Federated Learning Empowered Mobility-Aware Proactive Content Offloading Framework for Fog Radio Access Networks

Sanaullah Manzoor[a,b*], *Adnan Noor Mian*[a], *Ahmed Zoha*[b], *and Muhammad Ali Imran*[b]
[a]*Department of Computer Science, Information Technology University, Lahore, Pakistan.*
[b]*School of Engineering, University of Glasgow, Glasgow, United Kingdom.*

ABSTRACT

Proactive content caching has emerged as a promising solution to cope with exponentially increasing mobile data traffic. The popular user contents can be cached near the network edge for faster retrieval and processing. Current state-of-the-art approaches adopt a centralized model training mechanism that requires high communication and data exchange overheads to predict content popularity. Moreover, these approaches fail to deal with the dynamicity of the environment since they do not take into account the users' mobility information and are unable to incorporate content offload timings. In this paper, we address these limitations by proposing a novel federated learning-based Mobility and Demand-aware Proactive Content Offloading (MDPCO) framework. MDPCO exploits distributed learning strategies and capitalizes on users' mobility and demand information for proactive content offloading. Extensive simulations are carried out to validate the efficacy of MDCPO against local and cloud-based models. Our proposed model yields an average performance improvement of 6.7% in comparison to the cloud-based model. Furthermore, with the increase in the number of fog servers, the MDPCO achieves a 9.8% higher data offloading ratio and 1.18% increase in the downlink rates while being more energy-efficient than cloud-based approaches.

1. Introduction

With the vast deployment of the 5G and beyond wireless networks, Internet of Things (IoT) services and data-hungry applications have become a significant contributor to Internet traffic growth. Prominently, IoT-based data streaming services are growing with an annual growth rate of 69% [1]. The rapid advancement in next-generation communication networks supports the accessibility and deployment of IoT services more than before. It is estimated that nearly 82% of the Internet traffic is originated from network streaming applications [2]. By the end of 2022, more than 50 billion devices would be connected to the Internet, and each mobile user would generate 60 GB of data per month [3]. Such a tremendous amount of data flow will pose a significant challenge to cellular and IoT service providers with the existing network infrastructure and may lead to core network overloading. To tackle this ever-increasing traffic burden, fog radio access networks (F-RANs) have been proposed due to their vital admissible features, such as edge content caching, information processing, and communication capabilities [4, 5].

Content caching has been considered as a viable solution to alleviate backhaul congestion, enhances system capacity, and reduces the end-to-end service delays [6, 7, 8, 9]. Traditional content caching methods such as least recently used (LRU) and first-in-first-out (FIFO) fail to account for future content popularity, resulting in low cache efficiencies [10]. In this regard, various content caching schemes have been proposed to predict future content popularity [11, 12, 13, 14, 15, 16, 17]. To the best of the authors' knowledge, these existing works fall short of the best network services mark due to the following limitations. Firstly, the future content popularity prediction approaches usually operate in a centralized manner which requires a large amount of local data transfer to the central server. Secondly, these schemes provide the solutions for, *which* contents to be offloaded, and ignore *when* to offload the predicted contents due to lack of users' mobility information [11, 13]. Thirdly, as the number of users and data grow, the centralized models may face scalability issues. In short, these centralized schemes have significant communication exchange overheads; a lower cache hit under high user mobility; and are vulnerable to data privacy violations [18].

In order to tackle the above mentioned limitations, federated learning (FL) can be seen as a promising solution to enable decentralized future content predictions [19]. In FL, edge devices collaborate to train a local model based on locally available data and share their weights to form a global model, instead of uploading large volumes of raw data at the remote server [20]. In this paper, we demonstrate the application of FL for the F-RANs content offloading systems. We define communication latency and data storage overheads under centralized and FL-based training models. In the centralized training, base stations (BSs) send data to the fog server, where model inference and cache decisions took place, then eventually shared with the corresponding BSs. Figure 1 shows a network with a cloud server supervising multiple BSs and several users, the arrow width corresponds to the amount of shared data exchange between the entities. Figure 1(a) represents the centralized learning scenario in which the overall latency, during content demand learning, consists of (i) time during which the raw data is transferred from BS to fog server t_{bf}, (ii) time during for

