Communication-efficient Federated Learning Design with Fronthaul Awareness in NG-RANs

Ayman Younis, Chuanneng Sun, and Dario Pompili
Department of Electrical and Computer Engineering
Rutgers University–New Brunswick, NJ, USA
Emails: {a.younis, chuanneng.sun, pompili}@rutgers.edu

***Abstract*—Next Generation Radio Access Networks (NG-RANs) have become a promising paradigm to meet the strict demands of the 5G and beyond applications by distributively pushing the radio and computing functionalities in the close approximate to end-users. With the emergence of new technologies, network densification, and richer and more demanding applications, the limited capacity of the fronthaul links and privacy concerns poses a severe constraint on realizing NG-RAN systems in the real environment. To tackle these challenges, we propose a Federated Learning (FL)-based NG-RAN algorithm, named FedNG, in which the User Equipment (UEs), as well as NG-RAN infrastructures, collaborate throughout the learning process and the sharing prediction model to ensure privacy and relieve the burden on fronthaul interface. Specifically, our proposed scheme enables Distributed Units (DUs) to cooperatively learn a shared predictive model by taking the first-phase training models of the DUs as the initial input of the local training and then uploading sub-optimal DU models to the Central Unit (CU) to involve in the next phase of global training. Finally, numerical results are provided to evaluate our proposed scheme in terms of accuracy, service latency, and traffic size. The convergence of our proposed algorithm confirms that our approach significantly outperforms the existing state-of-the-art solution based on Federated Averaging (FedAvg).**

***Index Terms*—Federated learning; Next Generation Radio Access Network; Fronthaul Capacity; Resource Managements.**

I. Introduction

Background and Motivation: The last decades have witnessed huge growth in portable and Internet of Things (IoT) equipment, making numerous limited computation capacity devices interact with wireless network systems via cellular channels. It is expected that the connected number of IoT devices increasing to 14.7 billion by 2023 [1]. Practically, the portable IoT sensing units usually are embedded with lightweight capabilities (i.e., computing, storage, and communication), and they continuously run to generate a large amount of traffic data demanded to process and analyze. The most popular solution to address this issue is based on cloud-based computation task offloading approaches, such as in [2]–[4], in which the User Equipment (UEs) (e.g., IoT devices, smartphones, wireless sensors, etc.) offload the raw data, pre-processed data, to edge cloud servers for additional assistance. However, computation task offloading incurs extra overheads in terms of latency and computing cost due to the additional *communication cost* needed for transmitting workload from end-user devices to edge servers and *computation cost* required for data executing.

In light of the above, many mobile network operators are working to meet beyond 5G network demands, including; i) opening and virtualizing the Radio Access Network (RAN) layers; ii) achieving ultra-low latency, high data rate, and low operational expenses; and iii) providing privacy, scalability, efficiency, as well as enabling machine learning-based cloud services to end-users. Next Generation RAN (NG-RAN) has been recently emerged as a key candidate solution to enable flexibility and efficiency [5]. The NG-RAN architecture, defined by 3GPP, comprises a Distributed Units (DUs) located in the close proximity to the Base Station (BS) tower that can communicate with a Central Unit (CU) via Next Generation Fronthaul Interface (NGFI) standard [6], in which the PHY/MAC layers of the network flexibly splits between the CU and DU locations. Despite the significant improvements brought from deploying NGFI in NG-RAN, different specifications in terms of reducing fronthaul traffic and securing privacy should be considered to better support the NG-RAN large-scale deployment and prevent the fronthaul interface from being the bottleneck of NG-RAN.

The RAN systems with an embedded traditional machine learning approach usually consider that all local data is forwarded to a centralized cloud server for performing and training. However, the difficulty of satisfying private constraint and the high cost of transmitting the raw data to the central servers due to high *round trip latency* are driving the need for a highly decentralized machine learning approach. Motivated by this, Federated Learning (FL) has emerged to realize the collaborative training of a machine learning model without requiring to publish the original stream data with any third-party application. In such a scenario, it is possible for machine learning algorithms to gain experience from a vast range of data located at several locations. Enabling FL in NG-RAN is beneficial in terms of providing privacy for the end-users while running applications. Accordingly, the DUs and CUs can be enabled to collaborate on the development of tanning models, in a distributed machine learning manner, without needing to directly share sensitive data collected from user devices. Specifically, In this work, we aim at answering the key question, how can FL be leveraged in NG-RAN systems to ensure privacy and relieve the burden on the fronthaul interface?

Related Works: Several existing works with respect to wireless radio systems generally emphasize on centralized learning schemes [7], [8]. The work [7] presented a comprehensive survey of implying centralized deep learning and machine learning models to realize intelligent network paradigms. The authors in [8] proposed a distributed cooperative massive access approach based on deep reinforcement learning to meet the user's demands while satisfying reliability and latency constraints in massive access scenarios. However, applying the existing centralized-based machine learning in RANs is challenging. That is because it demands large truing data in order to perform better than other techniques. Besides, due to complex data models, it is considered an costly method to get high accuracy in terms of computation resources.

