Federated Distillation Based Indoor Localization for IoT Networks

Yaya Etiabi, Graduate Student Member, IEEE, and El Mehdi Amhoud, Member, IEEE

Abstract—The federated distillation (FD) paradigm has been recently proposed as a promising alternative to federated learning (FL), especially in wireless sensor networks with limited communication resources. However, all state-of-the-art FD algorithms are designed for classification tasks only, and less attention has been given to regression tasks. To take advantage of this promising paradigm for localization which is inherently a regression problem, we introduce in this work an FD framework that is tailored to address regression learning problems. More specifically, we propose an FD-based indoor localization system that shows a good tradeoff in communication load versus accuracy compared to FL-based indoor localization. With our proposed framework, we reduce the number of transmitted bits by up to 98%. We analyze the energy efficiency regarding the number of communication rounds for both FD and FL systems to achieve the same localization accuracy. The results revealed that FD comes with greater energy efficiency by reasonably saving transmission energy at the expense of computation energy. This notably represents a significant advantage in the context of wireless sensor networks, particularly given the prevalence of battery-powered Internet of Things (IoT) devices operating with constrained bandwidth. Moreover, we show that the proposed framework is much more scalable than FL and, thus more likely to cope with the expansion of wireless networks.

Index Terms—Federated distillation (FD), indoor positioning, Internet of Things (IoT), localization, radio signal strength indicator (RSSI) fingerprinting, wireless networks.

I. INTRODUCTION

LOCATION-BASED services (LBSs) play an important role in several applications by providing targeted information to individuals or entities based on their geographic location in real or near-real time, typically through wireless communication networks. These applications include navigation, individual tracking, emergency services, asset tracking, logistics planning, workforce management, location-based advertising, and social networking. In recent decades, LBS have grown considerably and are now more than ever at the core of the digital revolution we are witnessing. Their market is expected to reach $318.64 billion in 2030 [1] due to the increase in demand from different sectors such as agriculture, defense, transportation, energy, and healthcare. This growth is the direct result of the evolution of the underlying technologies centered around wireless networks.

Today with the advent of 5G [2], and the prospect of 6G [3], the number of connected devices will grow at an unprecedented rate, resulting in the massive deployment of Internet of Things (IoT) infrastructures [4]. Moreover, 6G comes with new applications such as multisensory extended reality (XR) applications, connected robotics, and autonomous systems [5], [6]. It is also expected to offer sensing and localization as new services [7], consolidating communication, radar sensing, and localization services over the same wireless network [8], [9]. All these transformational applications drive the need for accurate localization systems which require lots of resources due to the massive deployment of IoT devices. Yet in the literature, in addition to the range-free techniques such as centroid method [10] and distance vector hop (DV-Hop) technique [11], typical ranging techniques based on channel state information (CSI) [12], angle of arrival (AoA) [13], time of arrival (ToA) [14], [15], time difference of arrival (TDoA) [16], and radio signal strength indicator (RSSI) [17] using various wireless technologies such as radio frequency identification (RFID) [18], ultra-wide bandwidth (UWB) [19], WiFi [20], LoRaWAN [21], and Bluetooth [22] have been proposed for indoor positioning. All of these techniques present a number of issues, including low precision, high computational complexity, and unreliability due to wireless channel impairments such as multipath effects caused by nonline of sight (NLOS) propagation in indoor environments [23], while most positioning devices lack sufficient computing power. Moreover, these techniques require the building of empirical

Manuscript received 1 December 2023; accepted 19 January 2024. Date of publication 29 January 2024; date of current version 2 April 2024.

The associate editor coordinating the review of this article and approving it for publication was Dr. Avik Santra. (Corresponding author: Yaya Etiabi.)

The authors are with the College of Computing, Mohammed VI Polytechnic University, Ben Guerir 43150, Morocco (e-mail: yaya.etiabi@um6p.ma).

Digital Object Identifier 10.1109/JSEN.2024.3357798
models that capture all the channel effects including noises, multipath fading, and channel variations due to the indoor movements, leading to a very high time complexity and a high dependence on channel conditions.

Thus, due to the difficulty of deriving robust models that capture these indoor channel impairments, researchers turn toward data-based localization using machine learning (ML) which does not require empirical models but rather uses constructed datasets that capture all variations in the indoor environments. Indeed, ML is used to tackle the aforementioned limitations of traditional methods enlightened in the works in [24], [25], [26], and [27]. In these works, we can see that ML is a very promising and game-changing technology for IoT localization in the sense that it provides robust and scalable localization systems with improved accuracy and relatively low complexity.

Nonetheless, ML-based localization systems require important data exchange between IoT devices and the central server. To proceed with the latter, ML operations with respect to the privacy of users, federated learning (FL), has been introduced.

FL, originally introduced by Google researchers [28], is an ML paradigm where end devices collaboratively learn a shared prediction model while maintaining all training data on-device. This collaborative learning process involves iterative model updates through the sharing of model weights. Indeed, the work in [29] has proposed an FL-based localization framework coined FedLoc which underscores the prominence of such collaborative algorithms, especially in addressing privacy concerns in IoT collaborations. However, the scale of IoT networks, characterized by a massive deployment of sensors, challenges the effectiveness of FL. The implementation of capacity-constrained communication links can lead to significant limitations in terms of communication cost and latency, impacting overall performance. To address these challenges, federated distillation (FD) emerges as a compelling alternative [30]. FD, a distributed learning solution, distinguishes itself by exclusively exchanging model outputs, which are significantly smaller in dimensions than the model size. Although in the literature FD has primarily been employed in classification tasks such as handwritten digits recognition [31] and image classification [32], this work extends its application to a regression problem. Specifically, we focus on predicting the location of an IoT device based on its RSSI measurements in that location. The motivation for this research is twofold as follows.

1) *Alternative to FL in Bandwidth-Constrained IoT Networks*: In environments characterized by a massive deployment of sensors and constrained communication links, FL faces substantial challenges related to communication cost and latency. Our work addresses this by proposing FD as a more suitable paradigm for bandwidth-constrained IoT networks. FD, with its emphasis on reduced bandwidth consumption through output-only exchanges, is instrumental in mitigating the challenges associated with FL in such environments.

2) *Extension of FD to Regression Tasks for Efficient Target Location Estimation*: While FD has shown success in classification tasks, its application to regression problems, particularly in target location estimation, remains unexplored. This research aims to bridge this gap by adapting FD to regression tasks, specifically focusing on predicting the location of an IoT device based on its RSSI measurements. This extension seeks to enhance the precision and efficiency of target location estimation in IoT environments.

In summary, this work underscores the need for an alternative learning paradigm in bandwidth-constrained IoT networks and emphasizes the unexplored potential of extending FD to regression tasks for more efficient target location estimation. The main contributions of this work are summarized as follows.

1) We develop a FD framework for regression tasks since the previously proposed FD algorithms are dealing with classification problems and no attention has been given to the regression ones. We validate our FD framework through performance evaluation using different publicly available experimental datasets.

2) We propose an IoT localization system based on our proposed FD framework. We prove that this localization framework works for indoor systems as well as for outdoor systems. Also to the best of our knowledge, this work is the first to tackle the localization problem under a FD framework which considerably reduces the communication complexity over bandwidth-constrained wireless networks.

The remainder of this article is organized as follows. Section II presents related works. In Section III, we describe our proposed FD framework followed by its performance evaluation in Section IV. Section V comes with discussion and future works. Finally, Section VI concludes our work and provides future research directions.

## II. Related Work

In this section, we delve into the existing literature related to RSSI fingerprinting-based localization, federated learning (FL)-based IoT localization frameworks, and FD, laying the groundwork for our proposed approach.

### A. RSSI Fingerprinting-Based Localization

Fingerprinting-based methods involve constructing an RSSI database by extracting representative and distinguishable parameters, termed fingerprints, from received signals at various locations within the IoT network. This constructed database is then utilized to predict the location corresponding to a new RSSI recording based on its similarity with the recorded fingerprints. However, the accuracy of RSSI measurements is challenged by channel impairments, particularly multipath fading induced by NLOS signals, resulting in the analytical complexities associated with finding solutions. These challenges are further accentuated by the growing scale of IoT networks, necessitating a systematic adaptation of localization schemes to align with the mainstream requirements of LBS. In response to these challenges, ML, particularly deep learning frameworks, has been introduced to strike a balance between accuracy and complexity [33], [34]. Recent advancements in
ML, as shown in these works and more specifically in [24], have led to the proposal of various deep learning frameworks tailored for RSSI fingerprinting-based localization, highlighting the continual evolution of methodologies in achieving accurate and efficient localization within IoT networks.