ORCID(s):
*sanaullah.manzoor@itu.edu.pk

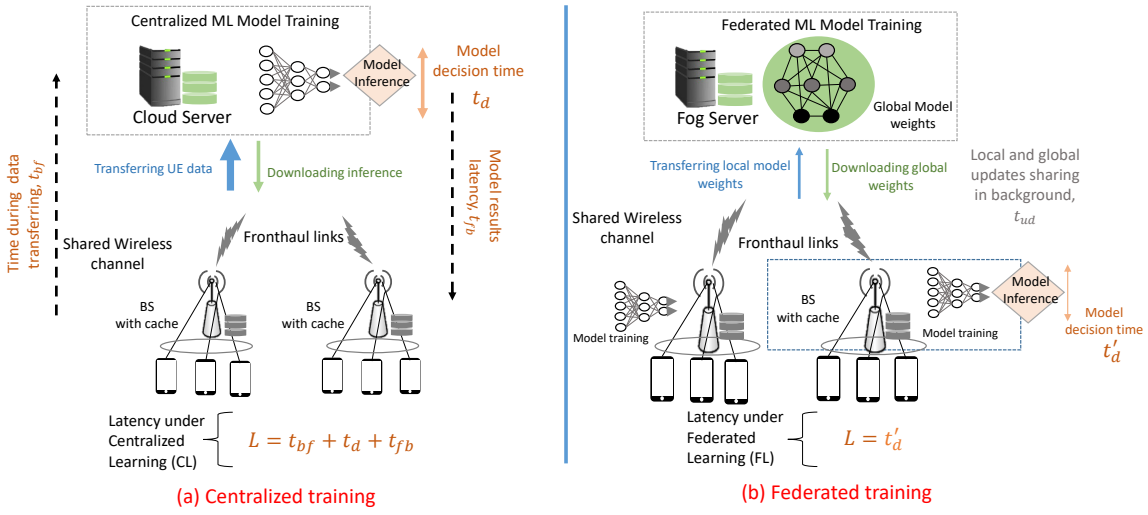

Figure 1: A motivating example for comparison of centralized and federated learning based content caching

model training and inference t_d, and (iii) the time required to share the model results towards the corresponding BSs t_{fb}. Thus, the overall communication delay under centralized model training can be represented as $L = t_{bf} + t_d + t_{fb}$. In contrast the FL scheme as shown in Figure 1 (b) trains the model based on the local data only and shares model updates with the fog sever where global weight aggregations are performed. During each communication round, a BS downloads the global weights and starts the next cycle of local training. The local updates sharing and global weight downloading process repeats in the background which requires t_{ud} time. In the case of centralized learning, whenever a prediction request is made, the data is transferred to the cloud server where predictions are made available, and then these predictions are sent to the corresponding BS. On the other hand, in case of FL the BS model fulfils these requests locally and requires only t'_d time. In short, under FL t'_d time would be the only impactful latency which can be represented as $L = t'_d$. Furthermore, the latency during model training and inference under FL would be less due to the small cache matrix size. In contrast, cloud server requires more computation time for the learning decision due to larger data volumes, i.e., $t'_d < t_d$. Thus, the overall the content demand prediction latency under FL would be significantly lower than the centralized learning.

In this paper, we propose a novel FL-based Mobility and Demand-aware Proactive Content Offloading (MDPCO) framework with cache-enabled F-RAN architecture. In MDPCO, local BSs estimate users' mobility and future content demand based on local data. The local mobility model is formulated using a long short-term memory (LSTM) network that takes joint information of the user's cell-IDs and sojourn times and predicts the next cells. We employee deep neural networks (DNNs) for the local content prediction model to predict the future content probabilities. The fog server aggregates local mobility and content prediction model updates to construct a global model. An optimization problem is formulated using joint coordination of federated mobility and content models to offload future contents. In terms of deep-learning architecture, mobility and content caching models are kept simplistic to support minimum computation load and higher energy efficiency. Our contributions are listed as follows:

- We propose a novel FL-based MDPCO framework that considers mobility and content demand statistics jointly to maximize users' quality of experience.
- We formulate a joint content offloading scheme that proactively exploits federated mobility and content score prediction models to cache the most likely future contents.
- We perform an extensive set of simulations to validate the effectiveness of the proposed MDPCO framework and compared it with state of the art content cache schemes.

The rest of the paper is organized as follows: Section **2** reviews state-of-the-art proactive content offloading schemes. Section **3** discusses the proposed system model and the problem formulation for the federated mobility and caching in wireless networks. The proposed mobility and demand-aware federated content offloading framework has been discussed in Section **4**. The results of the proposed MDPCO scheme have been presented in Section **5** followed by a discussion and insights in Section **6**. Lastly, Section **7** concludes this work.