In terms of the benefit of cloud-based RAN architectures, there has been a considerable number of works studying the cooperative communications perspectives to deal with resource allocation and cost reduction. Considering constraints on service latency, quality loss, and edge capacity, the problem of joint task offloading, latency, and approximate computing, the work in [9] proposed a novel optimization approach for latency and quality tradeoff task allocation in NG-RANs. In [10], the authors introduced a resource manager for virtual RANs based on deep reinforcement learning, named *vrAIn*, that can save in computing capacity of up to 30% over CPU-agnostic methods and improve throughput by 25% over existing schemes. The work in [11] presented allocation algorithms that optimize the energy consumption of a cloud-RAN, including real time experiments on a programmables cloud-RAN testbed to show the characteristics of the baseband unit in terms of CPU utilization and radio resources.

Recently, FL has attracted a lot of focus as a novel distributed machine learning approach over the past few years. For instance, based on iterative model averaging, Google presented, in 2017, a practical method for the FL to train a deep learning model without centralizing the data at the data center [12]. The work in [13] provided a survey on FL concepts in terms of classification, essential challenges, and new designs. Besides, FL has been shown as an effective method to fundamentally enhance the performance of cloud-based RAN systems [14]–[16]. In [14], the authors utilized the FL method in a Device-to-Device (D2D) communication and Mobile Edge Computing (MEC), in which the D2D groups transmit their training models to the MEC server to reduce traffic. Liu *et al.* [15] introduced a client-edge-cloud hierarchical FL system associated with a HierFAVG algorithm to enable multiple edge servers. Despite many research efforts, there are still open challenges for learning-based in NG-RAN. (i) privacy, most of the real-time applications (e.g., aerial search and rescue, mobile healthcare, and video caching) are developed on the gathered massive data from UEs. However, uploading these data to a central server for model training may outcome in critical privacy issues, and (ii) the fronthaul capacity constraint in NG-RAN represents the critical communication bottlenecks. Therefore, we propose a FL-based scheme to maintain privacy and relieve the burden on fronthaul in NG-RANs.

Main Contributions: The main contributions in this paper can be summarized as follows,

- To overcome the challenges of the privacy demand and the limited capacity of the fronthaul links in NG-RAN, we leverage FL for privacy-preserving and latency-aware services in the NG-RAN systems. Our proposed FedNG algorithm enables DUs to cooperatively learn a shared predictive model by assuming the first-phase training models of the DUs as the initial input of the local training and then uploading sub-optimal local models to the CUs to involve in the next phase of global training.
- We address the privacy and traffic management problem in two scenarios: (i) distributed FL solution, NG-FedAvg algorithm, in which a learning model can be trained cooperatively between the computing servers, DUs, and the end-users, UEs, to achieve the required accuracy level; and (ii) proposed FedNG algorithm, in which a two-layer-aggregation federated learning paradigm is to improve the overall system performance. Hence, a learning model can be trained cooperatively between the CU, DUs, and the UEs, to achieve the required accuracy level.
- To efficiently evaluate our proposed framework, we perform extensive experiments using three real-world datasets—MNIST with Dense Neural Networks (DNNs, CI- FAR10 with convolutional neural networks (CNNs), and IMDB sentimental analysis dataset with recurrent neural networks (RNNs). In the three datasets, the performance of our proposed framework shows a significant improvement compared with existing traditional Federated Averaging (FedAvg) in terms of accuracy, service latency, and traffic size.

Paper Organization: The rest of this paper is organized as follows. Sect. II describes the overall system models used in this paper. Sect. III introduces the proposed privacy-preserving FL schemes for NG-RAN systems. Simulation results are provided in Sect. IV. Finally, we conclude the paper in Sect. V.

II. Overall NG-RAN System Design

This section first presents the network description, including NG-RAN architecture and network layers. Followed by the learning model, which presents the main terms and equations of the FL framework, as well as a distributed solution for the NG-FedAvg algorithm.

A. Network Description

In this paper, we assume that a generic NG-RAN architecture comprises three layers (see Fig. 1) listed as follows,

Central unit layer: contains powerful processing servers, providing on-demand computation /radio functions for uplink/downlink wireless communication channels between the UEs and gNBs. In terms of FL, the local model generated by mobile user devices can be collected as a global model and sent back to the UEs for extra training.

Distributed unit layer: consists of a set of edge servers deployed in proximity to end-users to provide radio/computation services. In the context FL, DU servers can be exploited to

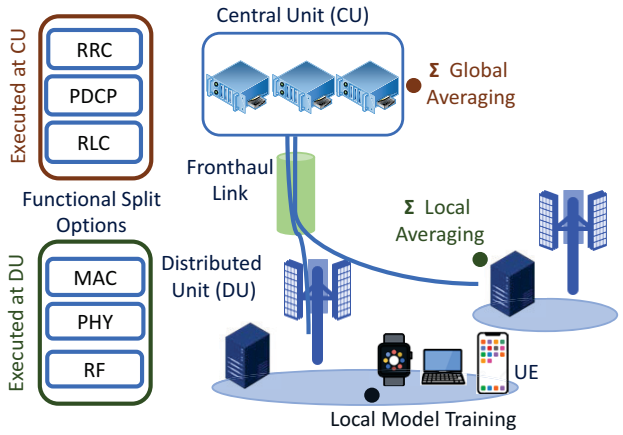


Fig. 1. NG-RAN architecture with enabling a federated learning technique.

perform local data processing functions, in which each DU server is used as a *local aggregator* to exchange the model between end-users and CU servers.