### B. FL-Based IoT Localization Frameworks

While RSSI-based fingerprinting has been a focal point of research in the past decade, achieving significant breakthroughs, it grapples with an inherent challenge: rapid degradation of localization accuracy over time due to the dynamic environment and unstable wireless devices. This often leads to a high calibration effort for fingerprint collection. Consequently, the conventional ML approach results in a large fingerprint database on the server, raising privacy concerns. To address these challenges, researchers have increasingly turned to FL models for the design of localization systems. For instance, the work in [35] introduces FLoc, an FL framework specifically tailored for indoor localization in an RSSI fingerprinting-based scenario. FLoc aims to tackle security challenges associated with updating the fingerprint database for localization purposes. Similarly, [29] and [36] also leverage FL in the design of localization systems, emphasizing the advantages of privacy preservation and reduced training computation load. More specifically, Wu et al. [37] introduced a personalized FL approach that contributes to addressing challenges related to dynamic data distributions, privacy concerns, and personalized localization requirements. Cheng et al. [38] proposed a computable metric, the area of the convex hull, to characterize database heterogeneity. They utilized this metric for aggregation in FL, resulting in enhanced performance in heterogeneous scenarios. Park et al. [39] introduced a novel model weight update method that specifically tackles challenges arising from nonindependently and identically distributed (non-IID) data configurations in FL-based indoor localization. The authors quantified the reliability of local clients based on model uncertainty, employing Monte Carlo dropout for computational efficiency in approximating Bayesian uncertainty. This approach demonstrated notable improvements in localization performance. In addition, Gufran and Pasricha [40] presented a framework that integrates indoor localization with selective weight adjustment, enhancing accuracy in device-heterogeneous environments while prioritizing user data privacy. These works collectively showcase substantial efforts dedicated to advancing indoor localization systems by leveraging various FL improvements, with a keen focus on computational efficiency and privacy preservation aspects. However, despite the advantages of FL in privacy preservation and computation load reduction, it falls short of meeting the bandwidth and energy constraints of real-world IoT systems. The communication load and energy consumption involved in FL when exchanging the model parameters are often unsuitable for IoT devices with limited capabilities. In this work, we address these limitations by introducing FD.

### C. Federated Distillation

FD is a unique amalgamation of knowledge distillation (KD) and FL. KD, as introduced in [41], adopts a student–teacher paradigm where a student model is assisted in training by a pretrained teacher model. This approach facilitates knowledge transfer from the teacher to the student, expediting the learning process. Notably, both the student and the teacher traditionally share access to the same dataset. A distributed variant, named co-distillation (CD), has been proposed in [42], where each student, instead of relying on a pretrained teacher, leverages the aggregated knowledge of the remaining students as its teacher. This leads to an online teacher model training concurrently with student models. FD can be viewed as a derivative of CD, as elucidated in [43], with a distinguishing feature—the possession of individual data by each FD student, akin to the paradigm in FL. The asymptotical analysis, utilizing the kernel method presented in [44], demonstrates that FD can outperform KD, especially when the teacher model in the latter is not well-pretrained. Recent works [30], [45] introduce FD to address the limitations of FL, particularly in terms of bandwidth requirements and energy constraints. Despite these advancements, all proposed FD frameworks to date have been exclusively tailored for classification tasks, neglecting regression tasks. This gap is significant as many real-world ML target variables are continuous and necessitate treatment as regression problems. In response, our contribution begins by introducing an FD framework specifically designed for regression tasks. Subsequently, we leverage this framework to construct an indoor localization system.

### III. Proposed Framework

#### A. Localization System Model

In traditional settings, RSSI data from multiple anchor nodes are collected through a crowdsourcing system and merged in a server for centralized training of the deep neural network (DNN)-based localization model. In this work, we substitute this with a decentralized approach where collected RSSI data are no longer sent to the server. Instead, a pool of IoT devices, namely, FD clients is selected to crowdsouce RSSI data all over the network. Then, each selected device trains a local model with its crowdsourced data. In the FD training setting, FD clients share their knowledge by exchanging their models’ outputs so that at the end of the training, the local models predict nearly the same positions given new RSSI measurements. Thus, for a node to be localized in this network, it records RSSI from in-range anchors and sends them to the nearest FD client which outputs the corresponding position by running its model. Note that the predicted position remains essentially the same as if the prediction was done by any other FD client. Indeed, the models are trained in a manner that provides them with a global knowledge of the overall network. The FD system architecture and the whole FD training process are described in Sections III-B and III-D respectively.

#### B. FD System Architecture

The overall framework contains three main components, namely, the parameter server called the coordinator also known as the teacher, the clients or workers also known as students, and the communication system.
ETIABI AND AMHOUD: FEDERATED DISTILLATION BASED INDOOR LOCALIZATION FOR IoT NETWORKS 11681

Fig. 1. Illustration of FD-based Regression training process. (a) Illustration of segment-based knowledge sharing. (b) Target variable segmentation.

1) **Server**: The server (coordinator or teacher) is the entity that supervises the whole FD training process. It starts by defining all the common parameters such as the total number of communication rounds, the number of clients to be selected for each communication round, and the wireless communication settings. The server is also in charge of defining the global DNN model. The necessary information to reproduce locally the global model is then broadcast to the clients with the help of the communication system.

2) **Clients**: The clients (workers or students) are in charge of training their respective local models using their private datasets. During the training, workers periodically upload their local average estimations per each target segment to the parameter server which will aggregate them to obtain a global estimation per each corresponding segment. The global estimations from the server are then used by the clients to update their local average estimations per each target segment to train their respective local models using their private datasets. During the training, workers periodically upload their local average estimations per each target segment to the parameter server which will aggregate them to obtain a global estimation per each corresponding segment. The global estimations from the server are then used by the clients to update their local average estimations per each target segment to train their respective local models using their private datasets.

3) **Communications System**: The role of the communication system as its name suggests is to deal with the wireless communication techniques used to communicate the model and the target estimations from the server to clients and vice versa. This module is further discussed in Section III-F.

C. Problem Formulation

The objective of our proposed framework is to cooperatively train a DNN using a communication-efficient FL scheme, namely, FD, in a network of IoT devices. For the sake of convergence analysis in Section III-E, we consider a three-layer neural network comprising an input layer, a hidden layer, and an output layer with, respectively, \( N_i \), \( N_h \), and \( N_o \) neurons as shown in Fig. 2. Given an input vector \( X_i \), the prediction of the target \( y_i \) is given by \( \hat{y}_i = F_{\Theta}(X_i) \), where \( F_{\Theta} \) is the function representing our DNN and \( \Theta = [ W a ] \) (see Fig. 2) is the set of the DNN’s weights. Then, for standalone training, the goal is to minimize the loss function given by

\[
\mathcal{L}(y, \hat{y}) = \sum_i \mathcal{L}(y_i, \hat{y}_i) \tag{1}
\]

where

\[
\hat{y}_i = F_{\Theta}(X_i) = \frac{1}{\sqrt{N_0}} \sum_{n=1}^{N_h} \alpha_n \ell_n(X_i) \tag{2}
\]

with \( \ell_n(X_i) = \sigma(W_n^T X_i) \) is the logit representing the output of the \( n \)th neuron of the last hidden layer, \( \sigma(\cdot) \) a nonlinear activation function, and \( W_n \) the weights vector of the \( n \)th neuron. For the localization framework, the input \( X_i \) is a vector containing the RSSI measurements from all the access points in the network, and the target variable \( y_i \) is the 2-D coordinates of the location where these RSSIs have been recorded. To simplify the notations we will consider \( F_{\Theta}(X_i) = F(X_i) \) in the next sections.

In the vanilla distillation process so-called KD, each student independently learns the knowledge of the teacher by adding a regularization term regarding the gap between its prediction and the teacher’s. Ideally, similar to the distillation in classification context in [42], the distillation for a learning task can be formulated as

\[
\mathcal{L}(y, \hat{y}) = \sum_i \mathcal{L}(y_i, \hat{y}_i) + \lambda \sum_i \sum_{n=1}^{N_h} \mathcal{L}(\ell_n(X_i), L_n(X_i)) \tag{3}
\]

where \( L_n(X_i) \) is the pretrained teacher \( n \)th output logit given an input \( X_i \) and \( \lambda \) is the regularization coefficient, assuming that the teacher and the student have access to the same dataset.