2. Related Work

In this section, we highlight the state-of-the-art cloud-based and FL-based content offloading schemes for F-RANs. Subsequently, we provide a comparative analysis of cloud versus FL-based content offloading schemes, highlighting the gaps and motivation for our proposed approach. Table

Table 1
State of the art content offloading approaches in F-RANs

Paper	Approach Mode	Objectives	Mobility	Content-Aware	Summary
Rehman et al. [21]	Cloud	Network delay	No	No	Fronthaul load-aware caching scheme to reduce content-service delay using loosely-couple F-RANs
Shnaiwer et al. [22]	Cloud	Load balancing	No	No	An opportunistic network-based coding based caching strategy to offload cloud base station load
Sun et al. [16]	Cloud	Load balancing	No	No	Content offloading scheme to alleviate the load of central resource manager that is based-on the statistics of user requests and channel gains.
Fan et al. [14]	Cloud	QoE	No	Yes	Q-learning based caching scheme in fog-enabled cellular networks to predict future content popularity
Feng et al. [13]	Cloud	QoE	No	Yes	Content popularity prediction scheme by exploiting deep learning model to build a content prediction classifier for every content class
Shi et al. [17]	Cloud	Load Balancing	Yes	No	A reinforcement learning-based cache placement and user association scheme to alleviate fronthaul load.
Jiang et al. [23]	Cloud	Network delay	No	Yes	A cloud-based coded caching and content offloading mechanism for F-RANs
Yan et al. [17]	Cloud	QoE	Yes	No	A joint machine learning based content placement and user association scheme for traffic offloading.
Wang et al. [24]	Federated	Network delay	No	No	A FL-based-edge caching scheme in F-RANs to reduce communication overheads.
Cui et al. [25]	Federated	QoE	No	No	Block-chain enabled federated caching mechanism for proactive content offloading scheme.
Tuo et al. [26]	Federated	QoE	No	Yes	A FL-based content prediction model for F-RANs.
Xiao et al. [26]	Federated	Energy Efficiency	No	Yes	Secure FL-based content placement and cache allocation scheme for F-RANs.
Xue et al. [27]	Federated	Security	No	No	A FL-based resource-constraint edge caching scheme for E-health systems.
Yu et al. [28]	Federated	QoE	Yes	No	Mobility-aware FL-based content offloading in vehicular networks.
Cheng el al. [29]	Federated	QoE	No	Yes	Blockchain enabled FL-based content caching framework for D2D networks
Proposed MDPCO	Federated	QoE	Yes	Yes	FL-based MDPCO framework that takes into account users' mobility and demands statistics jointly to proactively offload the predicted contents

I summarizes related content offloading strategies in terms of key objectives, mode of operation, with or without mobility, demand-context and summary of contributions.

2.1. Cloud-based Learning Approaches for Content Offloading in F-RANs

Conventional machine learning approaches have been extensively researched for content offloading in F-RANs. These studies focused on formalizing the centralized content fetching problem and optimizing downlink rates, energy efficiency and cache storage sizes. Here, we report the most recent studies for centralized content offloading in F-RANs. Jiang et al. proposed a cloud-based coded caching and content offloading mechanism to minimize network delay [23]. Rehman et al. proposed a fronthaul load-aware caching scheme to reduce content-service delay by leveraging loosely coupled FRAN architectures [21]. Shnaiwer et al. proposed opportunistic network coding-based caching strategy to offload BS load in the heterogeneous F-RANs [22]. Shi et al. proposed reinforcement learning-based cache placement and user association scheme to reduce fronthaul load [17]. Jinag et al. introduced a graph-based cooperative content caching scheme using joint content clustering and placement in F-RANs [15]. Sun et al. proposed a content offloading scheme for F-RANs to reduce the burden on central resource management, in which a network long-term utility function-based optimization problem is defined utilizing users' request data and channel gains [16]. Fan et al. proposed a Q-learning based caching scheme in fog-enabled cellular networks to predict users' future content popularity. The content caching algorithm is developed using users' content preferences and popular topics information [14]. Yan et al. proposed a joint content placement and user association in F-RANs in which machine learning models predict the popularity of unknown contents, and an optimization problem for joint caching and user association is formulated. [17]. Feng et al. proposed a content popularity prediction scheme by using a deep learning model to create a content prediction classifier for each content class [13]. Recently, Li et al. proposed a latency-aware content caching scheme by utilizing user's mobility and cache capacity [30]. The major issue with all these cloud-based approaches is their inability to scale as the number of users increases in the network.

