IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Radio Environment Map Construction Based On Privacy-centric Federated Learning

SHAFI ULLAH KHAN[1], CARLA E. GARCÍA[2], TAEWOONG HWANG[1], and INSOO KOO[1]
[1]Department of Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan 680-749, South Korea
[2]Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, Luxembourg.

Corresponding author: Insoo Koo (e-mail: iskoo@ulsan.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) through the Korean Government's Ministry of Science and ICT (MSIT) under Grant NRF-2021R1A2B5 B01001721, and in part by the Regional Innovation Strategy (RIS) through the NRF funded by the Ministry of Education (MOE) under Grant 2021RIS-003.

ABSTRACT In today's digital age, coverage prediction is essential for optimizing wireless networks and improving user experience. While numerous path loss models and advanced machine learning algorithms have been developed to achieve high prediction performance, they predominantly operate within a centralized learning paradigm. While effective, this conventional approach often suffers from scalability and privacy limitations that are critical to the successful deployment of wireless maps. Conversely, in this paper, we propose a novel decentralized approach based on a federated learning long short-term memory (LSTM) model to accurately predict network coverage in indoor environments. The proposed FedLSTM is a method that allows multiple users, or clients, to train the model without sharing their personal data directly with a central server. In an experimental setup, we used real data collected from numerous clients moving along different paths. The FedLSTM model is evaluated in terms of root mean square error (RMSE), mean absolute error (MAE), and R2. Furthermore, compared to a centralized counterpart, FedLSTM shows a slight increase in RMSE from 2.4 dBm to 2.5 dBm and an increase in MAE from 1.7 dBm to 1.9 dBm. In addition, we evaluate the proposed FedLSTM considering variations in the number of participating clients and the number of local training epochs. The results show that even devices with limited computational power can meaningfully contribute to the training of the federated model, with fewer epochs achieving competitive results. Graphical analyses of the radio environment maps (REMs) generated by both FedLSTM and the centralized LSTM highlight their similarities. However, FedLSTM provides client privacy while reducing communication overhead and server strain.

INDEX TERMS radio environment map (REM), coverage prediction, received signal strength indicator (RSSI), LiDAR sensor, federated learning.

I. INTRODUCTION

IN the context of the Fourth Industrial Revolution, awareness of operating environment conditions is becoming increasingly important for the efficient management of resources in diverse dynamic systems. This era, marked by the emergence of advanced technologies such as artificial intelligence, big data, the Internet of Things (IoT), and robotics, has significantly transformed modern industrial ecosystems, especially smart factories. Our research is strongly motivated by the urgent need to address the complex and multifaceted challenges of ensuring robust, efficient, and secure wireless connectivity in such environments. Smart factories, which are not only hubs of innovation but also arenas where the reliability and precision of wireless communications are rigorously tested, have leveraged these technologies to automate production processes. This automation facilitates automatic assessment of process status, enabling timely intervention and improving overall operational efficiency [1].

Central to the functionality of smart factories is wireless communications, which assumes a pivotal role in empowering instantaneous tracking and surveillance of production processes. Moreover, the flexibility and simplicity of wireless communications make it the preferred mode of connectivity in the dynamic landscape of production environments, compared to cumbersome wired alternatives. However, the surging count of wireless devices introduces a challenge—potential interference with the industrial, scientific, and medical (ISM) band [2]. Moreover, obstacles within indoor environments can attenuate communication signals, leading to regions of radio shadow. Accordingly, a precise

assessment of the extent of radio communication coverage emerges as a priority. In addressing the aforementioned challenge, radio environment map (REM) construction has been investigated as an innovative tool that serves to furnish intricate details about the radio environment within specific geographic areas. By harnessing the insights provided by the REM, informed decision-making is facilitated, and network operators can seamlessly identify coverage gaps and high-traffic regions [3].

In the literature, path loss models have been investigated for coverage prediction. However, these models depend on various factors, such as the distance between transmitter and receiver as well as the height of the receiver and transmitter above ground, which increases the difference in error prediction between real and estimated values [4]. Hence, machine learning (ML)-based approaches have emerged as innovative predictive methods capable of effectively addressing the intricate operational challenges within communication networks. These techniques have demonstrated a remarkable ability to achieve high prediction accuracy [5], [6]. For example, the authors in [7] introduced a three-level Reconfigurable Intelligent Surface (RIS) framework to improve the signal quality of wireless communications, focusing on efficient channel state information (CSI) acquisition with low latency and pilot overhead. It uses a sparse connected long short-term memory (SCLSTM) neural network to decompose and predict the dynamic channels between base stations and user equipment. This approach significantly outperforms traditional channel estimation methods in terms of accuracy and robustness. Moreover, researchers have projected the path loss in an urban setting in Beijing, China, when utilizing artificial neural networks (ANNs), support vector regression (SVR), and random forest (RF) models [8]. Assessments of performance were measured through root mean square error (RMSE), yielding results ranging between 4 dB and 5 dB. In [9],the authors proposed the extra tree regressor-based approach for REM construction in wireless communications networks for indoor environments. The results showed that the extra tree regressor can obtain the best accuracy with less computational time than other ensemble learning baseline schemes. To the best of our knowledge, the researchers described above considered a centralized manner where the learning process is managed by a central server or a base station. In this paper, we propose a federated learning (FL) approach called FedLSTM, which works with a long short-term memory (LSTM) model to provide distributed learning between users. Note that in the conventional centralized approach, more data are transmitted since both the features and the labels must be sent. On the other hand, in a distributed manner, the user only sends the weights of the local model that are entailed in computation.

Moreover, the proposed FedLSTM scheme allows both server and users to generate the REM. Then, the server can be considered network planning to obtain a REM that can solve coverage problems (installing APs or relays). Meanwhile, users can more fully appreciate coverage of the area, and can redirect to a better coverage area by looking at the REM. In addition, the proposed FL-based approach provides security because the data sent to the server are the weights, not the labels and features of each user. This scenario is very useful in commercial, hospital, and military environments where users do not want to share their location with unknown people, thus guaranteeing privacy and security.

We propose a novel FL-based approach to coverage prediction in indoor environments that not only minimizes data transmission but also enables network planning and empowers users to make informed decisions about their coverage. Additionally, it prioritizes user privacy and security, making it highly applicable in various sensitive settings

The main contributions of this paper can be summarized as follows.

- First, we propose a novel FL-based approach, called FedLSTM, providing coverage prediction for indoor environments. FedLSTM enables distributed model training by having users send only model weights to the server. This is in stark contrast to centralized approaches where users must transmit both features and labels, leading to a significant increase in data transmissions.
- Secondly, the data utilized in this study were collected from a real environment with location points captured using Emesent's Hovermap and real received signal strength indicator (RSSI) values obtained via Raspberry Pi. After collection, we preprocess the location data with Emesent's software and synchronize it with the cleaned RSSI readings from Raspberry Pi by using timestamps.
- Third, we construct a REM by using Python software to enhance coverage prediction visualization. For this objective, we generate a grid of data comprising 1000 $\times$ 1000 grid points within our area of interest to plot coverage prediction over a 2D map.
- Furthermore, our FL-based scheme empowers both server and users to generate the REM. The server functions as the network planner, leveraging the REM to address coverage issues by installing access points or relays. On the other hand, users can assess the coverage of their area and relocate to areas with better coverage by examining the REM.
- In addition, we compared the FedLSTM model with its centralized counterpart, showing that our research ensures security by transmitting only the weights of each user's model to the server, while the labels and features remain private. This approach is particularly advantageous in commercial, hospital, and military settings where users are hesitant to share their location data with unknown entities.

The remainder of this paper is organized as follows. Section II describes related work, and Section III describes data collection and preprocessing. Section IV outlines overall system model, including the FL scheme, the FedLSTM architecture, and REM construction. Section V presents numerical results and a computational complexity analysis, with visual findings

presented in Section VI. Finally, conclusions are in Section VII.

II. RELATED WORK

Radio maps (commonly known as REMs) and their construction play a very important role in modern communication systems. These maps offer a comprehensive view of the radio spectrum environment by retaining various types of information, from geographic and land features to spectrum usage characteristics. Over the years, a significant amount of research has been conducted to enhance and diversify their applications. Introduced in 2006 [10], REMs have facilitated a multitude of applications, ranging from network monitoring [11], localization [12] and resource management [13] to V2X communication [14], [15].