On the other hand, CD is a distributed version of this system where workers have to distill knowledge from each other. In fact, the main principle of CD is to consider a set

Fig. 2. Three-layer multilayer perceptron architecture.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
of predictions from several individual models as the teacher’s knowledge, which is frequently more correct than the individual predictions. In this configuration, each student sees the ensemble of remaining students as its teacher. Consequently, (3) is transformed into

\[ \mathcal{L}(y, \hat{y}) = \sum_k \left[ \mathcal{L}(y_i, \hat{y}_i) + \lambda \sum_n \mathcal{L} \left( L^k_n(X_i), \ell^k_n(X_i) \right) \right] \]  

(4)

where \( L^k_n(X_i) = (1/K - 1) \sum_{k \neq k} \ell^k_n(X_i), \forall k \) with \( K \) the number of students. Note that this loss function encompasses all the individual losses for students since this distributed student–teacher configuration creates an interdependence in the distillation process. However, for the sake of computation and privacy of the federated system. As such, one approach to get the boundaries of each local dataset, which may violate the

\[ I_s = \{ -\infty, y^1 \} \quad \text{if} \quad s = 1 \]

\[ \{ y^{s-1}, y^s \} \quad \text{if} \quad 1 < s < S \]

\[ \{ y^{s-1}, +\infty \} \quad \text{if} \quad s = S \]  

where \( y^s = y_{\min} + s \times \epsilon \) with \( \epsilon = (y_{\max} - y_{\min})/S \).

Remark 1: The number of segments \( S \) is a hyper-parameter depending on the resolution \( \epsilon \) set for the learning task, i.e., regarding a regression problem, the resolution \( \epsilon \) can be the maximum tolerable error in estimation. Then, \( S \) is given by

\[ S = \frac{y_{\max} - y_{\min}}{\epsilon}. \]

Remark 2: Since the data is distributed across devices, computing the min and the max requires the server to know the boundaries of each local dataset, which may violate the privacy of the federated system. As such, one approach to get this done securely is through a multiparty computation scheme between devices. However, for the sake of computation and communication efficiency, the server can set them arbitrarily or request devices to upload their local min and max using cryptographic tools with order-preserving encryption to preserve devices’ data privacy.

For a client \( k \) in the network, we define the following settings:

\[ \mathcal{I}^k_s = \{ i \} \text{ such that } y_i \in I_s \]

and \( (X_i, y_i) \in D_k \), with \( |\mathcal{I}^k_s| = N^k_s \)

\[ \ell^k_{n,s} = \frac{1}{N^k_s} \sum \ell^k_{n,s}(X_i) \]

\[ L^k_{n,s} = \frac{1}{K - 1} \sum_{k \neq k} \ell^k_{n,s} \]

where \( I^k_s \) is the partition of \( k \)th client’s dataset belonging to the \( s \)th segment. \( \ell^k_{n,s}(x) \) is the \( n \)th output logit of client \( k \) when the input \( x \) belongs to segment \( s \). \( \ell^k_{n,s} \) \( L^k_{n,s} \) are, respectively, the local and global average of the \( n \)th output logit over the samples in segment \( s \).

The above formulations lead to the new global loss function of the federated system given by

\[ \mathcal{L}(y, \hat{y}) = \sum_k \sum_s \left( \sum_{i \in \mathcal{I}^k_s} \mathcal{L}(y_i, \hat{y}_i) \right) + \lambda \sum_{i \in \mathcal{I}^k_s} \sum_n \mathcal{L} \left( L^k_{n,s}, \ell^k_{n,s}(X_i) \right) \]  

(7)

D. FD for Regression Algorithm

Considering only the computation aspect of our framework, the overall process of solving the problem in (7) is described by Algorithm 1 which comprises both client-side and server-side pseudocode.

In fact, in this configuration, in contrast to KD, we do not have a trained teacher model. So both the student and the teacher are learning during the process especially since the teacher’s knowledge for a given student is actually the aggregated knowledge of the remaining students.

As a consequence, at the initial communication round, it is not possible to obtain the regularization term of the loss function since no prediction has been done yet. That is why, as enlightened in “line 1–line 3” of Algorithm 1, we start by training the local models for a few steps. Until then, the distillation process can start (“line 6–line 36”).

The overall process in Algorithm 1 can be summarized in the following steps.

1) Each client trains its local model using its private dataset and stores locally the average estimations per segment.
2) Each client periodically uploads its local average estimations per segment to the server.
3) The server computes the global average estimations per segment by averaging all the local average estimations per segment sent by all the clients.
4) Each client downloads the global average estimations per segment and updates its loss function according to (7).
5) Each client repeats all the above steps until convergence.
E. Convergence Analysis

After applying our segmentation technique and considering the mean square error (MSE) as the loss function, i.e., \( \mathcal{L}(y, \hat{y}) = \text{MSE}(y, \hat{y}) \) in (7), we globally try to solve the following optimization problem:

\[
\min_{W_1, W_2, \ldots, W_k} \sum_{s} \sum_{i \in \mathcal{I}_s^k} (y_i - \hat{y}_i)^2 + \lambda \sum_{s} \sum_{j \in \mathcal{J}_s^k} \left[ (L_{n,s}^k - \ell_{n,s}^k(X_i))^2 \right].
\]  

The convergence of the latter optimization problem can be established through the asymptotical analysis of the two foundational building blocks of FD, namely, KD [41] and CD [42] by exploiting the theory of neural tangent kernel (NTK) [44].

1) Convergence Analysis of KD: The problem of KD in this configuration is cast into the minimization of the objective function defined as follows:

\[
J(W_k) = \sum_s \sum_{i \in \mathcal{I}_s^k} (y_i - \hat{y}_i)^2
\]

where \( W_k \) contains the weights of the model of a client \( k \). We assume that the activation function \( \sigma(\cdot) \) in the hidden layers is Lipschitz continuous and differentiable. The first derivative \( \sigma'(\cdot) \) is also assumed to be Lipschitz continuous. Using the gradient descent algorithm with an infinitesimal step-size to solve the problem in (3) in the NTK settings [44] results in the following dynamics of the weights:

\[
\frac{d}{dr} \Phi_n^k = \sum_s \frac{\alpha_n}{\sqrt{N_h}} (y^s - \hat{y}^s) + \lambda \left( \frac{L^k_{n,s} - \ell^k_{n,s}}{\sqrt{N_h}} \right) \tag{10}
\]

Proof: In the kernel regime [44], with an infinitesimal step size \( \eta \), the gradient descent iterations depicted by\( W_k^n (r + 1) = W_k^n (r) - \eta \nabla J(W_k^n (r), D) \), \( \forall n = 1, 2, \ldots, N_h \) can be translated into continuous time domain variation given by

\[
\frac{d}{dr} W_k^n = \sum_s \Phi_n^k \left( \frac{\alpha_n}{\sqrt{N_h}} (y^s - \hat{y}^s) + \lambda \left( \frac{L^k_{n,s} - \ell^k_{n,s}}{\sqrt{N_h}} \right) \right).
\]

Let \( \Phi_n^k \) be the line vector containing the values \( X_i \cdot \sigma'(W_k^n X_i) \), \( y^s \) the vector of labels \( y_i \) such that \( i \in \mathcal{I}_s^k \). Similarly, \( \ell^k_{n,s} \) and \( L_n^k \) are the vectors containing the \( n \)th predicted logits of all the samples with labels in segment \( s \) (i.e. \( y_i / i \in \mathcal{I}_s^k \)) for the student and the teacher, respectively. Then, it follows that:

\[
\frac{d}{dr} W_k^n = \sum_s \Phi_n^k \left( \frac{\alpha_n}{\sqrt{N_h}} (y^s - \hat{y}^s) \right).
\]
Indeed, where dynamics of the logits as analyzing the dynamics of the weights, (10) is translated in discrete algorithm to a smooth curve. Due to the difficulty of analyzing the dynamics of the weights, (10) is translated in dynamics of the logits as

\[
\frac{d}{dt} \ell^s_k = \sum_s \Phi^s_n \left[ \frac{\alpha_n}{\sqrt{N_h}} (y^s - \hat{y}^s) + \lambda (L^s_{n,k} - \ell^s_n) \right].
\]

(11)

Indeed

\[
\frac{d}{dt} \ell^s_k(X_i) = \frac{d}{dW_k} \left[ \Phi^s_n \left[ \frac{\alpha_n}{\sqrt{N_h}} (y^s - \hat{y}^s) + \lambda (L^s_{n,k} - \ell^s_n) \right] \right].
\]

Proof: The theory of linear systems with a finite order [46] has established that linear dynamics of finite order can roughly represent the gradient descent behavior on the over-parameterized neural network, and the evolution of \( F^s(\tau) \) can be written as

\[
F^s(\tau) = F^s_\infty + \sum_{j=1}^{d} \phi^s_j e^{-\gamma \tau}, \forall s
\]

with \( d \) representing the order of the linear system. \( \{\phi^s_j\}_{j=1}^{d} \) are complex-valued vectors determined by the specifications of the dynamics, whereas \( \{\gamma^s_j\}_{j=1}^{d} \) are the poles of the linear system. Note that the nonzero complex-values \( \gamma^s_j \neq 0 \) correspond the singular points of the Laplace transform of \( F^s(\tau) \) except the zero component \( \gamma^s_0 = 0 \) which corresponds to the constant \( F^s_\infty \).