2.2. Federated Learning-based for Content Offloading in F-RANs

Recently, much attention has been paid to FL-based content offloading schemes. We now discuss the most relevant FL-based content offloading strategies. Wang et al. proposed a FL-based edge caching scheme to reduce communication overheads [24]. Cui et al. presented a block-chain enabled federated caching system for proactive content offloading. They trained local models and shared compressed model updates with the global model [25]. For F-RANs, Xiao et al. have presented a FL-based content placement and cache allocation system [26]. For E-health systems, Xue et al. presented a FL-based resource-constraint edge caching approach [27]. Xiao et al. proposed reinforcement learning-

based edge content offloading which focused on the signal jamming and interference [31]. Similarly, Cheng el al. proposed a block-chain enabled FL-based content caching framework for D2D networks [29]. Qi et al. proposed block-chain-enabled FL-based scheme for traffic flow prediction [32]. All of the above mentioned approaches ignore mobility patterns during content offloading. Yu et al has proposed a mobility-aware content offloading mechanism [28]. Nonetheless, this approach has two significant flaws. Firstly, it assumed that mobility sequences follow truncated Gaussian distribution, which is not an accurate representation of users' mobility. In reality, the mobile users' mobility is diverse, irregular, and dynamic. Secondly, the approach ignored users' content demand statistics for content offloading.

In summary, none of the related works proposed a federated machine learning-based framework that considers users' mobility and content demand statistics for proactive content offloading. The existing content offloading approaches mainly operate in a centralized manner. A large volume of local data is required to upload at the central server, which leads to higher communication and data exchange overheads. Furthermore, in the absence of the user's next cell mobility information, the schemes [24, 25, 26, 27, 11, 13, 31] are unable to incorporate content offloading timings and hence they can not deal with the dynamicity of the environment. Additionally, if the number of users and data grow, the centralized content offloading schemes may face scalability issues. Therefore, the current content offloading strategies in F-RANs possess the following prominent drawbacks:

- Operate in a fully centralized manner, leading to higher communication and data exchange overheads as they require large volumes of local data to be uploaded at the central server.
- Unable to incorporate content offloading timings due to lack of users' next cell mobility information.
- May result significant communication overhead during uploading of users' data to the central server.

In order to counter the aforementioned issues, we have come up with a new FL-based MDPCO framework that considers mobility and content demand statistics to offload users' future content. The MDPCO framework works in a distributed manner and restricts data locally, reducing communication overheads.

3. System Model

This section presents the system model for the proposed federated machine learning-based mobility and demand-aware proactive content offloading (MDPCO) framework in cache-enabled F-RANs. he proposed framework consists of several base stations (BSs) supervised by the fog server and the core network that shares information and services via fog servers. The system model is shown in Fig 2.

We consider a F-RAN architecture in which dense B cells $b = \{1, 2, ..., B\}$ are deployed under a fog-server ς. In this system, it is considered that there is no overlapping area between cells, and at a time each user is connected to only one cell. Figure 2 shows our system model. Furthermore, we consider that system has core network functionalities in which fog servers are connected to evolved packet core layer (EPC) i.e. mobility management entity (MME) and serving gateway (S-GW). BSs are also interconnected via $X2$ links that allow them to communicate with each other and share handover information [34]. Here, we define the following entities: let $u = \{1, 2, ..., U\}$, $\mathcal{Z}$, and $\mathcal{W}$ and $u = \{1, 2, 3, ..., U\}$ is the number of users, network bandwidth and the bandwidth of each physical resource block (PRB) respectively. Let $\mathbf{g} = \left[g_1, g_2, ..., g_B\right] \in \left\{0, \mathbb{Z}^+\right\}$ shows the backhaul link capacities between the fog-server and cells. Furthermore, our file library is expressed as $\mathbf{f} = \left[f_1, f_2, ..., f_F\right]$, where each file is atomic and of size υ_i, where files size vector is denoted by $\mathbf{v} = \left[\upsilon_1, \upsilon_2, ..., \upsilon_F\right] \in \mathbb{Z}^+$ [35]. Moreover, the terms cell and base station (BS), and content and file are interchangeably used throughout the paper. Table 2 shows important notations. If $\mathcal{P}_b$ and $\mathcal{H}_{bu}$ represents BS transmission power and channel gain between BS b and UE u, then the signal to interference and noise ratio (SINR) is as follows:

$$\eta_{bu} = \frac{P_b \mathcal{H}_{bu}}{\sum_{j \in B/\{b\}} P_j \mathcal{H}_{ju} + \gamma^2} \tag{1}$$

where γ^2 represents noise power per PRB, which is additive white Gaussian noise (AWGN). For a given P_b the maximum downlink (DL) achievable rate from the cell b to user u is expressed using Shannon theorem:

$$R_{bu} = \mathcal{W} log_2 \left(1 + \frac{P_b \mathcal{H}_{bu}}{\sum_{j \in \mathcal{B}/\{b\}} P_j \mathcal{H}_{ju} + \gamma^2}\right) \tag{2}$$

where the term $\mathcal{W}$ is usually $180kHz$ in an Orthogonal Frequency Division Multiple Access (OFDMA) based cellular