UE layer: comprises a set of UEs, $\mathcal{U} = \{1, 2, ..., U\}$, randomly distributed in the network cell. Each UE is equipped with an antenna/sensor to collect data, which is used for training purposes, from the NG-RAN environment. Due to the UE location in the vicinity of the DUs, each UE has the ability to exchange the training model to its DUs for local aggregations to experiment with high QoS and low latency. Accordingly, we assume that each DU interacts with a set $\mathcal{S} = \{1, 2, .., S\}$ of S servers. Plus, we assume that DU s associates with a set $\mathcal{U}_s = \{1, 2, ..., U_s\}$ of U_s UEs with $U = \sum_{s\in\mathcal{S}} U_s$, and each UE is served by one DU server.

B. Learning Model

Let consider a NG-RAN system, as depicted in Fig. 1, in which each mobile user $u, \forall u \in \mathcal{U}_s$, in this system is connected to DU $s, \forall s \in \mathcal{S}$ via wireless channel to collect a local input dataset $\mathcal{D}_{us} = \{\mathbf{x}_d, y_d\}_{d=1}^{|D_{us}|}$, where $\mathbf{x}_d \in \mathbb{R}^f$ and $y_d \in \mathbb{R}$ are a f-dimensional input vector and the corresponding label, respectively. Assuming non-i.i.d. distributed data thought the wireless network, we consider that $\mathcal{D}_{us} \cap \mathcal{D}_{ú\acute{s}} = \emptyset, \forall (u, s) \neq (ú, \acute{s})$.

Definition 1. *The terms "local model" and "local aggregation model" are referred to as the models generated by UEs and DUs, respectively, while averaging at CU is referred to as the "global model".*

The key goal of the FL system is to leverage the datasets of all UEs without sacrificing their privacy. In the light of this, we first denote a loss function as $l(\mathbf{w}, \mathbf{x}_d, y_d)$ for each data sample $(\mathbf{x}_d, y_d)$ to specify the estimated error between the input $\mathbf{x}$ on the learning model $\mathbf{w} \in \mathbb{R}^f$ and the corresponding label y_d. Similar to [17], [18], the local loss function of the learning model $\mathbf{w}$ on the dataset D_{us} can be defined as,

$$L_{us}(\mathbf{w}|\mathcal{D}_{us}) = \frac{1}{\mathcal{D}_{us}} \sum\nolimits_{d\in\mathcal{D}_{us}} l(\mathbf{w}, \mathbf{x}_d, y_d). \quad (1)$$

Due to the randomness of the UE data distributions, we assume that the empirical loss function thought the overall network dataset $D = \cup_{u,s} D_{us}$ can be modeled as,

$$L(\mathbf{w}|\mathcal{D}_{us}) = \frac{\sum_{u\in\mathcal{U}_s} \sum_{s\in\mathcal{S}} L_{us}(\mathbf{w}|\mathcal{D}_{us})}{U} \quad (2)$$

In general, the target of designing a FL algorithm is to achieve the optimal model $\mathbf{w}^*$ minimizing the global loss value as,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^f} L_{us}(\mathbf{w}|\mathcal{D}_{us}) \quad (3)$$

In this work, we aim at developing an algorithm for the end users in the NG-RAN system to efficiently achieve the solution of the optimization problem in (3).

C. Distributed Solution

To achieve the optimal solution in (3), we have extended the Federated Averaging (FedAvg) algorithm [14], [19], which is a standard algorithm in the FL using a distributed approach to iterative minimize the local loss in (3). The main steps of our proposed NG-FedAvg algorithm for the NG-RAN system can be listed as follows,

(i) Data Collecting: For the first time, each UE sends its collected data from a third-party application. Then, the learning models can be constructed based on the empirical risk minimization criterion with respect to the loss function [20], in which the Deterministic Gradient Descent (DGD) method [12], [19] algorithms are usually utilized to adjust the local parameters.

(ii) Global model broadcasting: At the beginning of the global round i, CU sends the latest version of global model $\mathbf{w}^{(i)}$ to the all associated UEs in set U_s.

(iii) Local training update: In this phase, each UE u that associates with DU s updates the local parameters as,

$$\mathbf{w}_{us}^{(i),j+1} = \mathbf{w}_{us}^{(i),j} - \xi^{(i)} \nabla L_{us}(\mathbf{w}_{us}^{(i),j}|\mathcal{D}_{us}), \forall i \in \mathcal{I}, j \in \mathcal{J}, \quad (4)$$

where the learning set size, $\xi^{(i)} > 0$, is often decreased over the time; and $J = \{0, 1, .., J-1\}$ and $I = \{0, 1, .., I-1\}$ are the sets of I global rounds and J local iterations, respectively.

(iv) Local uploading: When the the local model, $\mathbf{w}_{us}^{(i),j}$, of UE (u, s) is accomplished, it will be sent back to the DU s server via the wireless cellular channels. Practically, the model parameters are sent into the baseband signals via different processes (e.g., modulation, coding, and compression) to preserve transmission reliability. Then, DU s forward the local training model, $\mathbf{w}_{us}^{(i),j}$, to CU for averaging.