Traditional methods of radio map construction often involve detailed field surveys, which can be time-consuming and labor-intensive. To address this, researchers have been developing algorithms to lower these costs. A number of path loss models have been influenced by factors like terrain suitability, the heights of the receiver and transmitter above ground level, their spatial separation, and the presence of intervening obstructions, among other things [16]. These elements can widen the gap between forecast and real signal degradation, with the extent of the variance hinging on the chosen propagation model. In the REM construction literature, ordinary kriging (OK) is frequently employed as a geostatistics-based spatial interpolation method [17], [18]. OK predicts unseen data points by considering the spatial relationships between recorded data and the relative locations of all sampled points [19].

In [20] Maiti and Mitra developed a radio map for indoor signal propagation, leveraging interpolation methodologies. Results showed that OK achieved better performance than ordinary methods like inverse distance weighting [21] and K-nearest neighbors (KNN). These methods were evaluated based on prediction error, i.e., RMSE. Although the OK-based model achieved better performance, it encounters a limitation in its computational proficiency, especially with more data points [18].

In [22], and [23], heuristic-derived methodologies were introduced providing indoor coverage prediction for indoor dominant path models. However, heuristic solutions are typically designed for specific problems, and might not generalize well to other scenarios or variations of the problem, making them inconsistent and unreliable in critical applications.

In classical prediction models, the design of mobile-device networks demonstrates an inherent lack of adaptability [24]. Predictions are constrained to specific conditions, such as frequency range, antenna height, and surrounding environmental conditions. Nonetheless, current observations indicate that the operational environment of modern radio networks is characterized by an elevated level of diversity and complexity [25]. Consequently, there is a strong need for prediction models that are more flexible and that can handle the challenges of modern networks.

ML-based prediction techniques are recognized as revolutionary within the realm of modern mobile-device network planning owing to their enhanced accuracy over age-old empirical prediction methods. When compared with deterministic-based models, the ML-based methods are notably superior in their data processing efficacy [26], [27].

For instance, the authors in [28] conducted research in an urban area of Lisbon, Portugal, at 3.7 GHz and 26 GHz frequency bands, leveraging an authentic 5G network. They utilized input factors, and the resultant values closely matched those in [29]. However, the dataset utilized was double in size. The study mainly focused on SVR and RF models, which showed error rates ranging between 6 dB to 7 dB.

Similarly, the authors in [30] conducted their research in suburban areas of South Korea, focusing on frequency bands of 450 MHz, 1450 MHz, and 2300 MHz. Although the input parameters were mostly similar to previous studies, a new parameter was introduced: the ratio between Tx height and Rx height. That research exclusively employed the Artificial Neural Network (ANN) and Gaussian Process Regression (GPR) ML models, both of which exhibited RMSE values ranging from 8 dB to 9 dB.

In addition, the authors in [31] evaluated ML models (ANN, SVR, and RF) in a rural environment in Greece in the 3.7 GHz band. Input parameters were 3D Tx-Rx distance, heights above sea level, and signal propagation (LOS/NLOS). The target was path loss, and RMSE ranged between 4 dB to 5 dB.

Lastly, the authors in [32] introduced a new approach using the extremely randomized trees regressor (ERTR) algorithm for mobile coverage prediction, and visualized results on a REM overlaid on top of Google Earth. Real measurement data from Victoria Island and Ikoyi in Lagos, Nigeria, were used. Through extensive simulations and comparisons with seven other ML algorithms, including ordinary kriging, the ERTR algorithm showed the lowest RMSE error at 2.75 dB with an R^2 score of 92%.

To the best of our knowledge, commonly employed methods for constructing radio maps heavily rely on centralized data approaches. While these methods are comprehensive, they bring forth computational complexities and potential vulnerabilities, particularly concerning user-specific data such as geospatial information. In environments such as commercial complexes, healthcare institutions, and military facilities, the adoption of a centralized approach is considered suboptimal. In these contexts, users place a high premium on privacy and security, necessitating measures to prevent the disclosure of location information to unauthorized parties. Our model incorporates the FL approach, advocating decentralized data processing across user nodes. Unlike the centralized framework, which necessitates transmission of both feature vectors and labels, the federated approach entails only transmission of model weight vectors. This reduction in data transmission overhead simultaneously enhances data security for users, with a minimal increase in prediction error.

III. DATA COLLECTION AND PREPROCESSING

In this study, we sourced our primary data from the Engineering Building at the University of Ulsan in South Korea. For the collection process, we systematically employed two key devices: the Emesent Hovermap and Raspberry Pi. The Emesent Hovermap, equipped with a state-of-the-art LiDAR sensor, operates using the simultaneous localization and mapping (SLAM) technique. It functions by emitting laser pulses and keenly noting the duration taken for these beams to bounce back after reflecting off surfaces. In relation to the speed of light, the LiDAR sensor can precisely calculate distances from the reflection time. Consequently, it creates an intricate 3D map based on these light reflections. After data collection, the raw data from Hovermap is processed and refined using the specialized Emesent software, making it fit for analytical use.

We used the received signal strength indicator as a metric to evaluate the quality of the connection between transmitting and receiving devices. In wireless systems, including Wi-Fi, Bluetooth, and cellular networks, RSSI is commonly used to measure the strength of a radio signal. These values in decibel milliwatts (dBm) are systematically obtained using the built-in Wi-Fi module in Raspberry Pi. The RSSI data is then pre-processed to remove any NaN values and outliers. The location data from the Emesent Hovermap and the RSSI data from Raspberry Pi are then synchronized using timestamps in Python to create a time series dataset. This synchronization aligns spatial coordinates with corresponding RSSI values over time, which is essential for LSTM model training and accurate REM construction in our federated LSTM approach.

IV. SYSTEM MODEL

Our experimental setup consists of 16 different clients that participate in model updates and a server that collects the updated models. Each of these clients traveled through 10 unique paths, updating its model with each trip. Each new data collection was used by the client to train its local model, adhering to predefined model and hyperparameter specifications. Upon completion of the training phase for a specified number of epochs, the local model weights are transmitted to a central server. At this central node, they are aggregated using the FedAvg to update the global model weight. This iterative process of individual learning and centralized aggregation was performed across all clients over 10 communication rounds.

A. FEDERATED LEARNING

FL is a novel approach to ML where a model is trained across multiple devices or servers while keeping the data localized. Instead of transferring raw data to a central server for training, FL pushes the model to edge devices (smartphones, tablets, IoT devices, etc.) and allows training to happen locally on each device. After local training, only the model updates are sent to the central server where they are aggregated and the global model is updated. Figure 1 shows the overall FL process. This approach addresses multiple concerns, such as privacy, security, and data ownership, while ensuring a quality model.

In the context of wireless communications, consider a system with N clients where each client, j, has its own distinct local dataset. The FL procedure begins with the central server initializing the global model. Let the parameters of this model be denoted by ω. For the i^{th} iteration of the FL process, the central server shares the current global model parameters, ω_i, with all N clients. Subsequently, each client j utilizes its dataset to conduct local training. Through this, it computes an updated version of the model parameter, which we label $\omega_{i,j}$. After completion of this local training phase, each client's updates are sent back to the central server. The server then aggregates these updates using Federated Averaging (FedAvg) shown in Algorithm 1. The aggregated update for the i^{th} iteration is described by the following equation:

$$\omega_{i+1} = \frac{1}{n} \sum_{j=1}^{N} \omega_{i,j} \tag{1}$$

Consensus in FL is critical to ensure that the global model accurately reflects the collective learning of all distributed clients. It helps to synchronize model updates from different data sources, thus maintaining the integrity and relevance of the model [33]. In our FL model, consensus is achieved by defining ϵ as the number of local epochs after which clients synchronize their updates. Each client j trains its local model for ϵ epochs and sends the updates to the server. The server aggregates these updates after receiving them from all clients. This aggregated model is then redistributed to a new set of clients for further training. This iterative process continues until global convergence is achieved. The main advantages of FL in the context of wireless communications include preserving the privacy of user data, reducing the communication load of transmitting large data sets, and taking advantage of distributed data resources.

Evaluating the computational complexity of federated learning algorithms is essential for assessing their scalability and resource requirements. This analysis focuses on the complexity of a specific federated learning process. The process involves I communication rounds, where in each round, where in each round N clients perform local training. Assuming that each client's training has a complexity of $O(L)$, the complexity per round is $O(N \times L)$. Therefore, the total complexity of the algorithm over I, rounds is $O(i \times N \times L)$, highlighting the interplay between the number of rounds, the number of clients involved, and the individual training complexity. This insight is crucial for optimizing federated learning systems for efficiency and practicality.