Table 2
Table of Important Notations

Notation	Description
u	User equipment (UE)
b	Base station/Cell
B	Number of unique (distinct) cells in network
ς	Number of fog servers
$\mathcal{Z}$	Total system bandwidth
$\mathcal{W}$	Bandwidth of a PRB
$\mathbf{g}$	Backhaul link capacities
$\mathcal{H}_{bu}$	Channel gain between user u and cell b
P_b	Transmit power of cell b
η_{bu}	Signal to interference and noise ratio (SINR)
r_{bu}	Down-link achievable rate
$\mathbf{f}$	File library
$\mathbf{o}$	Cell storage capacities
$\mathbf{x}$	Tuple representing cell-IDs and sojourn time
$\Psi_{b,u}$	Next cell probability of user u
$\mathbf{v}$	Files size vector
e_{bu}	Effective downlink rate of u
γ^2	Noise power of a PRB
σ	Sigmoidal activation function
$\mathbf{W_m}$	Mobility model trained local weights
$\mathbf{W_c}$	Caching model trained local weights
$\mathbf{C}_b$	Caching decision matrix
Ω	Data offloading ratio
τ	Downlink rate CDF

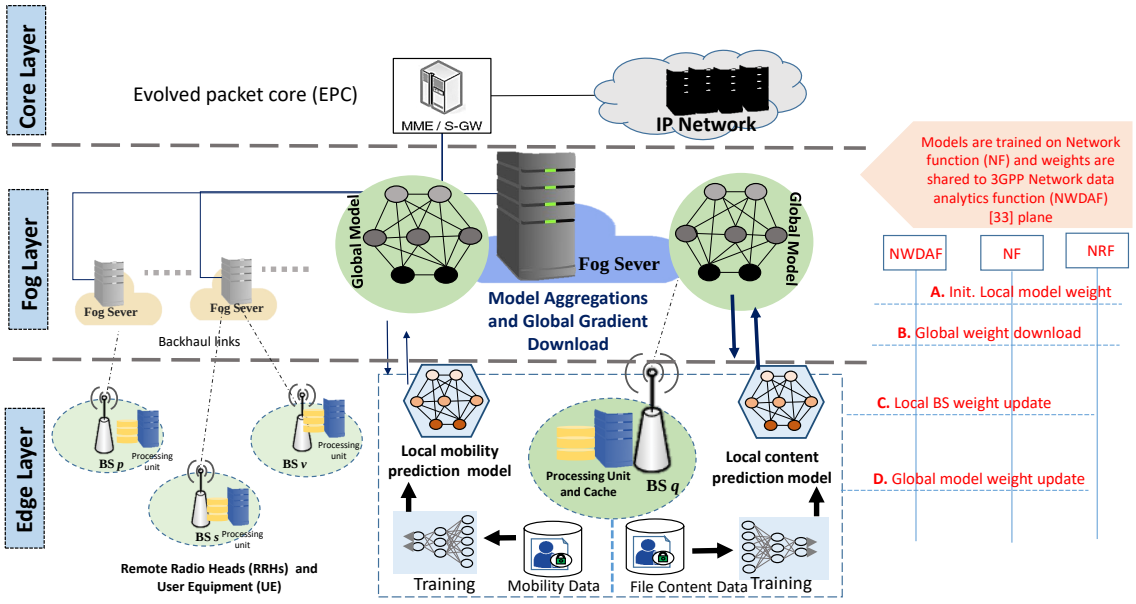

Figure 2: Proposed federated learning based framework. BS trains local mobility and future content prediction models using local data. Local models download global updates from the global model (hosted at fog model) for next communication round. The proposed framework works in-line with 3GPP's release for data analytics and machine learning support NWDAF [33].

system. If user u is associated to cell b then its load at cell b is given by $y_{bu} = \frac{\delta_u}{R_{bu}}$, while the term R_{bu} is maximum downlink rate and δ_u is actual receiving rate from cell b given channel conditions. Further, if user u is associated to cell b then user-cell association is represented by x_{bu}. Thus, by utilizing user's load and the user-cell association information the overall load of cell b will be $l_b = \sum_{u \in U} x_{bu} y_{bu}$. Actually, the DL rate offered by cell b is equally shared among all users U_b, thus the actual receiving rate not only affected by channel conditions but also depends on cell load l_b [36]. Therefore, user's perceived DL rate is $r_{bu} = R_{bu}/l_b(t)$. Moreover, $\mathbf{o} = [o_1, o_2, ..., o_B] \in \mathbb{Z}^+$ vector represents the storage capacities of cells to store the caching information. Now, let's consider a matrix $\mathbf{R}$ that expresses users' downlink rates from the corresponding cells, which is given by:

$$\mathbf{R} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_B \end{bmatrix} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,U} \\ r_{2,1} & \cdots & r_{2,U} \\ \vdots & \vdots & \vdots \\ r_{B,1} & \cdots & r_{B,U} \end{bmatrix} \in \{0, \mathcal{Z}^+\}^{B \times U} \quad (3)$$