(v) Global uploading: After uploading all local tanning models to the CU, CU conducts global training updates as,

$$\mathbf{w}^{(i)+1} = \frac{\sum_{u\in\mathcal{U}s} \sum_{s\in\mathcal{S}} \mathbf{w}_{us}^{(i),j}}{U}. \quad (5)$$

The steps $(i) - (v)$ in the NG-FedAvg algorithm are repeated until convergence. Although the NG-FedAvg algorithm can distributively find the solution for (3), forwarding the local models from DUs to CU can incur the fronthaul link extra traffic load, which is prohibitive in a large-scale NG-RAN. To

Algorithm 1 NG-FedAvg Algorithm

1: Initialize local model weights
2: Initialize CU Aggregation frequency F
3: **repeat**
4: The CU broadcast $\mathbf{w}^{(i)}$ thought DUs to all users
5: Each UE (u, s) computes the local training update according to (4)
6: UEs upload their updated learning model to the associated DUs
7: Each DU forwards the received learning model to CU
8: CU determines the global learning model according to (5)
9: **until** Convergence

tackle this issue, we will propose a FedNG algorithm for local aggregations in the next section. The NG-FedAvg algorithm is detailed in Algorithm 1.

III. FedNG Algorithm Design

In this section, we propose a FL algorithm, named FedNG, to achieve the optimal value of $\mathbf{w}$ in (3), followed by the proof of FedNG convergence.

A. Proposed FedNG Algorithm

One of the major challenges in performing the NG-FedAvg algorithm at the NG-RAN system is the capacity-limited fronthaul constraint. Hence, in Fig. 2, we have detailed the training processes of our proposed FedNG algorithm. In the beginning, at round i, the end-user (u, s) determines J gradient updates on the collected data. In the DU layer, each DU aggregates the collected local gradient models from the associated UEs and forwards these models to CU for global updating. In the CU layer, CU computes from the latest global model, $\mathbf{w}^{(i)}$, the updated global model, $\mathbf{w}^{(i)+1}$, which is later returned to the end-users through the DUs to start a new round. Specifically, we present the key steps of the FedNG algorithm as follows,

(i) Local model update: As mentioned in Sect. II-C, in the NG-FedAvg algorithm, each UE determines its training model by utilizing DGD method [12], [19], which is considered the fronthaul-capacity is consuming in a large-scale NG-RAN scenario. Therefore, in the FedNG algorithm, instead of performing the local training process for each UE in (4), we utilize the Stochastic Gradient Descent (SGD) method to determine the gradient on mini-batches. Accordingly, we assume that $\mathcal{A}_{us}^{(i),j}$ is the mini-bach with size $A = |\mathcal{A}_{us}^{(i),j}|$, which is randomly sampled from end-user (u, s) at round $(i)$of the local iteration j. Hence, UE (u, s) can update the local parameters as,

$$\mathbf{w}_{us}^{(i),j+1} = \mathbf{w}_{us}^{(i),j} - \xi^{(i)} \nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{A}_{us}^{(i),j}), j = 0, 1, .., J-1, \tag{6}$$

where the stochastic gradient can be calculated as,

$$\nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{A}_{us}^{(i),j}) = \frac{\sum_{d \in \mathcal{A}_{us}^{(i),j}} \nabla l(\mathbf{w}_{us}^{(i),j}, \mathbf{x}_d, y_d)}{A}. \tag{7}$$

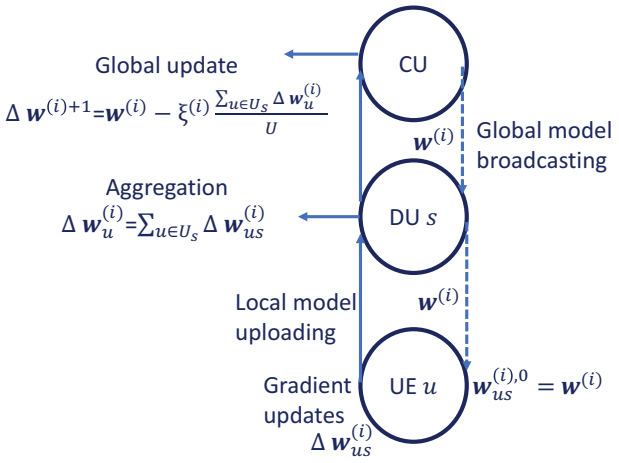


Fig. 2. An completed iteration in the FedNG algorithm.

Hence, the constraint, $\mathbb{E}\{\nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{A}_{us}^{(i),j})\} = \nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{D}_{us})$, should be established to estimate the gradient in (7) properly. Practically, since the only size of $\mathcal{A}_{us}^{(i),j}$ is fixed during the training rounds, the total stochastic updates for each end user can be written as,

$$\nabla \mathbf{w}_{us}^{(i),j} = \sum\nolimits_{j \in \mathcal{J}} \nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{A}_{us}^{(i),j}), \tag{8}$$

where (8) leads to satisfy the condition, $\mathbf{w}_{us}^{(i),J} - \mathbf{w}_{us}^{(i),0} = -\xi^{(i)} \nabla \mathbf{w}_{us}^{(i)}$.