B. FEDLSTM ARCHITECTURE

LSTM is a specialized type of recurrent neural network (RNN) designed to remember and utilize information over extended sequences. This capability addresses the vanishing gradient problem commonly observed in traditional RNNs, enabling LSTM to learn and retain long-term dependencies

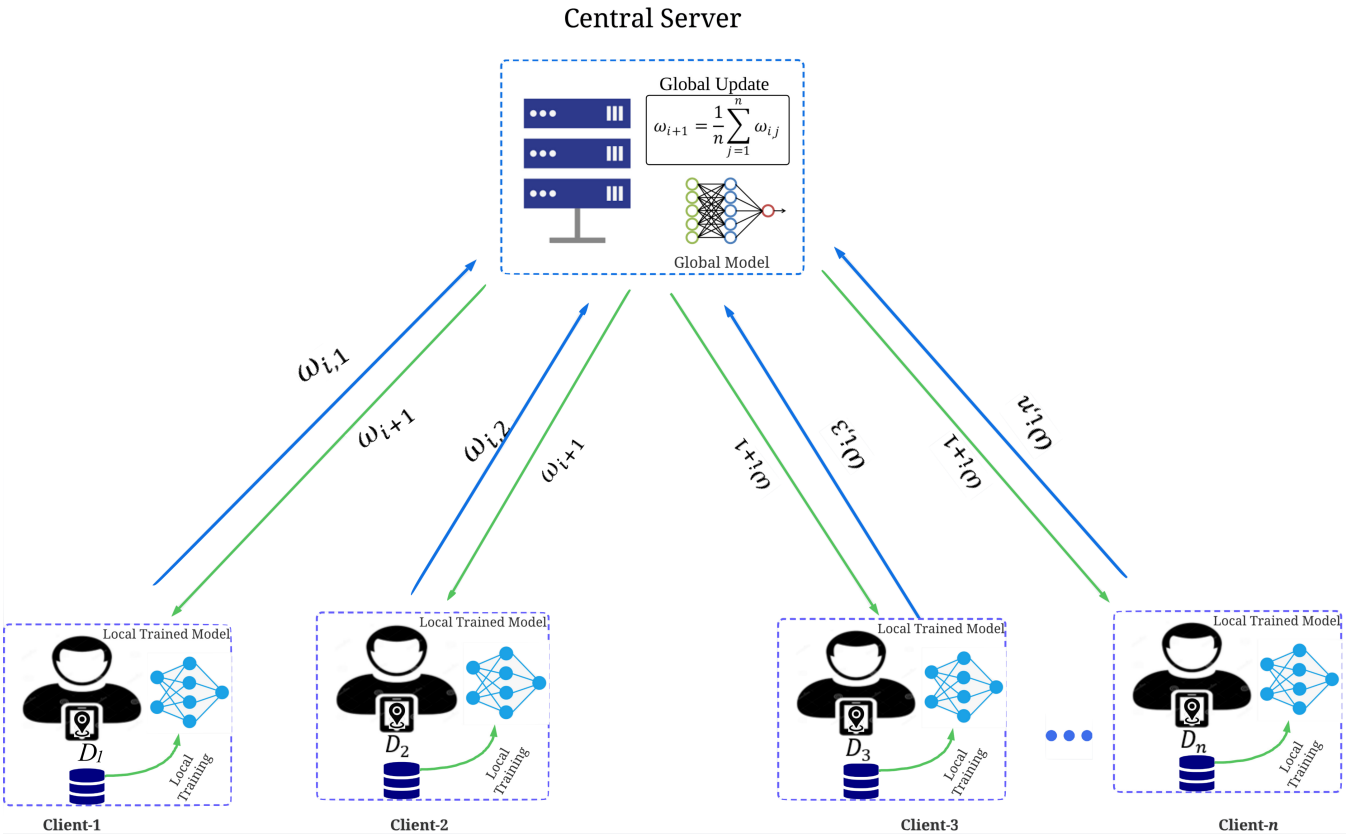


FIGURE 1. **Federated learning overview**

Algorithm 1 Federated Learning Process with FedAveraging

Initialize global model parameters ω on central server
for round $= 1$ to i **do** ▷ Repeat for i communication rounds
  selected_clients $\leftarrow$ Randomly select N clients
  for each client j in selected_clients **do**
    Send current model parameters ω_i to client j
    $\omega_{i,j} \leftarrow$ client j performs local training using its data
    Send $\omega_{i,j}$ back to central server
  end for
  $\omega_{i+1} \leftarrow 0$
  for each client j in selected_clients **do**
    $\omega_{i+1} = \omega_{i+1} + \omega_{i,j}$
  end for
  Apply FedAvg:
  $\omega_{i+1} = \omega_{i+1}/N$ ▷ Average the updates
  Send updated model parameters ω_{i+1} to all selected_clients
end for

in the data. Often utilized in time series prediction and other sequential tasks, LSTM significantly improves the efficiency and accuracy of deep learning models dealing with sequential data. RSSI prediction based on x and y geographic coordinates, can be done by the LSTM model. We train the LSTM model in a federated manner at the edge, and update the global model on the server. This proposed federated LSTM is called FedLSTM owing to the nature of its training. At first, the global model is initialized with random weights, and is then updated by the clients locally. The layer architecture starts with an LSTM layer of 256 units, designed to process spatial input and extract pertinent sequential patterns. To avoid overfitting, a subsequent dropout layer is employed, ensuring the model remains generalizable across diverse radio conditions. The succeeding layers (LSTM with 128 units and 64 units) progressively refine these patterns with intermittent dropout layers for regularization. The architecture ends with a pair of dense layers; the first handles the processed data, and the subsequent layer outputs the predicted RSSI value. The detailed, layer-by-layer architecture is described in Table 1.

What sets this research apart is its innovative adoption of federated learning. Rather than centralized model training, the architecture is distributed across multiple clients. Each client refines the model using local data before sending weight updates to a central server. This decentralized approach not only accentuates data privacy but also captures the diverse radio environments each client is exposed to, ensuring a more comprehensive and applicable real-world REM.

TABLE 1. Architecture of the LSTM Network

Layer (type)	Output Shape	Params	Description
lstm_3 (LSTM)	(None, 3, 256)	265,216	LSTM layer
dropout_3 (Dropout)	(None, 3, 256)	0	Regularization layer
lstm_4 (LSTM)	(None, 3, 128)	197,120	LSTM layer
dropout_4 (Dropout)	(None, 3, 128)	0	Regularization layer
lstm_5 (LSTM)	(None, 64)	49,408	LSTM layer
dropout_5 (Dropout)	(None, 64)	0	Regularization layer
dense_2 (Dense)	(None, 32)	2,080	Fully connected layer
dense_3 (Dense)	(None, 1)	33	Fully connected layer
Total Parameters: 513,857			
Trainable Parameters: 513,857			
Non-Trainable Parameters: 0			

C. REM CONSTRUCTION

In this subsection, our primary focus is the construction of a deep FL model designed to predict indoor propagation coverage for Wi-Fi networks using x and y coordinates of the area of interest. To design the deployment model, we trained the deep FedLSTM network in a distributed manner. This can be mathematically represented as

$$\mathcal{L}(\theta) = \sum_{j=1}^{N} w_j \mathcal{L}_j(\theta) \tag{2}$$

where $\mathcal{L}(\theta)$ denotes the global loss function, N is the total number of local datasets, w_j signifies the weight assigned to each local dataset, and $\mathcal{L}_j(\theta)$ represents the loss function for the j^{th} local dataset.

The input dataset, denoted $\mathcal{D}$, is partitioned into two subsets: $\mathcal{D}_{\text{train}}$ (for training) and $\mathcal{D}_{\text{valid}}$ (for validation or testing). This division is crucial for effective hyperparameter tuning. As a result, parameters θ are fine-tuned based on evaluation metrics such as relative error ϵ, MAE, RMSE, and R^2 scores.

To visualize the radio map over our target area, we constructed a mesh grid with 1000×1000 points. This grid was delineated by considering the minimum and maximum coordinates within our region of interest. To achieve this, we employed the Emesent Hovermap, which utilizes a LiDAR sensor in tandem with SLAM to produce a 3D representation of the environment. The resulting data are processed in CloudCompare, producing a point cloud. This point cloud can be further refined into a comprehensive 3D model, such as a mesh derived from the identified extreme points.

Subsequent steps involve feature normalization based on the Z-score. We then employ a FedLSTM-driven regression approach tailored for the prediction task. Using the model trained in a federated manner, coverage predictions are determined from the RSSI values for each grid point. The predicted values derived from the federated framework are then combined with horizontal and vertical coordinates of the grid. This results in a REM visualizing the predicted data points as a pseudocolor plot, which is rendered into a 2D map using the pcolor function. Complementing this, a bar graph is introduced to correlate our data with the associated color representation in each plot. For a clearer understanding, Figure 2 provides a detailed graphical representation of the entire procedure.