4. Federated Learning-based Mobility and Demand-Aware Proactive Content Offloading Framework

Under our system model, each BS can track users' statistics, including handover time, cell-ID, and content demand information, i.e. the number of requests and the request nature [37] [38, 39]. It is assumed that each BS has enough computation resources including cache memory to execute data-intensive machine learning models. Each BS can train its own mobility and content prediction models based on its local data. At some time interval t', these local training results are reported to the fog server, where the global model is updated accordingly. The results of the improved global model are sent back to BSs, where a new round of local training is initiated. Finally, the improved local models are constructed at each BS via model updates and collaborative information exchange between BS and fog server. The output of the federated mobility and caching algorithms are fed to the joint optimization model to offload future contents proactively. The proposed MDPCO framework architecture as shown in Figure 2 adopts the 3rd generation partnership project (3GPP) release 16 support for machine learning-based data-driven optimization [33]. 3GPP provides data collection and analytics support in the 5G network architecture known as network data analytics framework (NWDA). Basically, MDPCO framework can invoke NWDA functions to provide these two core functionalities; first, data collection from network functions (NFs) i.e., local data processing, and secondly, data analytics and future content and mobility predictions via FL. In the proposed MDPCO framework, users and BSs constitute edge layer and BSs act as FL nodes, and the fog server is responsible for the global weight averaging. Specifically, the proposed MDPCO framework serves the following key objectives:

1. **Federated Mobility Prediction Model:** A federated mobility prediction model is built using users' mobility history (base station ID, next cell handover time) to predict the next cells.
2. **Federated Content Prediction Model:** Users' content demand information is used to build a federated future content prediction model that caches future files.
3. **Joint Coordination of Mobility and Content Prediction Models:** The MDPCO framework coordinates

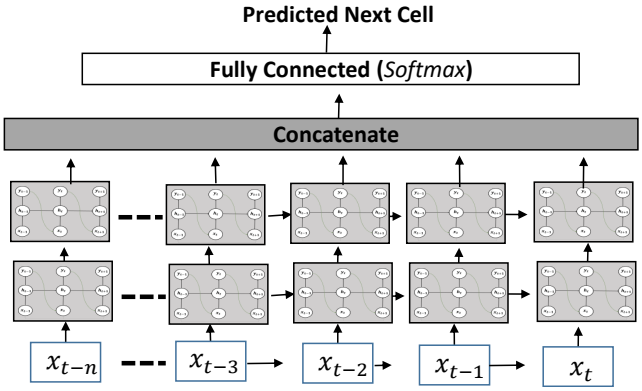

Figure 3: The proposed LSTM-based local federated next cell mobility prediction model

the mobility and content prediction models to proactively offload future contents at the network edge.

4.1. Federated Next Cell Mobility Prediction

This section explains the proposed federated next cell mobility model based on users' mobility cell sequences. The proposed long short-term memory network (LSTM)-based next cell mobility prediction model takes input sequence consisting of cell-IDs and sojourn times $(x_t, x_{t-1}, x_{t-2}, ..., x_{t-n})$ at current time t, and calculates hidden output h_t through previous hidden information h_{t-1} and x_t. Hidden layer outputs are propagated to output layer y_t which has same number of neurons as number of classes in the data. In our case, the number of classes are equal to total the number of cells B. The proposed LSTM-based model architecture is shown in Figure 3. In our model first LSTM layer with 12 neurons encodes the input and propagates to next LSTM layer which has 8 neurons. The output of second LSTM layer is passed to fully connected layer. *Sigmoid* function is used as hidden layer activation function. The fully connected layer has number of neurons equals to the number of unique cells in the data i.e., $|B|$ is number of unique classes. LSTM model uses *softmax* function as classification layer to project output vector into next cell probability vector. The network is trained to minimize the categorical cross-entropy loss stochastic gradient descent with ADAM optimizer [40]. The output of LSTM model is the next cell probability vector $\mathbf{Y} = \left[\hat{y_1}, \hat{y_2}, ..., \hat{y_B}\right]$, that is expressed as:

$$\mathbf{Y} = softmax(\mathbf{z})_i = \frac{e_i^z}{\sum_{b=1}^{B} e_c^z} \tag{4}$$