Thanks to the assumptions on the eigenvalues of the matrices \( \Psi \) at initialization (i.e., under mild assumptions on eigenvalues of \( \Psi(0) \)) in the kernel regime [44], [46] has shown that with bounded inputs, bounded weights, all existing poles are positive-valued, yielding \( \lim_{\tau \to \infty} F^s(\tau) = F^s_\infty \). Moreover, the analysis in the infinite width regime of the neural network \( (N_h \to \infty) \) with the set assumptions demonstrates the calculation of \( F^s_\infty \) and produces the result in (13).

Remark 4: The difference between the prediction of the student and that of the teacher can be approximated or estimated by

\[
\epsilon^s = \|F^s_\infty - y^s\|_2 = \frac{\alpha}{\alpha + \lambda} \left\| y^s - \sum_n \frac{\alpha_n L^s_n}{\sqrt{N_h}} \right\|_2, \forall s.
\]

(15)

This result implies that the learning accuracy of the student is highly correlated to the quality of the teacher in representing labels since for an imperfect teacher, the error \( \epsilon^s \) monotonically increases with \( \lambda \).

2) Federated Distillation: FD extends KD to CD where each worker possesses its own data and sees the other workers as labels since for an imperfect teacher, the error \( \epsilon^s \) monotonically increases with \( \lambda \).

Lemma 1: The student network output converges asymptotically as follows:

\[
\lim_{\tau \to \infty} F^s(\tau) = F^s_\infty = \frac{1}{\alpha + \lambda} \left( \alpha y^s + \lambda \sum_{n=1}^{N_h} \frac{\alpha_n L^s_n}{\sqrt{N_h}} \right), \forall s
\]

(13)

where \( \alpha = \sum_{n=1}^{N_h} (\alpha_n / N_h) \alpha_n^2 \).

Lemma 2: Based on the results of Lemma 1, and the interactions between students, our optimization problem in (8) converges asymptotically as

\[
\lim_{r \to \infty} F_k(r) = y, \forall k \in \{1, 2, \ldots, K\}
\]

(16)

where \( r \) designates the \( r \)th communication round.

Proof: The first iteration of KD is performed after initialization and a first run of the gradient descent by all students. Consequently, the first updates i.e., \( \{F^s(0)\}_{s=1}^{K} \) are shared. Then, each student \( k \) locally runs gradient descent and, by Lemma 1, its model converges to

\[
F^k(1) = \frac{1}{\alpha + \lambda} \left( \alpha y + \lambda \sum_{k' \neq k}^{K} F^k(0) \right).
\]

(17)
Therefore, at the $r$th communication round, the output of the model of the student $k$ converges to

$$F_k(r) = \frac{1}{\alpha + \lambda} \left[ \alpha y + \frac{\lambda}{K - 1} \sum_{k' \neq k} F_{k'}(r - 1) \right]$$

$$= \frac{1}{\alpha + \lambda} \left[ \alpha y + \frac{\lambda}{K - 1} \sum_{k' \neq k} \left( 1 + \frac{1}{\alpha + \lambda} \alpha y \right) \right]$$

$$+ \frac{\lambda}{K - 1} \sum_{k' \neq k} F_{k'}(r - 2)$$

$$= \frac{1}{\alpha + \lambda} \left[ \alpha y + \frac{\lambda}{(K - 1)(\alpha + \lambda)} \left( \alpha(K - 1)y + \lambda F_k(r - 2) \right) \right]$$

$$= \frac{1}{\alpha + \lambda} \left[ \alpha y + \frac{\lambda}{(K - 1)(\alpha + \lambda)} \left( \alpha(K - 1)y + \lambda F_k(r - 2) \right) \right]$$

On the other hand, (18) can be rewritten as

$$(\alpha + \lambda)F_k(r) - \alpha y = \frac{\lambda}{K - 1} \sum_{k' \neq k} F_{k'}(r - 1).$$

Then, let

$$u_{r+1} = \frac{\lambda}{K - 1} \sum_{k' \neq k} F_{k'}(r) = (\alpha + \lambda)F_k(r - 1) - \alpha y.$$ 

By introducing $u_{r+1}$ in (19), we construct a linear non-homogeneous recurrence relation given by

$$u_{r+1} = \frac{(K - 2)\lambda}{(K - 1)(\alpha + \lambda)} u_r + \frac{\lambda^2}{(K - 1)(\alpha + \lambda)^2} u_{r-1}$$

$$+ \frac{\lambda^2 \alpha y}{(K - 1)(\alpha + \lambda)^2} + \frac{\lambda \alpha y}{(\alpha + \lambda)}.$$ 

The resolution of such recurrence relation as presented in [47], provides a close-form solution defined as

$$u_r = \lambda y + \gamma \left( \frac{\lambda}{\alpha + \lambda} \right)^r + \rho \left( -\frac{\lambda}{(K - 1)(\alpha + \lambda)} \right)^r$$

with $\gamma = (\lambda/K) \sum_{k=1}^{K} F_k(0) - \lambda y$ and $\rho = (\lambda/K(\alpha + \lambda)) \sum_{k' \neq k} F_{k'}(0) - (\lambda/K)F_k(0)$. Combining (20) and (21), we obtain the following:

$$F_k(r) = \frac{1}{\alpha + \lambda} (\alpha y + u_r).$$

It follows that:

$$\lim_{r \to \infty} F_k(r) = \frac{1}{\alpha + \lambda} (\alpha y + \lim_{r \to \infty} u_r) = y$$

due to the fact that $\lim_{r \to \infty} u_r = \lambda y$, for $K \geq 2$. Indeed for $K \geq 2$, $|-(\lambda/(K - 1)(\alpha + \lambda))| < 1$ and $|1/(\alpha + \lambda)| < 1$.

Extensive numerical evaluation using simulation as well as experimental data in Section IV confirms the convergence property of our FD system.

**F. Computation Complexity and Communication System**

In this section, we conduct an analysis of the computational and communication complexities inherent in our proposed system, juxtaposed against the conventional FL approach. Within the orchestrated training loop, the aggregate energy consumption across all FL/FD components can be dissected into distinct categories, as illustrated in Fig. 3. Specifically, with $R$ communication rounds, the cumulative energy $\epsilon$ is expressed as

$$\epsilon = \left( \epsilon_T^y + \epsilon_T^d + \sum_{k=1}^{K} \epsilon_c^k + K\epsilon_u + \epsilon_a \right) \cdot R.$$ 

Here, $\epsilon_T$ represents the complexity stemming from uplink and downlink transmissions $\epsilon_T^y$ and $\epsilon_T^d$, respectively. $\epsilon_C$ encapsulates the overall computational complexity, incorporating local model update ($\epsilon_u$), local training ($\epsilon_c^k$), and aggregation ($\epsilon_a$) complexities at the server. As delineated by (6), the total energy consumption hinges on two pivotal components: 1) transmission energy and 2) computation energy. For computation energy, it is noteworthy that FL and FD exhibit comparable complexities, that is, $\epsilon_C(FD) \approx \epsilon_C(FL)$. Indeed, both FD and FL involve local computations on individual devices during the training process. The regularization term introduced in FD’s loss functions in (6) while affecting the loss calculation slightly, does not result in a significant deviation from the computation energy observed in FL. This innovative communication strategy of FD, focusing on transmitting predictions instead of model parameters, does not introduce a substantial divergence in the computational energy requirements. Hence, the crux of the comparative analysis between our proposed FD framework and FL predominantly centers on transmission energy $\epsilon_T$. Furthermore, given the inherent asymmetry of wireless channels, with only the server utilizing the downlink, the primary transmission challenge manifests in the uplink. Consequently, for the sake of simplicity and without loss of generality, our focus narrows down to the uplink transmission energy in $\epsilon_T$.

