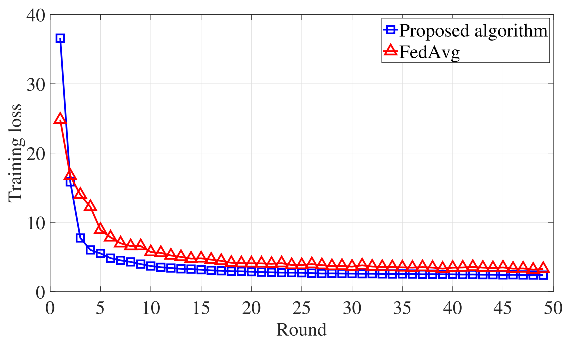

Fig. 4. Training loss versus round.

validation inputs, which is given by

$$\mathcal{U}_c = \frac{1}{N^v} \sum_{i=1}^{N^v} \sigma_{c,i}^2. \tag{7}$$

Finally, we propose the reliability of the c-th client which is calculated by $\mathcal{R}_c = (1/\mathcal{U}_c)^\alpha$; the value of reliability gets larger when the client has a certain model. The reliability is reciprocal with uncertainty, and α is an exponent parameter of inverse of uncertainty.

C. Global Model Weight Update

After the server calculates the reliability $\mathcal{R}_c^\gamma$ with the trained local model weight in the γ-th round, the global model updates the global weight $\mathbf{g}^{\gamma+1}$. Our proposed FL system applies the normalized reliability $\mathcal{R}_c^\gamma / \sum_{c=1}^{C} \mathcal{R}_c^\gamma$ as the linear combination coefficient of the c-th client u_c^γ in (2). The global model weight of the next round is updated as follows

$$\mathbf{g}^{\gamma+1} = \sum_{c=1}^{C} \frac{\mathcal{R}_c^\gamma \mathbf{w}_c^\gamma}{\sum_{c=1}^{C} \mathcal{R}_c^\gamma}. \tag{8}$$

As these processes repeat, the server updates the global model weights which are biased toward the local client with high reliability. To sum up the whole system, the uncertainty of the local model is approximated using MC dropout. Then, the server applies the client reliability into updating the global model, where the reliability is a value that is inversely proportional to local model the uncertainty. Algorithm 1 presents the details of summary for the proposed scheme.

V. Simulation Results

In this section, we present settings for FL fingerprint localization and then evaluate the performance of the proposed FL localization system with the real-life WiFi RSS dataset.

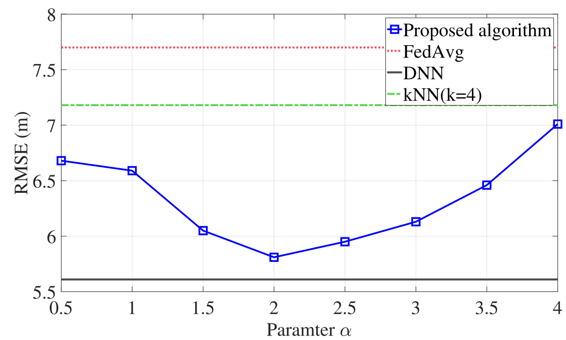

Fig. 5. Localization error versus the exponent parameter α.

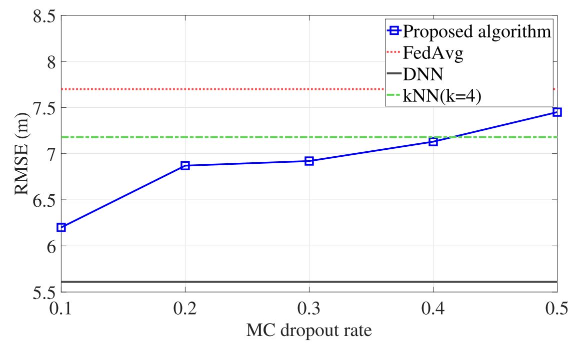

Fig. 6. Localization error versus the MC dropout rate p_d.

1) Simulation Setup: UJIIndoorLoc dataset was collected in the multi-building and multi-floor indoor environment over three buildings of about 110,000 m^2. In order to reduce the effort required for data collection, more than 20 mobile clients used different 25 devices to collect crowdsourced RSS fingerprints, which is appropriate in the FL system [13]. We utilized only two subsets in the dataset: training and testing. The validation dataset which is needed to calculate the uncertainty of local models is randomly selected from the testing dataset by the ratio of 0.2. To divide the local dataset for distributed learning, we used the labeled training data which is already categorized by 18 different mobile clients.