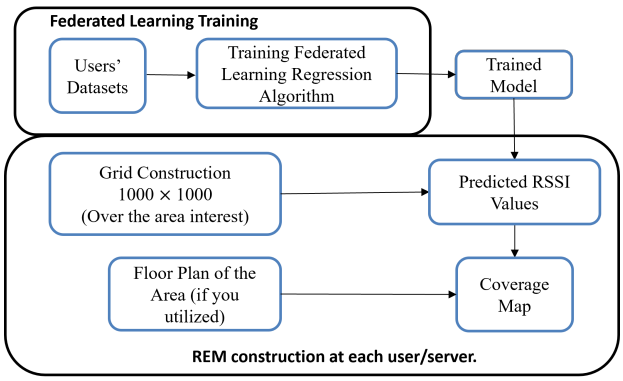


FIGURE 2. REM Construction.

1) The Fine Tuning Process

For construction of the REM, the fine-tuning procedure necessitates systematic optimization of various hyperparameters to achieve optimal model performance. Determining the optimal network architecture, including the choice of layers, is critical. Additionally, regularization parameters are carefully adjusted to prevent overfitting and improve the model's generalization capability. Time steps, which influence LSTM's ability to capture temporal dependencies, are also an essential aspect of this fine-tuning. The entire optimization process is carried out on a trial-and-error basis. The performance of each configuration is assessed using several metrics, namely RMSE, R^2 score, and relative error. Each iteration provides insights, helping subsequent adjustments to refine the model's performance for REM construction. The detailed fine-tuning process is illustrated in Figure 3.

D. MODEL EVALUATION

In this subsection, we evaluate the efficiency of the proposed federated model with different evaluation error metrics: RMSE, MAE, relative error, R^2 score,and mean absolute percentage error (MAPE). These error metrics can best evaluate the performance of the model in different ways. In the field of wireless communications, they play a vital role in evaluating the performance of the prediction model. Predicting RSSI at distinct x and y coordinates demands a comprehensive understanding of the deviations between true measurements

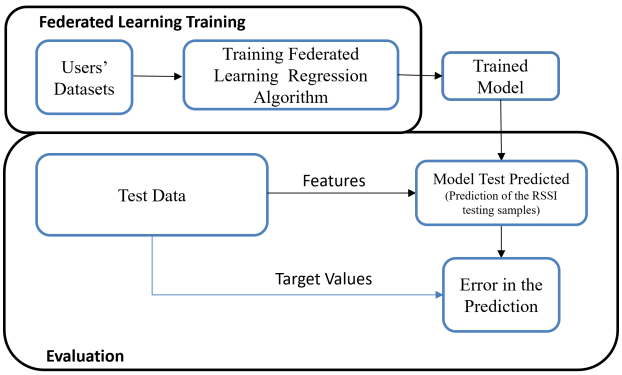


FIGURE 3. Fine-tuning the FedLSTM network for REM construction.

and predicted values, because these can greatly influence the precision of REM generation. Recognizing these error metrics aids in refining and bolstering predictive models. Some of the main metrics are explained below.

Root Mean Square Error shows the average squared deviations between true values, y, and predicted values, $\hat{y}$. In REM building, a low RMSE means the federated LSTM model predicts RSSI values accurately across different locations. The formula for RMSE is

$$Er_{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{3}$$

Mean Absolute Error calculates the average deviation between predicted and true values. This metric becomes essential in scenarios where accurate RSSI predictions are critical for precise REM construction:

$$Er_{\text{MAE}} = \frac{1}{m}\sum_{i=1}^{m}\left|y_i - \hat{y}_i\right| \tag{4}$$

Relative Error provides a deviation measure adjusted by the actual value. Considering the diverse range of wireless signal strengths, this metric evaluates the proportional differences between predicted and actual values across various signal levels:

$$Er_{Relative} = \frac{|\hat{y}_i - y_i|}{y_i} \tag{5}$$

Mean Absolute Percentage Error characterizes the deviations of predictions in terms of their percentage errors from actual observations. Given its foundation in relative accuracy, MAPE is paramount for discerning the fidelity of REM predictions in the context of empirical data:

$$Er_{MAPE} = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{6}$$

The **R^2 score**, also known as the coefficient of determination, measures how well the model's predictions correspond to the actual data. An R^2 value approaching 1 indicates the model accounts for most of the variability in the data, signifying precise RSSI predictions suitable for REM creation. Conversely, an R^2 value approaching 0 suggests the predictions are predominantly aligned with the data's average, potentially lacking in genuine predictive power:

$$Er_{R^2} = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y}_i)^2}, \tag{7}$$

with $\bar{y}_i$ representing the mean target value.

Overall, these metrics address various dimensions of prediction accuracy. By analyzing them collectively, we obtain a comprehensive understanding of the model's performance and robustness, which is critical for optimal wireless communications and precise REM generation.

1) Implementation Platform

We utilized the NVIDIA GeForce RTX 3060 Graphics Processing Unit (GPU) for our training processes. This GPU enables us to undertake sophisticated deep-learning tasks with remarkable efficiency and speed. The NVIDIA GeForce RTX 3060 is equipped with a state-of-the-art Ampere architecture boasting 3584 CUDA cores capable of delivering impressive computational throughput. Alongside this, the card is complemented with 12 GB of GDDR6 memory, ensuring swift data handling during intensive operations. For our development environment, we utilized Python, and structured our deep learning architectures using the TensorFlow framework.

V. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed FL model using various testing scenarios. We established multiple experimental setups to rigorously analyze the behavior of the FedLSTM network under different conditions. Initially, the REM was constructed using a centralized LSTM network with a consistent architecture. This served as a baseline to compare against the federated configuration of the REM employing the same architecture. We conducted evaluations of FedLSTM based on different numbers of clients in the training process, enabling an assessment of performance fluctuations. The effect of varying local epochs on client devices was also studied to evaluate the adaptability of less computationally robust devices. Additionally, FedLSTM's performance was examined across different client configurations and batch sizes, elucidating their respective impacts on model efficiency. Detailed numerical results of these evaluations are provided in the subsequent subsections.

A. COMPARISON WITH A CENTRALIZED LSTM NETWORK

To evaluate our federated model's performance, it is essential to compare it against a centralized model. In our experiment, we employed a federated setup consisting of 10 clients. Each client moved along a different path to collect data from different parts of the area of interest. The model was created on the main server and then distributed to the participating clients. After each client trains the model using their data for these

two rounds, the updates are sent to the central server. The server integrates each update with those from other clients, and then relays the consolidated model to another randomly chosen client. For this experiment, data processing was conducted in batch sizes of 32. Referring to the results in Figure 4, our Federated LSTM model's performance metrics (RMSE, MAE, and R^2 score) were assessed against the centralized model.

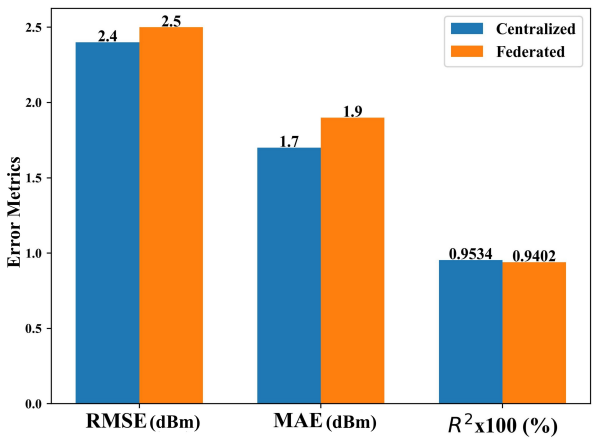


FIGURE 4. Comparison of centralized and federated LSTM networks

The federated model's RMSE was marginally elevated by 0.1 dBm in comparison to its centralized counterpart. The MAE metric indicated a prediction difference of approximately 0.2 dBm. When evaluating the R^2 score, the centralized model exhibited a slight advantage. However, it is crucial to highlight that while the centralized model offers marginally better performance, it raises concerns over data privacy, demands higher computational resources, and creates more communications overhead. Particularly in the process of aggregating data across an expansive geographic spectrum for REM construction, the centralized methodology demands substantial bandwidth for data transmission, which is a concern for participants reluctant to disclose their location due to potential security and privacy breaches.