To clearly illustrate the communication behavior of our framework, we consider a simple multiple-access communication system. In fact, on the uplink, FD clients share a Gaussian
multiple-access channel whose equation is given by

$$y_r = \sum_{k=1}^{K} h_k^r x_k^r + z_r$$  \hspace{1cm} (27)$$

where $x_k^r$ is the signal to be transmitted by client $k$ at the $r$th communication round, $h_k^r$ the corresponding channel response and $z_k$ an additive independently and identically distributed (I.I.D) Gaussian noise with $z_k \sim \mathcal{N}(0, 1)$. For the sake of simplicity, the downlink communication is assumed to be noiseless so that we can focus on the more challenging shared uplink. Thus, for FD, at each iteration, each client transmits its local average estimations per segment over a wireless shared uplink channel with the access point connected to the server. Assuming that the target variables have dimension $N_o$ each divided into $S$ segments, the output vector to be transmitted is given by

$$\mathbf{v} = \begin{bmatrix} \mathbf{f}_1^r \\ \vdots \\ \mathbf{f}_{S}^r \end{bmatrix} \text{ with } \mathbf{f}_s^r = \begin{bmatrix} \mathbf{f}_s^r(1) \\ \vdots \\ \mathbf{f}_s^r(L) \end{bmatrix}.$$  \hspace{1cm} (28)$$

Thus, unlike FL where each device transmits $W \times R$ bits to the server for each communication round, in FD the total number of transmitted bits per device per communication round is given by

$$N_{\text{bits}} = S \times N_o \times R$$  \hspace{1cm} (29)$$

where $R$ is the bits resolution. The total number of parameters of our considered neural network is defined as

$$W = (N_t \times N_h) + N_h + (N_h \times N_o) + N_o.$$  \hspace{1cm} (30)$$

Remark 5: It can be noticed that $S \times N_o \ll W$ which makes our model much more communication-efficient than FL.

The lightweight nature of our system allows it to overcome bandwidth and energy limitations, as each device participating in the training process is subject to the power constraint $E[\|x_k^r\|_2^2]/T \leq \mathcal{P}$ [48], where $T$ is the number of channels and $\mathcal{P}$ is the maximum transmit power. Indeed, considering a conventional digital implementation as in [49], the channel’s uplink capacity is equally shared by the devices leading to bandwidth constraints limiting the number of bits $B$ that can be transmitted by each device. In fact, considering Shannon’s capacity [50], the maximum number of bits per transmission for each device is given by

$$B_{\text{max}}^r = \frac{T}{K} \log_2(1 + K \|h_r^r\|_2^2 \mathcal{P}), \text{ so that } B \leq B_{\text{max}}^r.$$  \hspace{1cm} (31)$$

To sum up, this constraint does not affect too much the FD system since the condition $N_{\text{bits}} < B_{\text{max}}^r$ is very likely to always hold as demonstrated by the results in Section IV.

IV. PERFORMANCE EVALUATION

A. Indoor Localization With FD

1) System Model and Data Collection: Our goal is to construct a DNN-based indoor positioning system (IPS) to predict the location of an IoT device given its RSSI measurements in that location. Therefore, the dataset is an ensemble of measurements of RSSI from different access points (APs) at different locations and at different times to capture the channel variations. To do so, we consider an indoor IoT network in an area of interest of size $l \times w \ m^2$ containing $M$ APs randomly distributed over the area. For data crowdsourcing, we consider a pool of $K$ FD clients. Each client will collect RSSI data from the environment to train a local model. The environment is characterized by its path loss exponent $\beta$ and its sigma-shadowing $\sigma$ representing the channel attenuation.

In this localization system, we exploit the RSSI values gathered from different APs in the network. Consequently, for the data collection, we consider $N$ reference positions (RPs) such that at each RP, we capture $M$ RSSI values from the $M$ APs, taking into account the out-of-range APs. At each captured RSSI, the corresponding RP is added as a label. Moreover, at each RP, the operation is repeated for $T$ time intervals in order to capture the variations and different impairments experienced by the wireless channel.

2) Simulated RSSI Data: The RSSI measurements are strongly time-space varying and depend on the path loss model for the corresponding environment. In this work, similar to work in [51], we use the log-distance path loss model as follows:

$$\text{PL} = \text{PL}_0 + 10 \beta \log_{10} \frac{d}{d_0} + X_{\sigma}.$$  \hspace{1cm} (32)$$

where $\beta$ is the path loss exponent, $d_0$ is the reference distance, and $X_{\sigma}$ represents the log-normal shadowing with standard deviation $\sigma$ in decibel. $\text{PL}_0$ is the path loss at the reference distance $d_0$ and given by the free space propagation model as

$$\text{PL}_0 = 10 \log_{10}\left(\left(\frac{4\pi d_0}{c}\right)^2\right).$$  \hspace{1cm} (33)$$

where $f$ is the frequency of the signal and $c$ is the speed of light.

Then, the RSSI is computed as follows:

$$\text{RSSI} = P_{R_x} = P_{T_x} - \text{PL}.$$  \hspace{1cm} (34)$$

Consequently, the RSSI value captured from the $m$th AP at the $n$th RP at time $t$ can be expressed as

$$\text{RSSI}_{n,t}^m = P_{T_x} - \left[20 \log_{10}\left(\frac{4\pi d_0}{c}\right) + 20 \log_{10}(f) + 10 \beta \log_{10}\left(\frac{d_n^m}{d_0}\right) + X_{\sigma,n,m}\right].$$  \hspace{1cm} (35)$$

B. Simulation Results

1) Set-Up: For the simulation, we consider a WiFi-powered IoT network of $l \times w = 20 \times 20 \ m^2$ with $M = 10$ APs where the data are collected from $N = 100$ RPs. The environment
variables \( \beta \) and \( \sigma \) are set to 3.23 and 2, respectively, according to the experimental measurements conducted in [52]. The number of repetitions is set to \( T = 10 \). In order to make the dataset reproducible, the RSSI values are generated using (35) with a random seed set to 200.

As the objective is to train the DNN localization models using FD, we set the number of clients (workers or students) \( K \) to 5. For the DNN models, we consider a multilayer perceptron (MLP) characterized by the configuration given in Table I. The simulations were executed using Keras with the TensorFlow backend on Jupyter notebooks as the interactive computing platform. These simulations were performed on a MacBook Pro mid-2014, which features a quad-core Intel Core i7 processor, 16 GB of RAM, and Intel’s Iris Graphics.

2) Results: The models are trained by the \( K \) students and the FD server using Algorithm 1. The evaluation is done using the mean absolute error (MAE) metric defined by

\[
\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - \hat{x}_i| + |y_i - \hat{y}_i|}{2}
\]

(36)

where \( y_i = [x_i \ y_i]^T \) and \( \hat{y}_i = [\hat{x}_i \ \hat{y}_i]^T \) are, respectively, the true and estimated locations corresponding to the \( i \)-th RSSI recording in the validation dataset. We further utilize the root mean squared error (RMSE) in our evaluation, which is defined as follows:

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}.
\]

(37)

The results are presented in Fig. 4. In fact, we can see from Fig. 4(a) that all the students’ models converge to the same MAE value, meaning that the federated students are effectively learning from each other in order to consolidate their models. To analyze the performance of our framework, we consider different learning scenarios, namely, FL, centralized learning (CL) which is the traditional training approach, and standalone learning (SL) where the student trains independently of its model. As such, in Fig. 4(b), a single data point per reference position is chosen to feed the trained models, and the predictions are compared to the ground-truth labels, and we reported the cumulative distribution function (cdf) of the corresponding

| Parameter | Description | Value |
|-----------|-------------|-------|
| Optimizer | DNN Model Optimizer | Adam (53) |
| \( \mu \) | Learning rate | 0.0001 |
| \( \beta_1, \beta_2 \) | Exponential decay rates | 0.1, 0.99 |
| \( N_1 \) | Input layer units | 10 |
| \( N_h \) | Hidden layer units | \( \times 1000 \) |
| \( N_o \) | Output layer units | 2 |
| \( \sigma_h(\cdot) \) | Hidden Activation function | ReLu |
| \( \sigma_o(\cdot) \) | Output Activation function | Linear |
| \( R \) | Number of students (clients) | 5 |
| \( \mathcal{R} \) | Batch size | 32 |
| \( \lambda \) | Communication rounds for FL and FD | 100 |