(ii) Local model aggregation: In this step, DU s repeatedly aggregate gradient parameters as,

$$\begin{aligned} \nabla \mathbf{w}_s^{(i)} &= \sum\nolimits_{u \in \mathcal{U}_s} \sum\nolimits_{j \in \mathcal{J}} \nabla L_{us}(\mathbf{w}_{us}^{(i),j} | \mathcal{A}_{us}^{(i),j}) \\ &= \sum\nolimits_{u \in \mathcal{U}_s} \nabla \mathbf{w}_{us}^{(i)}. \end{aligned} \tag{9}$$

(ii) Global model update: After all the DUs forward the aggregated gradient parameters to CU for global update, which can be determined as,

$$\begin{aligned} \mathbf{w}^{(i)+1} &= \mathbf{w}^{(i)} - \frac{\xi^{(i)} \sum_{u \in \mathcal{U}_s} \sum_{s \in \mathcal{S}} \nabla \mathbf{w}_{us}^{(i)}}{U} \\ &= \mathbf{w}^{(i)} - \frac{\xi^{(i)} \sum_{s \in \mathcal{S}}}{U}. \end{aligned} \tag{10}$$

After the aggregated gradient parameters are determined, it will be forwarded back to the end user to begin a new global round. Algorithm 2 shows the details of the proposed FedNG algorithm. It can be observed that our FedNG algorithm does not require end users to send their raw data to DUs and CU. Hence, the FedNG algorithm can secure more privacy for UEs while reducing the overhead fronthaul traffic in NG-RAN.

B. FedNG Convergence

The convergence of our FedNG algorithm is discussed and proven in Theorem 1 as,

Algorithm 2 FedNG Algorithm

Input: I, J, U, S $\mathcal{D}_{us}, \forall u \in \mathcal{U}, s \in \mathcal{S}$
Initialize local model weights
Initialize the global model, $\mathbf{w}^{(0)}$, at CU with learning rate $\xi^{(0)}$
for $i \in \mathcal{I}$ **do**
  $\mathbf{w}^{(i)}$ sent from CU to all DUs
  for $u \in \mathcal{U}_s$ *in parallel* **do**
    for $s \in \mathcal{S}$ *in parallel* **do**
      DU s sends $\mathbf{w}_{us}^{(i),j}$ to each UEs
      for $j \in \mathcal{J}$ **do**
        Each UE samples a new minbach $\mathcal{A}_{us}^{(i),j}$ with size A and calculates $\nabla L_{us}(\mathbf{w}_{us}^{(i),j}|\mathcal{A}_{us}^{(i),j})$ as in (7)
      end for
      Each UE us calculate $\nabla \mathbf{w}_{us}^{(i),j}$ as in (8) and uploads it to the associated DU
    end for
    Each DU s aggregates all the received gradient parameters as in (9)
  end for
  CU calculates global update as in (10)
end for
Output: $\mathbf{w}^{(I)}$

Theorem 1. *Given* $\mathbf{w}^*, \mathbb{E}\{||\mathbf{w}^0 - \mathbf{w}^*||^2\}$, $\Upsilon = \max\{\frac{64\Lambda}{\kappa}, 4G\}$ *and learning step size* $\xi^{(i)} = \frac{16}{\kappa((i)+1+\Upsilon)}$, *the upper bound of our FedNG can be achieved after* I *global bound as,*

$$\mathbb{E}\{||\mathbf{w}^{(I)} - \mathbf{w}^*||^2\} \leq \frac{\max\{\Upsilon^2 \mathbb{E}\{||\mathbf{w}^0 - \mathbf{w}^*||^2\}, (\frac{16}{\kappa})^2 IQ\}}{(I+\Upsilon)^2} \tag{11}$$

where $Q = 2(G\mu)^2 + (2 + \kappa/4\Lambda)(I-1)L\mu^2 + \frac{I\sum_{u\in\mathcal{U}_s}\sum_{s\in\mathcal{S}}\kappa_{us}^2}{U^2} + \frac{6\Lambda I}{U}\sum_{u\in\mathcal{U}_s}\sum_{s\in\mathcal{S}}\epsilon_{us}$

Proof. In order to prove Theorem 1 (i.e., the convergence of our proposed algorithm), we have made several assumptions while doing the analysis.

Assumption 1. *Let* $L_{us}(\mathbf{w}) : \mathbb{R}^f \to \mathbb{R}$ *is* Λ*-smooth and* κ*-strongly convex function [21], i.e.,* $L_{us}(w) \geq L_{us}(\hat{\mathbf{w}}) + \nabla L_{us}(\mathbf{w})(\mathbf{w} - \hat{\mathbf{w}}) + \frac{\kappa}{2}||\mathbf{w} - \hat{\mathbf{w}}||^2, \forall \mathbf{w}, \hat{\mathbf{w}} \in \mathbb{R}^f$. *Also,* $L_{us}(\hat{\mathbf{w}})$ *is an* Λ*-Lipschitz continuous gradient, i.e.,* $||\nabla L_{us}(\mathbf{w}) - \nabla L_{us}(\hat{\mathbf{w}})|| \leq \Lambda||\mathbf{w} - \hat{\mathbf{w}}||, \forall \mathbf{w}, \hat{\mathbf{w}} \in \mathbb{R}^f, \Lambda \geq 0$.