B. IMPACT OF THE NUMBER OF CLIENTS ON GLOBAL CONVERGENCE

In this assessment, we evaluate the impact of the number of clients participating in convergence of the federated model. The model was employed in different federated settings; each time, the number of participating clients was changed to observe the behavior of the model. We evaluated the regression model with five error metrics, namely, RMSE, MAE, R^2, relative error, and MAPE. In general, the performance of the model can attain the expected efficiency with various numbers of clients participating in the FL model. However, the more clients, the sooner the system will converge to the global minimum.

Therefore, with a large number of clients using FedLSTM, prediction accuracy can quickly achieve high performance, and thus, the accuracy will be more stable through communication rounds. Figure 5 shows that for C = 16, the R^2 score of the FedLSTM model reached 90% after just two communication rounds, and reached 93% after 10 rounds. In contrast, the system with few assigned clients (C = 4) had a relatively lower R^2 score, not going above 90% even after 10 communication rounds.

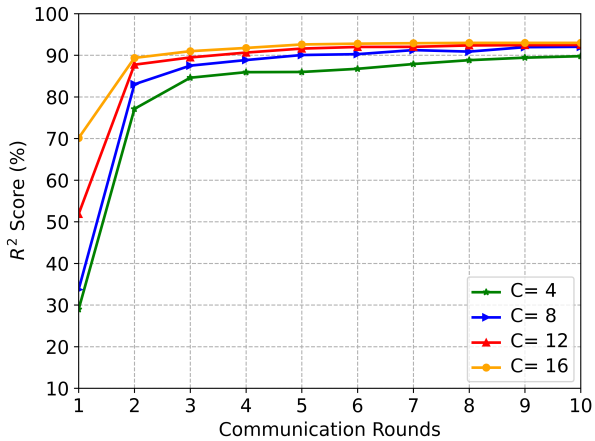


FIGURE 5. R^2 variations from the number of clients

Similarly, Figure 6 shows that the RMSE of the federated model also showed a lower value, which means best performance, when the number of clients in model training was high (C = 16). The proposed federated model showed a lower RMSE in all communication rounds when C was 16. This error metric was high when C was 4. In the early communication rounds, the model with four clients showed fast convergence. However, convergence was reduced after five communication rounds. This means that with fewer clients, the model may start with sharp convergence, but it may stop further along with certain communication updates.

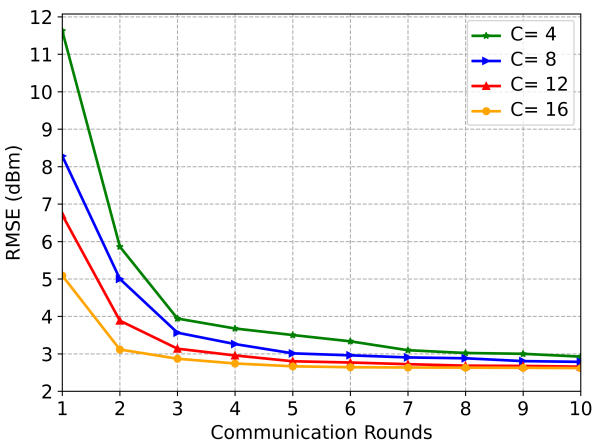


FIGURE 6. RMSE variations from the number of clients

In addition, MAE calculates the average deviation between predicted and true values, and in our setting, MAE also fol-

lowed the trends in Figure 7. The graph depicts high MAE in every communication round when C = 4. Even after 10 rounds, MAE was 3 dBm, just above 2 dBm when clients numbered 8 and 12. However, MAE decreased continuously to 2 dBm after eight communication rounds, and the trend continued, decreasing to below 2 after 10 rounds.

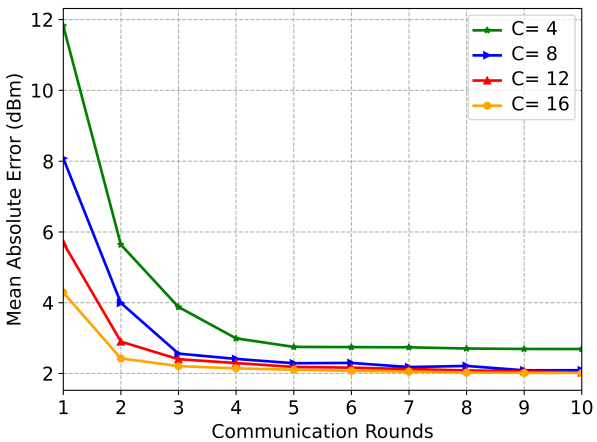


FIGURE 7. MAE variations from the number of clients

Furthermore, the model was evaluated with relative error to show the deviations from actual values and the proportional differences between predicted and actual values across the data. The relative error graph in Figure 8 illustrates that an increase in clients had a strong impact. Relative error was above 0.06 dBm before five communication rounds had been completed, and then dropped to about 0.05 dBm after 10 rounds. This error metric followed the same trend as metrics discussed above, meaning that relative error decreased with each client added to train and update the model. This error was near 0.03 dBm when C = 16 after 10 rounds for updating the model. With that many clients, the error dropped very fast in the beginning as well.

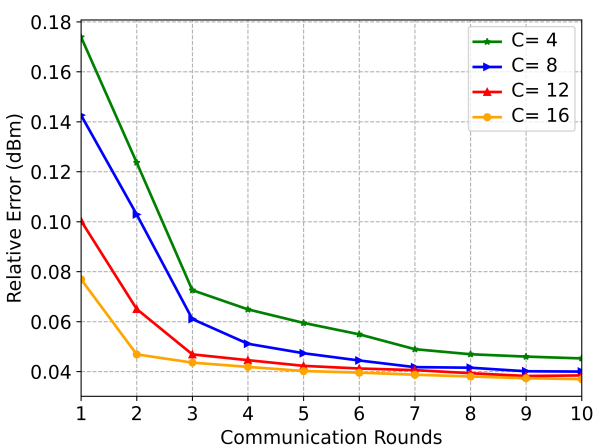


FIGURE 8. Relative error based on the number of clients

Finally, to characterize deviations in predictions in terms of percentage errors from actual values the mean absolute percentage error was evaluated based on variations in the number of clients updating the model. Figure 9 shows MAPE

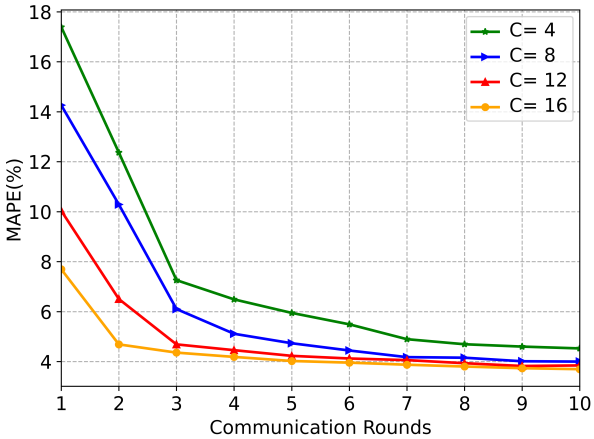


FIGURE 9. MAPE variations based on the number of clients

was below 4% when C = 16 after 10 rounds. In contrast, this error increased to around 5% when C = 4. MAPE was around 4% with eight and 12 clients, but converged to these values in the final updates, compared with 16 clients updating model weights.

Overall, for a system with 16 clients, in comparison with the four-client system, accuracy and precision were higher with notably fewer errors. Generally, in a predetermined number of training rounds, the number of clients affects the FL process accuracy and convergence speed. As more clients participated in the FL process in each round, the model's absolute precision and training speed suffered from fewer adverse effects. Nevertheless, once C improved to a particular level, advancement of system performance was less noteworthy, and sometimes even degraded. When putting the FL process into practice, we can face difficulties in that as C increases, more clients update local parameters to the server. As a consequence, the communication and computation costs of the FL model amplify significantly. This is encouraging because, in real-world applications with large-scale clients, we only need to select a set of clients from the network to execute the FL process in each communication round. This procedure saves considerable communication costs in the FL process.

C. CONTRIBUTION OF LOCAL MODEL EPOCHS TO GLOBAL CONVERGENCE

In ML, an epoch is one complete cycle through the full training dataset. Specifically, each epoch involves both forward propagation (estimation of output given the input) and backward propagation (adjustment of model weight based on error). The number of epochs is a hyperparameter that determines how many times the learning algorithm will work through the entire training dataset. In the context of FL, where

data remain distributed across multiple devices or nodes and do not converge centrally, the local number of epochs becomes a pivotal factor. It dictates how much training is conducted on each local dataset before aggregating model updates into the global model. This subsubsection delves into the nuanced influence that varying the local number of epochs has on convergence speed and quality in FL setups.