TABLE I

Parameters of the MLP Model

| Model | RMSE(m) | bits/round |
|-------|---------|------------|
| FD Model | 0.35 | 3200 |
| Standalone Model | 0.38 | - |
| FL Model | 0.21 | 2080320 |
| Central Model | 0.20 | - |

TABLE II

Network Global RMSE

RMSEs. We can see that the FD model improves the localization accuracy compared to the SL since the students share their knowledge with each other. Nonetheless, it remains less accurate than FL and CL due to the nature of its operating mode. It is important to note that this accuracy decrease is the price of a huge communication gain which is the principal goal of this work. Indeed, in terms of communication complexity as shown in Table II, with a bits resolution \( R = 32 \), the number of segments \( S = 10 \) and the output dimension \( N_o = 2 \) using \( K = 5 \) students, the FD model is far better than the FL. In fact, with FD, only \( 5 \times 640 = 3200 \) bits (29) are transmitted at each round in comparison to \( 5 \times 416064 = 2080320 \) bits (30) for FL, leading to a FD-to-FL ratio of 0.15%. Consequently, FD can save up to 99.85% of the transmission energy \( E_T \) used in FL while remaining only 1.6× less accurate than FL.

Furthermore, we can see from Fig. 4(c) how accurate is our localization framework by observing the estimation errors plotted therein. The network overall root mean squared error (RMSE) is computed for each model and the results are presented in Table II. These results attest that our framework is suitable for localization in indoor IoT networks since it presents relatively good accuracy regarding the geometry of the network and preserves network resources.

In summary, as outlined in Section III-F, the energy efficiency of our framework is closely tied to the communication load, as we have shown that FL and FD exhibit similar computational complexity. Given the substantial reduction in transmitted bits achieved by FD, the overall energy consumption in communication is significantly lower. This stands in stark contrast to FL, which necessitates the transmission of a considerable number of bits due to its larger model size. Through extensive simulations, the communication efficiency of FD over FL is determined to be 99.85%. Section IV-C aims to solidify these findings with experimental data.

C. Experimental Validation and Benchmark

In this section, we implement our framework using experimental localization data as well as other regression task data. First of all, we perform indoor localization using UJIIndoorLoc WiFi fingerprinting dataset [54] collected on a campus area of approximately \( 390 \times 270 \) m containing 21,048 entries. The results are presented in Fig. 5. Fig. 5(a) shows the convergence of all the MAE of all students to the same MAE value which confirms the convergence of the proposed FD algorithm. In Fig. 5(b), our proposed FD model is benchmarked with the newly introduced FL model in [55] in the context of indoor localization using the same dataset in [54]. Note that we used the same DNN model presented in [55] on all the datasets. In order to highlight the relevance of our FD
框架，我们保持显示结果的 standalone 模型（独立学习）以及无论可能的集中系统。结果表明，FD 学习略微提高估算精度相比其 standalone（基线）模型，而保持比 FL 更低的准确度。

FD 与 FL 在能量效率方面的优点见图 5(c)，图中比较了迭代通信的次数和传送的比特数。实际上，尽管需要几个轮次，FD 传送的比特数少于 FL 来达到相同的精度。由于计算能量比传输能量低，FD 消耗的能量显著低，这对于 IoT 系统非常有限的带宽优势。因此对于相同的定位精度，提出的 FD 框架带来了更大能量效率通过合理的节省传输能量。简单地说，除了 FD 小幅提高精度以外，框架的主要优点是通信效率，如表 III 所示。 FD 可以节省超过 98% 的传输比特数每个通信轮次，因此传输能量 

Moreover, we perform outdoor localization in an urban long range wide area network (LoRaWAN) using the dataset reported in [56] and diamond prices prediction using the dataset in [57]. We also present the results of the baseline models used in the original works where the datasets are presented. The results are summarized in Table III for the communication complexity and Table IV for the accuracy in terms of RMSE. In the UJIIndoorLoc experiment, we included the MAE as it is commonly used in existing works with this dataset. It can be seen from the results that the FD framework while sacrificing a small portion of its accuracy, achieves a remarkable improvement in communication efficiency. Note that for a fair comparison between different systems, the same number of epochs are used for standalone training as well as for centralized training considering the number of local epochs per round and the number of communication rounds in the federated settings. It is worth noting that for UJIIndoorLoc and LoRa datasets, the target variables have dimension 2 (2-D localization) while the diamond dataset is 1-D (the diamond’s price). The results assert that our framework can be used for any localization system in wireless networks and by extent, for any regression task since it has also shown great performance with the diamond pricing task which is a typical regression problem.

**D. Scaling Capability**

In this section, we explore the scalability potential of our framework by varying the number of students engaged in the learning process. Our objective is to investigate the impact of scale on accuracy and communication efficiency, as denoted by the RMSE and the number of transmitted bits \( N_{\text{bits}} \). The results are presented in Fig. 6.

Fig. 6(a) illustrates that as the number of students increases, both FL and FD systems experience improved accuracy, with a more pronounced effect observed in the FL system. This observation aligns with the inherent differences between FD and FL. In FD, students contribute to the learning process...
by sharing only their output, contrasting with FL where all model parameters are shared across the network. On the other hand, Fig. 6(b) provides insights into communication loads by depicting the total number of transmitted bits from all students to the server. This analysis underscores the advantages of FD, confirming its bandwidth conservation compared to FL. Notably, the results validate (29), demonstrating that FD is more bandwidth-efficient than FL. To further emphasize this efficiency, Fig. 6(c) presents the FD/FL ratios for accuracy and communication, revealing that FD transmits only 1.46% of the bits transmitted in FL. This indicates that FD is approximately ∼ 98, 54% more efficient than FL in terms of communication while maintaining accuracy levels only around 1× to 2.7× less precise. In summary, our proposed system exhibits strong scalability characteristics, showcasing improved accuracy with an increasing number of participants while ensuring remarkable efficiency in communication. These findings position our framework as a scalable solution that strikes a favorable balance between accuracy and communication efficiency.

E. Impact of Segmentation and Data Distribution

In this section, we analyze the impact of the segmentation on the system performance. Indeed, we consider two segmentation strategies as shown in Fig. 7(a). In the first strategy in Fig. 7(b), a uniform split which is so far used in this work and was explained in Fig. 1(b), we divide the output target into equal size segments referred to as intervals. So depending on the data distribution, some intervals may be empty or at least very less populated because the target value is far from uniformly distributed. Consequently, the private datasets of clients will only contain a subset of segments playing the role of labels here. This results in a missing label problem which needs to be properly mitigated depending on the dataset.

Alternatively, we design a second segmentation strategy in Fig. 7(c), a density-based split in which each segment/interval possesses the same number of samples, meaning that their sizes are necessarily different. This will consequently affect the accuracy of the models as shown in Fig. 8 so that in the end, the choice of a strategy relies on the data distribution. Moreover, intuitively, increasing the number of segments should improve the model accuracy but actually as shown in Fig. 8, it can in contrast degrade the model performance if the distribution impairments induced by the segmentation are not properly mitigated. The choice of a segmentation strategy holds practical implications for system performance, requiring thoughtful consideration of the underlying data distribution. It emphasizes the importance of addressing missing label issues and balancing segment sizes to maintain model accuracy. Furthermore, the counterintuitive impact of increased segments on model performance highlights the need for effective mitigation strategies to preserve overall system efficacy.
Fig. 7. Impact of segmentation on data distribution, UJIIndoorLoc dataset [54]. (a) Segmentation strategies. (b) Strategy 1. (c) Strategy 2.

Fig. 8. Impact of segmentation on system accuracy.

Fig. 9. Impact of the distillation regularizer on the FD system accuracy.

F. Impact of the Distillation Regularizer

The FD students learn from their teachers by distilling the teachers’ knowledge weighted by a regularizer. This regularizer is a hyperparameter of the FD system that plays an important role in how much the students learn from their teachers. Consequently, the choice of the regularizer directly impacts the students’ model accuracy. In Fig 9, we show the effect of the regularizer $\lambda$ in the FD-based indoor localization with the UJIIndoorLoc dataset. We can observe that for this application, the smaller the regularizer the better the accuracy. Thus, a good choice of $\lambda$ for this application has to meet the condition $\lambda < 1$. This result is in accordance with the theoretical analysis conducted in Section III-E, where it has been shown that the student prediction error increases with the value of $\lambda$ in the presence of an imperfect teacher model.