We present the localization performance compared to federated averaging (FedAvg), non-FL models with DNN and k-nearest neighbor (kNN) with $k = 4$ [13]. The value $k = 4$ was the one providing the best performance among several tested. FedAvg is a typical approach for implementing FL, where u_c^γ in (2) is set to $u_c^\gamma = N_c^\gamma / \sum_{c=1}^{C} N_c^\gamma$ [17]. In training the FedAvg and DNN deep learning models, MC dropout is not considered. Recall that kNN is a machine learning algorithm that does not employ a NN, which is a standard method of estimating client location in fingerprint localization systems. The simulation parameters are shown in Table I. Also the localization error is defined as root mean squared error (RMSE).

2) Localization Performance: Fig. 3 presents the localization error of the proposed algorithm compared to FedAvg, DNN and kNN results. In this simulation, the MC dropout rate of the proposed algorithm is set to 0.1, and the exponent parameter is $\alpha = 2$. The kNN shows the worst localization performance with a value of 7.18 m, where deep learning based method seems better than conventional methods. Since

TABLE I
SIMULATION PARAMETERS

Parameters	Value
Optimizer	Adam
Loss function	Mean Absolute Error
Learning rate	0.001
Network structure	$1024 \times 512 \times 64$
Activation function	sigmoid$\times$3, linear
Batch size	20
Epoch	20
Round	50

there is no process of repeatedly aggregating local model weights in DNN, the localization error of DNN is a constant value of 5.61 m over entire rounds. The DNN model has the optimal localization performance because it is centralized learning with fingerprints collected by all users. Compared to DNN and kNN, localization error of two FL approaches decrease and finally converge to 7.76 m in FedAvg and 6.06 m in the proposed algorithm at round 50. Although the performance degradation in FL is taken for granted because clients possess part of the dataset, the proposed algorithm improves the localization error by applying the reliability of clients compared to FedAvg.

3) Training Loss and Performance Evaluation With α and MC Dropout Rate: Fig. 4 shows the training loss for the training iterations. The proposed scheme considers the same assumptions as the FedAvg convergence conditions [18]. The training loss shows that the proposed algorithm can achieve better performance in terms of convergence time and accuracy, where the proposed algorithm converges about 2.5 after round 30, and the FedAvg converges 3.4 after round 37. Moreover the proposed algorithm introduces two system parameters: exponent parameter α and MC dropout rate. In Fig. 5 and Fig. 6, we present the impact of the exponent parameter and MC dropout rate on localization performance. The results of DNN, FedAvg, and kNN are constant values across those parameters since they do not depend on those, which is the same with the Fig. 3. At $\alpha = 2$, the localization error of the proposed algorithm is about 5.83 m, which achieves the best localization performance. For the MC dropout rate, the model uncertainty increases as the MC dropout ratio, which can lead to lower accuracy. Through this system parameter analysis, we observe that the proposed algorithm outperforms FedAvg, and that proper parameter adjustment can yield to a better localization performance.

VI. CONCLUSION

In this letter, we propose an improved FL approach by applying the reliability of mobile clients in the FL localization system. The reliability is quantified as the predicted variance of the local models that can be obtained by Bayesian deep learning. We designed the FL system to compute such approximate Bayesian deep learning by applying a Monte Carlo dropout for efficient computation. By introducing such reliability measure to the FL system, the resulting global model is biased towards the local model with higher reliability. The proposed approach can be generally applicable to FL schemes, while this letter focus on its use within an indoor fingerprinting-based localization system. Our simulation results show that the proposed algorithm has 1.7 m more accurate localization performance than the existing approaches.

REFERENCES

[1] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz, "Wireless RSSI fingerprinting localization," *Signal Process.*, vol. 131, pp. 235–244, Feb. 2017.

[2] D. Dardari, P. Closas, and D. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[3] G. Félix, M. Siller, and E. N. Alvarez, "A fingerprinting indoor localization algorithm based deep learning," in *Proc. 8th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2016, pp. 1006–1011.

[4] L. Gogolak, S. Pletl, and D. Kukolj, "Indoor fingerprint localization in WSN environment based on neural network," in *Proc. IEEE 9th Int. Symp. Intell. Syst. Informat.*, Sep. 2011, pp. 293–296.

[5] E. Arias-de-Reyna, P. Closas, D. Dardari, and P. M. Djuric, "Crowd-based learning of spatial fields for the Internet of Things: From harvesting of data to inference," *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 130–139, Sep. 2018.

[6] F. Yin *et al.*, "FedLoc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, 2020.

[7] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[8] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[9] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[10] P. Wu, T. Imbiriba, J. Park, S. Kim, and P. Closas, "Personalized federated learning over non-IID data for indoor localization," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2021, pp. 421–425.

[11] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[12] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," 2015, *arXiv:1505.05424*.

[13] J. Torres-Sospedra *et al.*, "UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2014, pp. 261–270.

[14] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn. (DIDL)*, Dec. 2018, pp. 1–8.

[15] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1050–1059.

[16] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," 2019, *arXiv:1910.09457*.

[17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[18] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, *arXiv:1907.02189*.