From Figure 10, it is evident that as the number of epochs for training on each selected client per communication round increased, the RMSE of the federated model decreased. To elucidate, in the federated setting where the number of epochs was set to two ($\epsilon = 2$), the model experienced rapid convergence initially but plateaued at a value of 2.75 dBm after 20 epochs. In contrast, as ϵ was incremented to 3, 4, and finally 5, the trends illustrate that the RMSE of the model with $\epsilon = 5$ remained consistently lower than the other three configurations in every communication round.

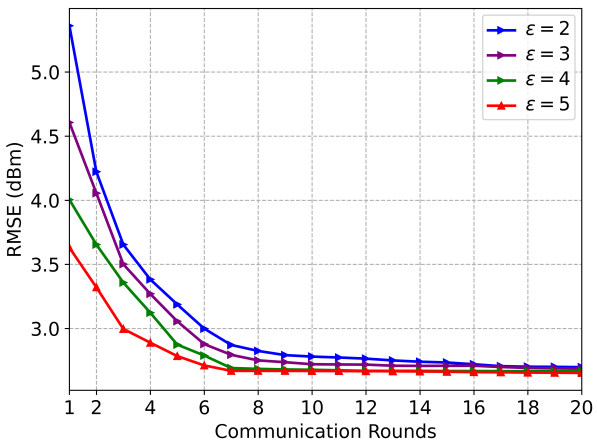


FIGURE 10. RMSE variations from the local number of epochs

Similarly, to assess the accuracy with which our federated model represents the data, we evaluated the R^2 score. As depicted in Figure 11, our federated model achieves a commendable R^2 score of 93% after only a few communication updates when ϵ is set to 5. However, when the local epochs ϵ are adjusted to 4, 3, and 2, there is a subsequent decrease in the R^2 score for each setting. However, their impact on the global model convergence decreases with the increase in the number of communication rounds.

Furthermore, the model was assessed using relative error to demonstrate deviations, adjusted by the actual values, and to display the proportional differences between the predicted and actual values across the data. Figure 12 illustrates that as ϵ increased from 2 to 5, the FedLSTM model mirrored the trend exhibited by variations in the number of clients. In the initial communication rounds, ϵ significantly impacted global convergence. There appears to be a substantial difference between $\epsilon = 2$ and $\epsilon = 5$ in the early stages; however, after each update, this difference decreased to a certain extent. By the 20th communication round, the difference in relative error amongst the four settings for ϵ (5, 4, 3, and 2) was

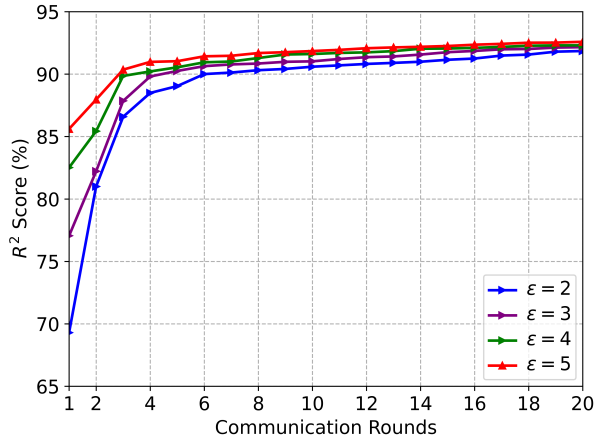


FIGURE 11. R^2 score variations based on the number of local epochs

minimal, indicating that with more communication rounds, models with fewer or more epochs can exhibit nearly identical performance.

Figure 13 illustrates how MAPE, expressed as a percentage, quantified prediction deviations from actual error values based on variations in the number of clients during model updates.

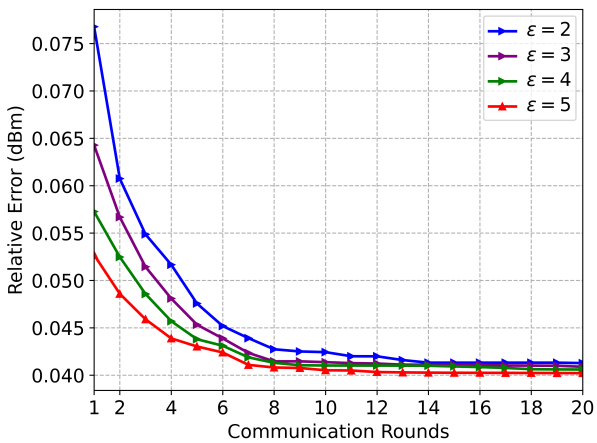


FIGURE 12. Variations in relative error based on the number of local epochs

Lastly, mean absolute error quantifies the average deviation between predicted and actual values. In our experiments, trends in MAE were consistent with those observed in other error metrics. Figure 14 illustrates how MAE was notably high during the initial communication rounds when ϵ =2. Conversely, when ϵ = 5 for the first model aggregation, MAE was significantly lower. Yet as the aggregation rounds proceeded, the disparity in MAE between different ϵ settings diminished considerably, eventually converging to a value slightly above 2 dBm across all settings. Nonetheless, the model with $\epsilon = 5$ consistently registered a lower MAE, whereas $\epsilon = 2$ consistently exhibited a higher error value

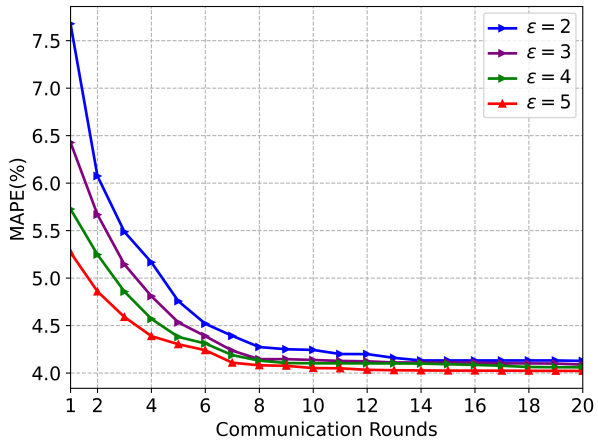


FIGURE 13. **Variations in MAPE based on the number of local epochs**

throughout the communication rounds.

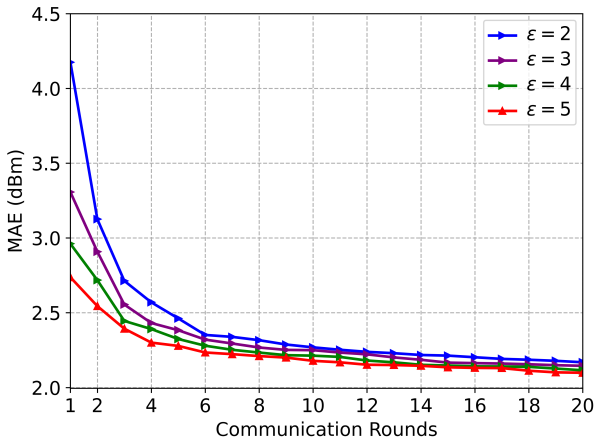


FIGURE 14. **Variations in MAE based on the number of local epochs**

VI. GRAPHICAL RESULTS

In the domain of wireless networking, the received signal strength indicator is a metric quantifying the signal power a device receives from the access point or router. Expressed in decibel milliwatts (dBm), RSSI is a pivotal criterion in evaluating the quality of a Wi-Fi connection. Under optimal wireless communications conditions, an RSSI value of -30 dBm represents peak signal reception. However, achieving such an ideal benchmark is infrequent in practical environments due to various interference and propagation factors. Signal strengths ranging from -50 to -30 dBm denote a very stable connection, facilitating seamless streaming and downloading. Between -60 and -50 dBm, the signal is still sufficiently strong for most standard applications, including uninterrupted streaming. However, as signal strength falls to the range of -70 to -60 dBm, performance becomes notably average. Connections in this range may struggle during intensive tasks, particularly if the network sees simultaneous activity from multiple devices or if the connected device is significantly mobile. Progressing further down the scale, a range of -80 to -70 dBm indicates an inconsistent connection prone to disruptions, especially during data-intensive operations like streaming. An even weaker range, -90 to -80 dBm, offers a highly unstable connection that is not only susceptible to frequent dropouts but also languid speeds. Any signal weaker than -90 dBm is practically non-functional, rendering a reliable connection almost impossible.

The dispersion of wireless signal strength is graphically represented as a 2D color map using the predicted RSSI values from the trained FL model. Figure 15(a) illustrates a LiDAR map of the area of interest constructed through the Emesent Hovermap, while Figure 15(b) displays the coverage map generated using the centralized LSTM model. In Figure 15(c), the REM constructed through the proposed FedLSTM model is presented. REMs constructed through the LSTM centralized model and FedLSTM are comparable, showing similar signal dispersion. Specifically, the region near the access point exhibits RSSI values above -50 dBm, indicating excellent signal strength. However, as one moves farther away from the access point, the RSSI values gradually decrease and eventually drop below -80 dBm at the far end of the area.