V. DISCUSSION AND FUTURE DIRECTIONS

This work addresses a critical gap in the field by introducing a novel methodology for federated KD in IoT networks, specifically tailored for regression problems. The simulation results and subsequent performance analysis serve as a foundational proof of concept for our proposed framework, showcasing its considerable potential across diverse applications, notably in the realm of IoT network localization. While the achieved performance of our framework is commendable, it does display certain shortcomings, providing opportunities for further refinement and enhancement, as elucidated below.

1) Unbalanced/ Missing Segments: Analogous to the issue of missing labels in classification tasks, the segmentation of target variables may result in unbalanced or missing data within local private datasets. This underscores the prevalence of non-IID data, necessitating the exploration of effective mitigation strategies. One plausible approach involves the incorporation of generative adversarial networks (GANs) to enhance data distribution.

2) Segmentation Strategy: The efficacy of an FL system is contingent upon data distribution, a factor significantly influenced by the employed segmentation strategy. An imperative area for further investigation lies in the development of a robust segmentation methodology to optimize the performance of the proposed framework.

3) Proxy Data: KD within our framework encounters challenges due to the inherently private nature of local data. The availability of public datasets could facilitate a more seamless distillation process, thereby augmenting system performance. Alternatively, the generation of synthetic data copies using techniques like GANs can be explored.

4) Error Prediction: The predictability of estimation errors, given a segmentation strategy, holds the potential for substantial system performance enhancement.
Mechanisms to forecast and adjust estimations based on error predictions warrant further investigation.

5) FL/FD Mix-Up in Heterogeneous IoT Networks: In wireless networks, the substantial asymmetry between uplink and downlink communication presents challenges for IoT devices, especially on the uplink. Exploring this asymmetry, a hybrid approach could be examined, where FD is implemented on the challenging uplink to reduce communication costs, while the less demanding downlink employs FL. Given the prevalent heterogeneity in IoT networks, an intricate strategy could be explored—utilizing FL in both uplink and downlink for high-capability devices, while reserving full FD for the remaining devices.

This comprehensive exploration of future directions aims to address current limitations and refine the proposed FD framework for regression problems in IoT networks.

VI. CONCLUSION

In this study, we introduced an innovative FD framework tailored for regression problems, specifically applied to the development of an indoor localization system. Our approach demonstrated remarkable results, showcasing substantial benefits in terms of bandwidth and energy savings when compared to localization systems based on FL. The framework’s novelty and communication efficiency are key strengths, evident in its ability to reduce communication loads by approximately 98% compared to FL frameworks. Beyond communication efficiency, our proposed framework exhibits significant scalability, rendering it well-suited for large IoT networks. However, despite its impressive performance, there exist areas for further enhancement. Challenges associated with segmentation and the inherent heterogeneity of IoT devices, particularly in terms of computational and communication capabilities, represent notable limitations that warrant attention for future refinements.

REFERENCES

[1] A. M. Research, “Location-based services market to reach $318.64 billion in 2030,” in GlobeNewswire News Room, Wilmington, NC, USA: Allied Market Research, Nov. 2020.

[2] S. Li, L. D. Xu, and S. Zhao, “5G Internet of Things: A survey,” J. Ind. Inf. Integr., vol. 10, pp. 1–9, Jun. 2018.

[3] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, “The road towards 6G: A comprehensive survey,” IEEE Open J. Commun. Soc., vol. 2, pp. 334–366, 2021.

[4] M. Jouhari, N. Saeed, M.-S. Alouini, and E. Mehdi Amhoud, “A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges,” 2022, arXiv:2202.11082.

[5] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” IEEE Netw., vol. 34, no. 3, pp. 134–142, May 2020.

[6] M. Naby Ndiaye, E. Houcine Bergou, and H. El Hammouti, “Muti-hop localization algorithm based on NSGA-II in Internet of Things,” Mathematics, vol. 7, no. 2, p. 184, Feb. 2019.

[7] C. De Lima et al., “Convergent communication, sensing and localization in 6G systems: An overview of technologies, opportunities and challenges,” IEEE Access, vol. 9, pp. 26902–26925, 2021.

[8] M. Delamou, G. Noubir, S. Dang, and E. Mehdi Amhoud, “An efficient OFDM-based monostatic radar design for multitarget detection,” IEEE Access, vol. 11, pp. 135090–135105, 2023.

[9] M. Delamou, A. Bazzi, M. Chafii, and E. Mehdi Amhoud, “Deep learning-based estimation for multitarget radar detection,” in Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring), Jun. 2023, pp. 1–5.

[10] L. Ru and L. Zhang, “A weighted centroid localization algorithm for wireless sensor networks based on weight correction,” in Proc. 9th Int. Conf. Adv. Infocomm Technol. (ICAIT), Nov. 2017, pp. 165–169.

[11] P. Wang, F. Xue, H. Li, Z. Cui, and J. Chen, “A multi-objective DV-hop localization algorithm based on NSGA-II in Internet of Things,” Mathematics, vol. 7, no. 2, p. 184, Feb. 2019.

[12] X. Tong, Y. Wan, Q. Li, X. Tian, and X. Wang, “CSI fingerprinting localization with low human efforts,” IEEE/ACM Trans. Netw., vol. 29, no. 1, pp. 372–385, Feb. 2021.

[13] S. Wang, X. Jiang, and H. Wymeersch, “Cooperative localization in wireless sensor networks with AOA measurements,” IEEE Trans. Wirel. Commun., vol. 21, no. 8, pp. 6760–6773, Aug. 2022.

[14] S. Wu, S. Zhang, and D. Huang, “A TOA-based localization algorithm with simultaneous NLOS mitigation and synchronization error elimination,” IEEE Sensors Lett., vol. 3, no. 3, Mar. 2019, Art. no. 6000504.

[15] Y. Etiabi, E. M. Amhoud, and E. Sabir, “A distributed and collaborative localization algorithm for Internet of Things environments,” in Proc. 18th Int. Conf. Adv. Mobile Comput. Multimedia, 2020, pp. 114–118.

[16] B. Jin, X. Xu, and T. Zhang, “Robust time-difference-of-arrival (TDOA) localization using weighted least squares with cone tangent plane constraint,” Sensors, vol. 18, no. 3, p. 778, Mar. 2018.

[17] S. Yiu, M. Dadhtli, H. Clausen, and F. Perez-Cruz, “Wireless RSSI fingerprinting localization,” Signal Process., vol. 131, pp. 235–244, Feb. 2017.

[18] Y. Zeng, X. Chen, R. Li, and H.-Z. Tan, “UHF RFID indoor positioning system with phase interference model based on double tag array,” IEEE Access, vol. 7, pp. 76768–76778, 2019.

[19] M. Di, O. A. Postolache, C. Mi, M. Zhong, and Y. Wang, “UWB indoor positioning application based on Kalman filter and 3-D TOA localization algorithm,” in Proc. 11th Int. Symp. Adv. Topics Elect. Eng. (ATEE), Mar. 2019, pp. 1–6.

[20] R. Joseph and S. B. Sasi, “Indoor positioning using WiFi fingerprint,” in Proc. Int. Conf. Circuits Syst. Digit. Enterprise Technol. (ICCSDET), Dec. 2018, pp. 1–3.

[21] Y. Etiabi, M. Jouhari, A. Burg, and E.-M. Amhoud, “Spread factor and RSSI for localization in LoRa networks with deep reinforcement learning approach,” in Proc. IEEE 97th Veh. Technol. Conf. (VTC-Spring), pp. 1–5, Jun. 2023.

[22] A. Satan and Z. Toth, “Development of Bluetooth based indoor positioning application,” in Proc. IEEE Int. Conf. Future IoT Technol. (Future IoT), Jan. 2018, pp. 1–6.

[23] Y. Etiabi, E. Amhoud, and E. Sabir, “Huber estimator and statistical bootstrap based light-weight localization for IoT systems,” in Proc. 7th Int. Symp. Ubiquitous Netw. (UNet), vol. 12845, 2020, pp. 79–92.

[24] A. Nessa, B. Adhihaki, F. Hussain, and X. N. Fernando, “A survey of machine learning for indoor positioning,” IEEE Access, vol. 8, pp. 214945–214965, 2020.

[25] S. R. Jondhale, M. A. Wakechaure, B. S. Agarkar, and S. B. Tambe, “Improved generalized regression neural network for target localization,” Wireless Pers. Commun., vol. 125, no. 2, pp. 1677–1693, Mar. 2022.