Assumption 2. *(*Bounding the variance*.) Similar to [22], [23], at UE* (u, s), *let* $\mathbb{E}\{\nabla L_{us}(\mathbf{w}_{us}^{(i),j}|\mathcal{A}_{us}^{(i),j}) - \nabla L_{us}(\mathbf{w}_{us}^{(i),j}|\mathcal{D}_{us})\} \leq \alpha_{us}^2 \forall u \in \mathcal{U}_s, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{J}$, *is an upper bound on the variance* α_{us}^2 *and* $\mathbb{E}\{||\nabla L_{us}(\mathbf{w}_{us}^{(i),j}|\mathcal{A}_{us}^{(i),j})||^2\} \leq \mu^2, \forall u \in \mathcal{U}_s, s \in \mathcal{S}, i \in \mathcal{I}, j \in \mathcal{J}$ *is an upper bound on the variance* μ^2.

By holding Assumption 1, we can have,

$$\mathbb{E}\{||\sum_{u\in\mathcal{U}_s}\sum_{s\in\mathcal{S}}\frac{1}{U}(\nabla L_{us}(\mathbf{w}_{us}^{(i),j}) - \nabla \hat{L}_{us}(\mathbf{w}_{us}^{(i),j})||^2\} \leq \frac{\sum_{u\in\mathcal{U}_s}\sum_{s\in\mathcal{S}}\alpha_{us}^2}{U^2} \tag{12}$$

By holding Assumption 2, we can get,

$$\frac{1}{U}\mathbb{E}\{||\hat{\mathbf{w}}^{(i),j} - \mathbf{w}_{us}^{(i),j}||^2\} \leq (J-1)J(\xi^{(i)})^2\mu^2 \tag{13}$$

By holding Assumptions 1 and 2, the expected upper bound of $\mathbf{w}^*$ when $\xi^{(i)} \leq 1/4\Lambda$, can be written as,

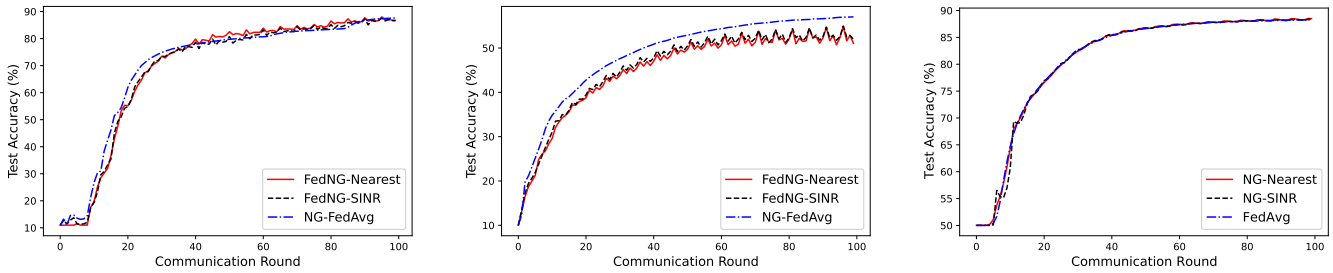
$$\mathbb{E}\{||\hat{\mathbf{w}}^{(i),l+1} - \mathbf{w}^*||^2\} \leq (1 - \frac{1}{2}\kappa\xi^{(i)})\mathbb{E}\{||\hat{\mathbf{w}}^{(i),l+1} - \mathbf{w}^*||^2\} + (\xi^{(i)})^2\Phi^{(i),j} + \frac{2\xi^{(i)}}{U}\sum_{u\in\mathcal{U}_s}\sum_{s\in\mathcal{S}}\mathbb{E}\{(\hat{L}_{us}(\mathbf{w}^*) - \hat{L}_{us}(\hat{\mathbf{w}}^{(i),j}))\}, \tag{14}$$

where $\Phi^{(i),j} = 2 + (\frac{\kappa}{\Lambda})(J-1)\mu^2 + \frac{1}{U^2}\sum_{u\in\mathcal{U}_s}\sum_{s\in\int}\alpha_{us}^2 + \frac{6\Lambda}{U}\sum_{u\in\mathcal{U}_s}\sum_{s\in\int}\tau_{us}$; $\tau_{us} = L_{us}(\mathbf{w}|D_{us}) - L_{us}^*$ is the data heterogeneity factor between one user and the other; and L_{us}^* is the minimum local loss function. Hence, the convergence of our FedNG can be calculated in (11). The proof is complete. □

IV. Performance Evaluation

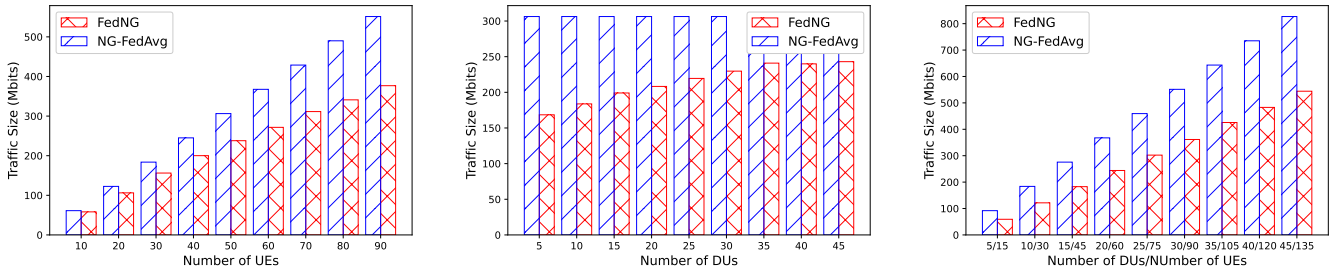
In this section, we evaluate our proposed FL-based scheme on three datasets with three different types of models– MNIST [24] with Dense Neural Networks (DNNs), CIFAR10 [25] with convolutional neural networks (CNNs), and IMDB sentimental analysis dataset [26] with recurrent neural networks (RNNs). We compare the different UE-DU association methods including *nearest*, in which each UE connects to the closest DU, and *best-SINR*, in which each UE selects the one with best SINR from the visible DUs. We also compare these two methods with the baseline model FedAvg.