To further evaluate and test our proposed federated model under different settings, we conducted experiments inside a room. For this purpose, we utilized Room 7-602 at the University of Ulsan in South Korea. In this federated setting, we engaged five clients, each of which updated the model over 10 iterations for 10 communication rounds. Figure 16(a) provides an overview of the room's layout, including an example path (*path_1*) followed by Client-1. In our scenario, we considered 10 paths taken by five clients. The centralized model's REM is Figure 16(b), and FedLSTM's REM in Figure 16(c) highlights the excellent coverage near the access point and the gradual reduction in signal strength when moving away from it. These graphic results demonstrate that in both room and corridor settings, FedLSTM achieved performance comparable to centralized approaches while concurrently ensuring data privacy, minimizing communications overhead, and reducing the server load.

VII. CONCLUSION

In this study, we presented an innovative approach called FedLSTM and based on FL for REM construction. In particular, the proposed FedLSTM framework is designed to predict indoor network coverage while addressing the pressing issue of data privacy. This proposed method is decentralized, allowing clients to participate in the training process without disclosing their data to a central server. For this approach, we utilized real measurements from various clients navigating different paths and updating their models with data from each path. When compared to a centralized counterpart, our model indicated a slight rise in RMSE to 2.5 dBm from 2.4 dBm and an increase in MAE to 1.9 dBm from 1.7 dBm.

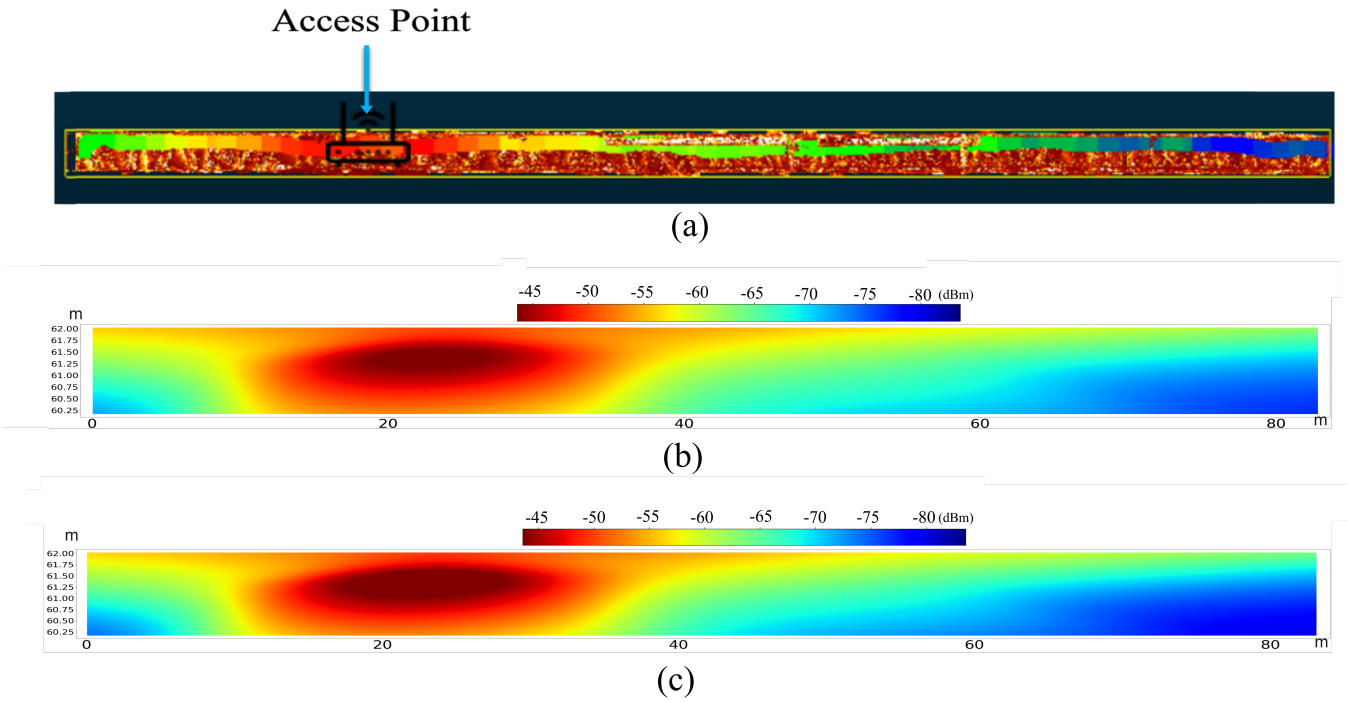


FIGURE 15. (a) Corridor layout, (b) REM from the centralized LSTM model, and (c) REM from FedLSTM

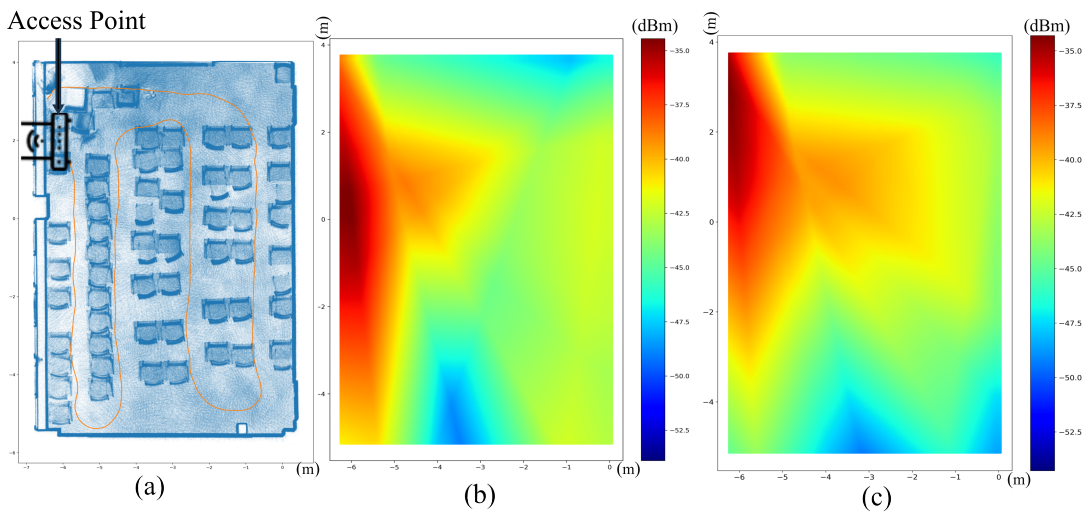


FIGURE 16. (a) Room layout, (b) REM from the centralized LSTM model, and (c) REM from FedLSTM

Subsequently, we evaluated the FedLSTM model considering variations in the number of participating clients. Numerical results revealed that increasing the number of clients enhanced performance metrics such as error rate and R^2 score. Additionally, adjusting the number of local training epochs showed that even devices with limited computational power can meaningfully contribute to training the federated model, with fewer epochs still achieving competitive results. Graphic comparisons of REMs from FedLSTM and centralized LSTM highlighted their similarities. However, FedLSTM distinctly provided the trifold benefits of data security, communications efficiency, and a reduced server load. This research emphasizes FL's potential to ensure data privacy while upholding robust performance from indoor network coverage predictions.

Our future work includes adapting the framework for mapping outdoor environments and integrating geographic data for more comprehensive REM construction. We also plan to develop optimized communication protocols between servers and clients, and create an advanced client selection algorithm. These enhancements aim to improve data diversity and client comfort, further extending the applicability and efficiency of federated learning in diverse settings.

• • •

REFERENCES

[1] Raman Kumar, Sita Rani, and Mohammed Al Awadh. Exploring the application sphere of the internet of things in industry 4.0 A review, bibliometric and content analysis. *Sensors*, 22(11):4276, jun 2022.

[2] Chen-Fu Chien, Wei-Tse Hung, and Eddy Ting-Yi Liao. Redefining monitoring rules for intelligent fault detection and classification via cnn transfer learning for smart manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 35(2):158–165, may 2022.

[3] Shih-Fan Chou, Hsiu-Wen Yen, and Ai-Chun Pang. A REM-enabled diagnostic framework in cellular-based IoT networks. *IEEE Internet of Things Journal*, 6(3):5273–5284, jun 2019.