[26] P. Ferrand, A. Decuminge, and M. Guillaud, “DNN-based localization from channel estimates: Feature design and experimental results,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2020, pp. 1–6.

[27] S. R. Jondhale, A. S. Jondhale, P. S. Deshpande, and J. Lloret, “Improved triartulation for indoor localization: Neural network and centroid-based approach,” Int. J. Distrib. Sensor Netw., vol. 17, no. 11, 2021, Art. no. 15501477211053997. [Online]. Available: https://api.semanticscholar.org/CorpusID:244110064

[28] B. McMahan et al., “Communication-efficient learning of deep networks from decentralized data,” in Proc. 97th Veh. Technol. Conf. (VTC-Spring), Jun. 2023, pp. 1–5.

[29] D. Ni, O. A. Postolache, C. Mi, M. Zhong, and Y. Wang, “CSI fingerprinting localization,” Sensors, vol. 19, no. 1, pp. 372–385, Feb. 2019.

[30] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Deep federated localization with low human efforts,” IEEE/ACM Trans. Netw., vol. 29, no. 4, pp. 2025–2038, Jul. 2022.
[33] W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue, “Deep CNN for indoor localization in IoT-sensor systems,” Sensors, vol. 19, no. 14, p. 3127, Jul. 2019.

[34] I. B. F. de Almeida, M. Chafii, A. Nimr, and G. Fettweis, “Blind transmitter localization in wireless sensor networks: A deep learning approach,” in Proc. IEEE 32nd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC), Sep. 2021, pp. 1241–1247.

[35] Y. Liu, H. Li, J. Xiao, and H. Jin, “Floc: Fingerprint-based indoor localization system under a federated learning updating framework,” in Proc. 15th Int. Conf. Mobile Ad-Hoc Sens. Netw. (MSN), 2019, pp. 113–118.

[36] Y. Etiabi, W. Njima, and E. M. Amhoud, “Federated learning based hierarchical 3D indoor localization,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Mar. 2023, pp. 1–6.

[37] Z. Wu, X. Wu, and Y. Long, “Prediction based semi-supervised online personalized federated learning for indoor localization,” IEEE Sensors J., vol. 22, no. 11, pp. 10640–10654, Jun. 2022.

[38] 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.

[39] B. Sait Ciftler, A. Albaseer, N. Lasla, and M. Abdallah, “Federated learning for localization: A privacy-preserving crowdsourcing method,” 2020, arXiv:2001.01911.

[40] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, “Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas,” Datu, vol. 3, no. 2, p. 13, 2018.

[41] F. Sattler, A. Marban, R. Rischke, and W. Samek, “Communication-efficient federated distillation,” 2020, arXiv:2012.00632.

[42] A. Rahbar, A. Panahi, C. Bhattacharyya, D. Dubhashi, and M. H. Chehreghani, “Analysis of knowledge transfer in kernel regime,” 2020, arXiv:2003.13438.

[43] K. H. Rosen, Discrete Mathematics and Its Applications, 7th ed. New York, NY, USA: McGraw-Hill, 2012.

[44] J.-H. Ahn, O. Simeone, and J. Kang, “Wireless federated distillation for distributed edge learning with heterogeneous data,” in Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC), Sep. 2019, pp. 1–6.

[45] J.-H. Ahn, O. Simeone, and J. Kang, “Cooperative learning VIA federated distillation OVER fading channels,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 8856–8860.

[46] T. M. Cover and J. A. Thomas, Elements of Information Theory. Hoboken, NJ, USA: Wiley, Apr. 2005, ch. 9, pp. 261–299.

[47] W. Njima, M. Chafi, A. Chorti, R. M. Shubair, and H. V. Poor, “Indoor localization using data augmentation via selective generative adversarial networks,” IEEE Access, vol. 9, pp. 98337–98347, 2021.

[48] W. Njima, M. Chafi, A. Nimr, and G. Fettweis, “Deep learning based data recovery for localization,” IEEE Access, vol. 8, pp. 175741–175752, 2020.

[49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015.

[50] T. M. Cover and J. A. Thomas, Elements of Information Theory. Hoboken, NJ, USA: Wiley, Apr. 2005, ch. 9, pp. 261–299.

[51] J. Park et al., “Federated learning for indoor localization via model reliability with dropout,” IEEE Commun. Lett., vol. 26, no. 7, pp. 1553–1557, Jul. 2022.

[52] D. Gufran and S. Pasricha, “FedHIL: Heterogeneity resilient federated learning for robust indoor localization with mobile devices,” ACM Trans. Embedded Comput. Syst., vol. 22, no. 5s, pp. 1–24, Oct. 2023.

[53] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in Proc. NIPS Deep Learn. Represent. Learn. Workshop, 2015, pp. 1–9.

[54] R. Anil, G. Pereyra, A. Passos, R. Ormandi, G. E. Dahl, and G. E. Hinton, “Large scale distributed neural network training through online distillation,” in Proc. Int. Conf. Learn. Represent., 2018, pp. 1–12.

[55] H. Seo, J. Park, S. Oh, M. Bennis, and S.-L. Kim, “Federated knowledge distillation,” 2020, arXiv:2011.02367.

[56] A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: Convergence and generalization in neural networks,” in Proc. 32nd Int. Conf. Neural Inf. Process. Syst., 2018, pp. 8580–8589.

[57] F. Sattler, A. Marban, R. Rischke, and W. Samek, “Communication-efficient federated distillation,” 2020, arXiv:2012.00632.

[58] A. Rahbar, A. Panahi, C. Bhattacharyya, D. Dubhashi, and M. H. Chehreghani, “Analysis of knowledge transfer in kernel regime,” 2020, arXiv:2003.13438.

[59] J.-H. Ahn, O. Simeone, and J. Kang, “Wireless federated distillation for distributed edge learning with heterogeneous data,” in Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC), Sep. 2019, pp. 1–6.

[60] J.-H. Ahn, O. Simeone, and J. Kang, “Cooperative learning VIA federated distillation OVER fading channels,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2020, pp. 8856–8860.

[61] T. M. Cover and J. A. Thomas, Elements of Information Theory. Hoboken, NJ, USA: Wiley, Apr. 2005, ch. 9, pp. 261–299.

[62] W. Njima, M. Chafi, A. Chorti, R. M. Shubair, and H. V. Poor, “Indoor localization using data augmentation via selective generative adversarial networks,” IEEE Access, vol. 9, pp. 98337–98347, 2021.

[63] W. Njima, M. Chafi, A. Nimr, and G. Fettweis, “Deep learning based data recovery for localization,” IEEE Access, vol. 8, pp. 175741–175752, 2020.

[64] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015.

[65] 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.

[66] B. Sait Ciftler, A. Albaseer, N. Lasla, and M. Abdallah, “Federated learning for localization: A privacy-preserving crowdsourcing method,” 2020, arXiv:2001.01911.

[67] M. Aernouts, R. Berkvens, K. Van Vlaenderen, and M. Weyn, “Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas,” Datu, vol. 3, no. 2, p. 13, 2018.

[68] S. Agrawal. Analyze Diamonds by Their Cut, Color, Clarity, Price, and Other Attributes. Accessed: Jan. 30, 2022. [Online]. Available: https://www.kaggle.com/shivam2503/diamonds

Yaya Etiabi (Graduate Student Member, IEEE) received the engineering degree in embedded systems and telecommunications from the National School of Electricity and Mechanics of Casablanca, Hassan II University, Casablanca, Morocco, in 2020. He is currently pursuing the Ph.D. degree with Mohammed VI Polytechnic University, Ben Guerir, Morocco.

He is actively involved in research, with a focus on wireless indoor positioning, and has delivered numerous talks at conferences and seminars. With a keen interest in the intersection of indoor localization, wireless communications, and advanced machine learning, he is committed to advancing the understanding of wireless positioning technologies throughout his academic journey and beyond.

El Mehdi Amhoud (Member, IEEE) received the Ph.D. degree in computer and communications science from Télécom ParisTech, Paris, France, in 2017.

He is currently an Assistant Professor with Mohammed VI Polytechnic University, Ben Guerir, Morocco. Prior to his current position, he was a Post-Doctoral Research Fellow with the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia. He holds several U.S. patents. His research interests include modeling and analyzing the performance of new generations of communication networks and the Internet of Things.

Dr. Amhoud received two awards of excellence for his outstanding Ph.D. thesis from the Mines-Télécom Institute, Paris, and the Marie Sklodowska-Curie Research Grant from the European Commission.