Simulation Settings: For NG-RAN simulation, we consider randomly scattered UEs with uniformly located DUs. Taking fading channel into account, we assume that large scale fading is the same for all sub-bands and small scale fading is frequency-selective and flat. Define g_{ju} as the channel gain from DU j to UE u and it is determined as, $g_{us} = d_{us}^{ls}|h_{us}|^2, \forall u \in \mathcal{U}, s \in \mathcal{S}$, where d_{us}^{ls} is the large scale fading including pass loss and shadowing, and h_{us} is the small-scale Rayleigh fading. To model the Rayleigh fading, we adopt Jake's model [27] and the small-scale fading is modeled as a first-order complex Gauss-Markov process and the update rule is, $h_{us} = \rho h_{us} + \sqrt{1-\rho^2}e_{us}, \forall u \in \mathcal{U}, s \in \mathcal{S}$, where $\rho = J_0(2\pi f_d T)$ is the correlation between two adjacent fading blocks, J_0 is the zero-order Bessel function of the first kind and f_d is the maximum Doppler frequency. T is the time separation that we re-estimate the channel gain. e_{us} is the channel innovation process and they follow circularly symmetric complex Gaussian distribution. A greater value of ρ means that the channel has changed significantly since the last channel estimation, which could be caused by large T or a rapidly changing f_d. Further, we used three types of models– DNN, CNN, and RNN. The DNN is realized with two layers with [128, 128] units for the layers. The CNN is implemented with three convolutional module, which consists of a 2D convoutional layer, a 2D max-pooling layer and a activation function, with kernel size 3 and stride 1. The RNN model is



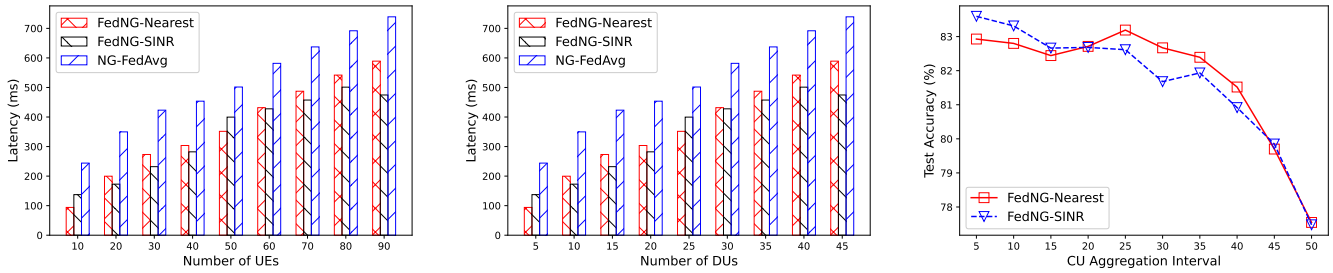
(a) (b) (c)

Fig. 3. Accuracy vs. number of training samples on dataset (a) MNIST with fully-connected neural networks; (b) CIFAR10 dataset with convolutional neural networks; and (c) IMDB sentimental analysis dataset with recurrent neural networks.



(a) (b) (c)

Fig. 4. Total traffic in the network vs. (a) the number of UEs; (b) the number of DUs; and (c) varied number of UEs and DUs with the same DU/UE ratio.



(a) (b) (c)

Fig. 5. (a) Latency against the number of UEs; (b) Latency vs. the number of DUs. (c) Testing accuracy vs. CU aggregation interval. Fig. (c) is generated on MNIST dataset.

realized with a two-layer Long Short Term Memory (LSTM) unit with one fully connected layer as the last layer for the output. The learning rate is 0.01 for all models and the batch sizes are 32, 4, and 128 for three datasets, respectively. The framework is implemented with Python 3.6 and Pytorch [28], a deep learning toolkit.

Comparison of Accuracy: To show how the proposed model work in terms of accuracy, we plot the accuracy change during training in Fig. 3. We can observe that in all three datasets, three models achieve similar performances. Particularly, in Fig. 3(b) FedAvg slightly outperforms the other two models but this advantage is acceptable. We will show later in this subsection that the proposed framework can achieve a much lower latency without reducing a significant level of the accuracy.

Comparison of Traffic: To compare the traffic between CU and DUs needed to update the proposed FL frameworks with other frameworks, we present the results in Fig. 4. We vary the number of UEs involved in the training process in Fig. 4(a) and it is obvious that as the number UEs increase, the traffic size increases as well. Additionally, we can also observe that, in FedAvg framework, more traffic is transmitted than it in NG-FedAvg and the difference is growing larger as the number of UEs increase. In Fig. 4(b), we vary the number of DUs to see how the traffic size changes. For FedAvg, every UE needs to connect to the CU for aggregation and therefore the traffic size does not change. For NG-FedAvg however, the traffic size grows slowly as the number of DU increases, which is because the DU distribution becomes more dense and UEs are connecting to more different DUs. Finally, in Fig. 4(c), we keep the ratio of the number of DUs over the number of UEs unchanged and vary the absolute values of them to see how the traffic sizes will be affected. From the figure, we can observe that the number of UEs seem to dominate the results as the difference between two frameworks grows as the number of UEs and DUs vary.