[4] Carla E. Garcia Moreta, Mario R. Camana Acosta, and Insoo Koo. Prediction of digital terrestrial television coverage using machine learning regression. *IEEE Transactions on Broadcasting*, 65(4):702–712, dec 2019.

[5] Carla E. Garcia, Mario R. Camana, and Insoo Koo. Ensemble learning aided QPSO-based framework for secrecy energy efficiency in FD CR-NOMA systems. *IEEE Transactions on Green Communications and Networking*, 7(2):649–667, jun 2023.

[6] Marwan Khan and Sanam Noor. Performance analysis of regression-machine learning algorithms for predication of runoff time. *Agrotechnology*, 08(01), 2019.

[7] Wangyang Xu, Jiancheng An, Yongjun Xu, Chongwen Huang, Lu Gan, and Chau Yuen. Time-varying channel prediction for ris-assisted mu-miso networks via deep learning. *IEEE Transactions on Cognitive Communications and Networking*, 8(4):1802–1815, 2022.

[8] Yan Zhang, Jinxiao Wen, Guanshu Yang, Zunwen He, and Jing Wang. Path loss prediction based on machine learning: Principle, method, and data expansion. *Applied Sciences*, 9(9):1908, may 2019.

[9] Carla E. Garcia and Insoo Koo. Coverage prediction and REM construction for 5g networks in band n78. In *"2023 15th International Conference on Computer and Automation Engineering (ICCAE)"*. IEEE, mar 2023.

[10] Youping Zhao, Jeffrey H. Reed, Shiwen Mao, and Kyung K. Bae. Overhead analysis for radio environment mapenabled cognitive radio networks. In *2006 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, pages 18–25, 2006.

[11] Andreas Achtzehn, Janne Riihihjärvi, Irving Antonio Barriía Castillo, Marina Petrova, and Petri Mähönen. Crowdrem: Harnessing the power of the mobile crowd for flexible wireless network monitoring. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 63–68, 2015.

[12] Jin Wang, Nicholas Tan, Jun Luo, and Sinno Jialin Pan. Woloc: Wifi-only outdoor localization using crowdsensed hotspot labels. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[13] Saptarshi Debroy, Shameek Bhattacharjee, and Mainak Chatterjee. Spectrum map and its application in resource management in cognitive radio networks. *IEEE Transactions on Cognitive Communications and Networking*, 1(4):406–419, 2015.

[14] V Chowdappa, C Botella, JJ Samper-Zapater, and RJ Martinez. Distributed radio map reconstruction for 5 g automotive, ieee intell. transp. syst. mag. 10 (2)(2018) 36–49, 2018.

[15] Sandra Roger, Carmen Botella, Juan J Pérez-Solano, and Joaquin Perez. Application of radio environment map reconstruction techniques to platoon-based cellular v2x communications. *Sensors*, 20(9):2440, 2020.

[16] Olaonipekun Oluwafemi Erunkulu, Adamu Murtala Zungeru, Caspar K. Lebekwe, and Joseph M. Chuma. Cellular communications coverage prediction techniques: A survey and comparison. *IEEE Access*, 8:113052–113077, 2020.

[17] Complexity reduction of ordinary kriging algorithm for 3d rem design. *Physical Communication*, 55:101912, 2022.

[18] Marko Pesko, Tomaž Javornik, Andrej Košir, Mitja Štular, and Mihael Mohorčič. Radio environment maps: The survey of construction methods. *KSII Transactions on Internet & Information Systems*, 8(11), 2014.

[19] Zhifeng Han, Jianxin Liao, Qi Qi, Haifeng Sun, Jingyu Wang, et al. Radio environment map construction by kriging algorithm based on mobile crowd sensing. *Wireless Communications and Mobile Computing*, 2019, 2019.

[20] Pradipta Maiti and Debjani Mitra. Ordinary kriging interpolation for indoor 3d rem. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15, 2022.

[21] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968.

[22] David Plets, Wout Joseph, Kris Vanhecke, Emmeric Tanghe, and Luc Martens. Coverage prediction and optimization algorithms for indoor environments. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–23, 2012.

[23] P Sruthi and K Sahadevaiah. A novel efficient heuristic based localization paradigm in wireless sensor network. *Wireless Personal Communications*, 127(1):63–83, 2022.

[24] Rongrong He, Yuping Gong, Wei Bai, Yangyang Li, and Ximing Wang. Random forests based path loss prediction in mobile communication systems. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1246–1250, 2020.

[25] Usama Masood, Hasan Farooq, and Ali Imran. A machine learning based 3d propagation model for intelligent future cellular networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[26] Harsh Singh, Shivam Gupta, Charchit Dhawan, and Amrita Mishra. Path loss prediction in smart campus environment: Machine learning-based approaches. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5, 2020.

[27] Stephen Ojo, Agbotiname Imoize, and Daniel Alienyi. Radial basis function neural network path loss prediction model for lte networks in multitransmitter signal propagation environments. *International Journal of Communication Systems*, 34(3):e4680, 2021.

[28] Marco Sousa, André Alves, Pedro Vieira, Maria Paula Queluz, and António Rodrigues. Analysis and optimization of 5g coverage predictions using a beamforming antenna model and real drive test measurements. *IEEE Access*, 9:101787–101808, 2021.

[29] Yan Zhang, Jinxiao Wen, Guanshu Yang, Zunwen He, and Jing Wang. Path loss prediction based on machine learning: Principle, method, and data expansion. *Applied Sciences*, 9(9):1908, 2019.

[30] Han-Shin Jo, Chanshin Park, Eunhyoung Lee, Haing Kun Choi, and Jaedon Park. Path loss prediction based on machine learning techniques: Principal component analysis, artificial neural network, and gaussian process. *Sensors*, 20(7):1927, 2020.

[31] Nektarios Moraitis, Lefteris Tsipi, Demosthenes Vouyioukas, Angelina Gkioni, and Spyridon Louvros. Performance evaluation of machine learning methods for path loss prediction in rural environment at 3.7 ghz. *Wireless Networks*, 27(6):4169–4188, 2021.

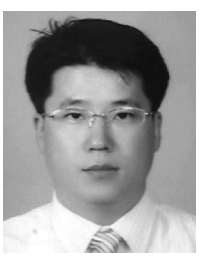
[32] Carla E. García and Insoo Koo. Extremely randomized trees regressor scheme for mobile network coverage prediction and rem construction. *IEEE Access*, 11:65170–65180, 2023.

[33] Wangyang Xu, Jiancheng An, Hongbin Li, Lu Gan, and Chau Yuen. Algorithm unrolling-based distributed optimization for ris-assisted cell-free networks. *IEEE Internet of Things Journal*, pages 1–1, 2023.

SHAFI ULLAH KHAN received his bachelor's degree in electrical engineering from the University of Engineering and Technology Peshawar, Pakistan, in 2020. He is currently pursuing the M.S. degree in electrical engineering with department of Electrical Electronics and Computer Engineering at University of Ulsan, South Korea. His research interests include Wireless communication, machine learning, Federated learning, digital health care and Internet of Things.

CARLA E. GARCÍA received the B.Eng. degree in electronics and telecommunications engineering from Escuela Politécnica Nacional (EPN), Quito, Ecuador, in 2016, and the M.Sc. and Ph.D. degrees from the University of Ulsan, South Korea, in 2020 and 2023, respectively. From 2017 to 2023, she was a Research Assistant with the Department of Electrical, Electronic and Computer Engineering, University of Ulsan. In September 2023, she joined as a Postdoctoral Researcher at the University of Ulsan. Her main research interests include artificial intelligence, security, NOMA, MIMO and satellite communications.

TAEWOONG HWANG received his bachelor's degree in electrical engineering from the University of Ulsan, South korea, in 2022. He is currently pursuing his master's degrees in electrical engineering with Department of Electrical Electronics and Computer Engineering, at university of Ulsan, South korea. His research interest includes IoT, Embedded System and M2M.



IN-SOO KOO received the B.E. degree from Konkuk University, Seoul, South Korea, in 1996, and the M.Sc. and Ph.D. degrees from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 1998 and 2002, respectively. From 2002 to 2004, he was a Research Professor with the Ultrafast Fiber-Optic Networks Research Center, GIST. In 2003, he was a Visiting Scholar with the KTH Royal Institute of Science and Technology, Stockholm, Sweden. In 2005, he joined the University of Ulsan, Ulsan, South Korea, where he is currently a Full Professor. His research interests include spectrum sensing issues for CRNs, channel and power allocation for cognitive radios (CRs), and military networks, SWIPT MIMO issues for CRs, MAC, and routing protocol design for UW-ASNs, and relay selection issues in CCRNs.