Comparison of Latency: In Figs. 5(a) and 5(b), we plot the latency perceived by the UEs for all three frameworks. We can observe that FedAvg has a much larger latency than the other two frameworks. This is because that for each communication round in FedAvg, the UE needs to communicate with the centralized CU while, in the proposed framework, UEs only need to exchange information with the DUs which is closer to them physically and thus the latency is reduced.

Impact of CU Aggregation Interval: In the proposed framework, the centralized CU aggregates the models on DUs with a predefined interval. To show how this interval affects the accuracy of the federated learning framework, we generate a corresponding plot in Fig. 5(c). As we can observe that, as the aggregation interval increases, the accuracy of the two models do not change much at the beginning, and then drop dramatically. We can observe a threshold at around 45 after which the performance of the models will drop significantly. Furthermore, we can see that different UE-DU association connection can get quite different results. This figure is generated using the MNIST dataset with 30 UEs.

V. Conclusion

In this paper, we introduced Next Generation Radio Access Networks (NG-RANs) as a promising architecture to satisfy the high on-demand requirements for 5G and beyond applications. To address the main challenging in NG-RAN, the limited fronthaul capacity and privacy, we proposed a Federated Learning (FL)-based NG-RAN algorithm, named FedNG, in which the User Equipment (UEs), as well as NG-RAN infrastructures, help each other throughout the learning and the training process to relieve the burden on fronthaul interface and secure the privacy for end-users. Finally, we carried out numerical simulations using three real-world datasets, MNIST, Fashion-MNIST, and IMDB. The performance of our proposed algorithm showed a significant improvement compared with existing traditional Federated Averaging (FedAvg) in terms of accuracy, service latency, and traffic size.

Acknowledgment: This work was supported by the US National Science Foundation under Grant No. ECCS-2030101.

References

[1] Cisco Visual Networking Index, "Cisco annual internet report, 2018–2023," *Cisco white paper, USA*, 2020.

[2] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, 2018.

[3] A. Younis, B. Qiu, and D. Pompili, "Latency-aware hybrid edge cloud framework for mobile augmented reality applications," in *Proc. IEEE SECON*, pp. 1–9, 2020.

[4] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, 2016.

[5] ETSI, "NG-RAN: Architecture description," *3GPP TS 38.401 Ver. 15.2.0 Release 15*, 2018.

[6] I. Chih-Lin, Y. Yuan, J. Huang, S. Ma, C. Cui, and R. Duan, "Rethink fronthaul for soft RAN," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 82–88, 2015.

[7] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *EEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 2019.

[8] H. Yang, Z. Xiong, J. Zhao, D. Niyato, C. Yuen, and R. Deng, "Deep reinforcement learning based massive access management for ultra-reliable low-latency communications," *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 2977–2990, 2020.

[9] A. Younis, B. Qiu, and D. Pompili, "QLRan: Latency-quality tradeoffs and task offloading in multi-node next generation RANs," in *Proc. IEEE WONS*, pp. 1–8, 2021.

[10] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrAIn: Deep learning based orchestration for computing and radio resources in vRANs," *IEEE Trans. Mobile Comput.*, 2020.

[11] A. Younis, T. X. Tran, and D. Pompili, "Bandwidth and energy-aware resource allocation for cloud radio access networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6487–6500, 2018.

[12] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[13] O. A. Wahab, A. Mourad, H. Otrok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2021.

[14] X. Zhang, Y. Liu, J. Liu, A. Argyriou, and Y. Han, "D2D-assisted federated learning in mobile edge computing networks," in *Proc. IEEE WCNC*, pp. 1–7, 2021.

[15] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE ICC*, pp. 1–6, 2020.

[16] Y. Cao, S.-Y. Lien, Y.-C. Liang, and K.-C. Chen, "Federated deep reinforcement learning for user access control in open radio access networks," in *Proc. IEEE ICC*, pp. 1–6, 2021.

[17] R. Jin, X. He, and H. Dai, "Communication efficient federated learning with energy awareness over wireless networks," *IEEE Trans. Wireless Commun.*, 2022.

[18] H. Chen, S. Huang, D. Zhang, M. Xiao, M. Skoglund, and H. V. Poor, "Federated learning over wireless IoT networks with optimized communication and resources," *IEEE Internet Things J.*, 2022.

[19] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, 2019.

[20] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Mag.*, vol. 37, no. 3, pp. 50–60, 2020.

[21] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge U. Press, 2004.

[22] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. ICLR*, pp. 1–17, 2019.

[23] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning," in *Proc. AISTATS*, pp. 1–11, 2021.

[24] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, 2012.

[25] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[26] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, (Portland, Oregon, USA), pp. 142–150, Association for Computational Linguistics, June 2011.

[27] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 458–461, 2017.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances Neural Inf. Process. Syst.*, vol. 32, 2019.