Each local LSTM-based mobility prediction model trains on the local data, and after each communication round t, the local model shares its trained weights with the fog-based mobility prediction model for model aggregation. Each local mobility model uploads learned weights after each communication round; the fog-based LSTM model aggregates these weights and formulates a global rating score prediction model. Each local BS downloads the global model from the

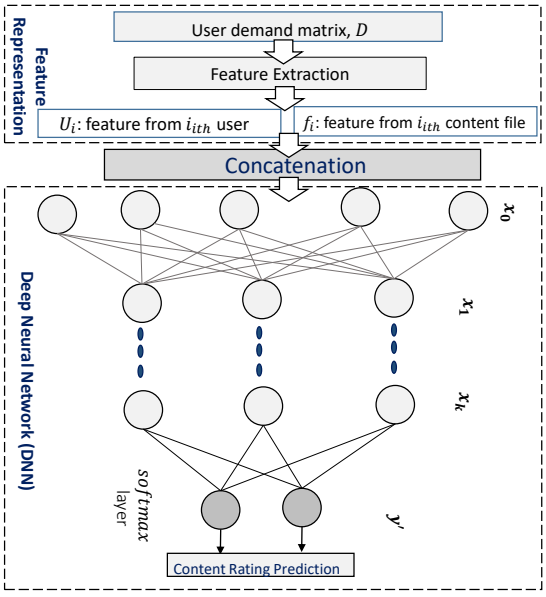

Figure 4: The proposed deep neural network-based federated future content prediction model

fog-server and starts its local LSTM model training based on local mobility data. The local model gets batch-size θ of local data and trains up-to e epochs. Next, at communication round $t = 1, 2, 3...\alpha$ the learned weights from each local LSTM model $\mathbf{W}_t^1, \mathbf{W}_t^2, \mathbf{W}_t^3, ..., \mathbf{W}_t^B$, are uploaded to fog-server for global model aggregation. The weight updates can be written as $\mathbf{V_t^b} = \mathbf{W_t^b} - \delta\mathbf{W}$. The fog server aggregates all the updates from local model to improve global model via *Federated Averaging* mechanism which is given by [41]:

$$V_t^F := \frac{1}{n_t} \sum_{i \in B} V_t^i \tag{5}$$

$$W_{t+1}^b = W_t^b + \eta_t V_t^b \tag{6}$$

where η represents the learning rate. At time t each local model downloads weight updates and finally the improved local model predicts the next future cells with the probability which is expressed as:

$$\Psi(t) = argmax\left[\hat{y_1}, \hat{y_2}, ..., \hat{y_B}\right], \forall b \in B \tag{7}$$

where B is total number of cells and function *argmax* returns maximum value from the vector Y'.

4.2. Federated Demand-Aware Future Content Prediction

A federated machine learning based future content prediction model is proposed using local BS and fog-based content prediction models. The proposed model exploits the contextual information between user and its previously requested files to predict future contents. Let a user u requests various contents f from library $\mathbf{f}$ an proposed content prediction model constructs user's content demand matrix $\mathbf{D} \in$

$\mathbb{R}^{U\times F}$. Each entry $d_{i,j}$ is file rating score from i_{th} user for the j_{th} item;if the score does not exist, then $d_{i,j} = 0$. The most requested contents in matrix $\mathbf{D}$ will be maximally rated while the least demanded or not demanded contents will least or zero rated. Now, demand matrix $\mathbf{D}$ is represented as:

$$\mathbf{D} = \begin{bmatrix} D_1 \\ D_2 \\ : \\ D_U \end{bmatrix} = \begin{bmatrix} d_{1,1} & ... & d_{1,F} \\ d_{2,1} & ... & d_{2,F} \\ : & : & : \\ d_{U,1} & ... & d_{U,F} \end{bmatrix} \in \left\{0, \mathbb{Z}^+\right\}^{U\times F} \quad (8)$$

Actually $F \gg U$, infect each user can not demand for every file from $\mathbf{f}$ library. therefore network can rate only the demanded files. Thus, matrix $\mathbf{D}$ would be spare with many 0 entries. For proactive content caching, there is need to determine full content score/rating matrix $\mathbf{D}$. Thus, consider that users' latent feature matrix $\mathbf{M} \in \mathbb{R}^{U\times a}$ in which the vector $\mathbf{M}_i$ represents i_{th} user's features and a represents dimensions of vector $\mathbf{M}_i$. Similarly, we have a content item's latent feature matrix $\mathbf{N} \in \mathbb{R}^{F\times b}$ with features vector $\mathbf{N}_i$. Further, we encoded vectors of matrix $\mathbf{M}$ and $\mathbf{N}$ such that:

$$M_i = NN\left(OneHot(i)\right) \quad (9)$$

$$N_j = NN\left(OneHot(j)\right) \quad (10)$$

where $OneHot(i)$ denotes One-Hot encoding of vector $\mathbf{M}_i$ and $NN(x)$ represents output of neural network. In order to estimate full demand matrix $\bar{D}$ we used deep neural network-based (DNN)-based model that takes users and items latent features as input and predicts file rating scores. Figure 4 shows the proposed future content model's architecture. In the proposed content prediction model, the input vector $\mathbf{x}_0$ is formulated using concatenation of latent feature vectors and which is given below:

$$x_0 = concatenate\left(\mathbf{M}_i, \mathbf{N}_i\right) \quad (11)$$

where *concatenate()* function concatenates $\mathbf{M}_i$ and $\mathbf{N}_i$ vectors and $\mathbf{x}_0$ is propagated to first hidden layer and is given as:

$$x_1 = ReLU\left(\mathbf{W_1}x_o + v_o\right) \quad (12)$$

where the terms $\mathbf{W_1}$ and v_o represents the weight matrix between the input and first hidden layer, and the bias vector respectively. *ReLU()* denotes the activation function of the hidden layer, which is used to introduce non-linearity in the neural network. At the output layer of DNN model we exploit *softmax() classification function to predict users' content rating score* $d_{i,j}$, and is given by:

$$Y^{'} = softmax\left(\mathbf{W_o}x_h + v_{out}\right) \quad (13)$$

where x_h, $\mathbf{W_o}$, and v_{out} represent output of hidden layer, weight matrix, and bias vector of output layer.

We formulated user's future rating prediction problem as a supervised learning problem and the a deep neural network (DNN) based local model is proposed which has an input layer followed by four hidden layers with the number of neurons 100, 50, 20 and 10 respectively. Figure 7 shows the proposed neural network based user's content rating prediction model. At input layer users' latent feature vector $\mathbf{M}_i$ and files' latent feature vector $\mathbf{N}_i$ is fed into neural network after concatenation as explain in 11. The input is then propagated to hidden layers and, eventually, at the output layer *softmax()* classification function is used to predict the user's rating score $d_{i,j}$. For prediction error minimization, a categorical cross-entropy loss function is used. Each local DNN-based rating prediction model trains on the local data, and after each communication round t, the local model shares its trained weights with the fog-based rating prediction model for model aggregation. Each local model uploads learned weights after each communication round; the global DNN model aggregates these weights and formulates a global rating score prediction model. These weights aggregations help to learn model generalization and also avoid data privacy violations as data is not uploaded to the fog-server.

The overall federated machine learning is preformed in such a way that initially each local BS download global model from the fog-server and starts local DNN model training based on data content history data. The local model gets batch-size θ of local data and trains up-to E epochs. Next, at communication round $t = 1, 2, 3...\alpha$ the learned weights from each local DNN model $\mathbf{W}_t^1, \mathbf{W}_t^2, \mathbf{W}_t^3, ..., \mathbf{W}_t^B$, are uploaded to fog-model for global model aggregation. The weight updates can be written as $\mathbf{V_t^b} := \mathbf{W_t^b} - \delta\mathbf{W}$. The fog server aggregates all the updates from local model to improve global model via *Federated Averaging* using 5 and 6. Each local model downloads weight updates and finally, the improved local model predicts i_{th} user's rating score $d_{i,j}^{'}$ on the j_{th} file which is given as follow:

$$d_{i,j}^{'} = argmax\left[\hat{y_1}, \hat{y_2}, ..., \hat{y_K}\right], \forall k \in K \quad (14)$$

where K is total number of classes in the rating matrix $\mathbf{D}$ and function *argmax* returns maximum value from the vector $Y^{'}$.

4.3. Joint Coordination of Federated Mobility and Caching Models for Content Offloading

Now, the joint coordination of federated prediction (mobility and content) models is formulated to enable mobility-aware caching placement in F-RANs. We aim to offload the predicted contents proactively before cell congestion takes place. It is found in [9] that the cell load fluctuates due to users' mobility which can be leveraged to enable the system's proactive response for content caching and offloading accordingly. The fog server exploits user mobility sequences (x_t, x_{t-1}, x_{t-2}, ..., x_{t-n}) for Eq. 7 to seek user's next cell association for $\Psi(t + t^{'})$ where $t^{'} = [1, \infty)$ is next time interval (*minutes*). Similarly, fog server constitutes the next cell probability matrix Υ for all users in the set $u = \{1, 2, .., U\}$

using Eq. 7, and it is given as:

$$\Upsilon(t+t^{'}) = \begin{bmatrix} \Psi_{1,1}\ \Psi_{1,2}\ ...\ \Psi_{1,U} \\ \Psi_{2,1}\ \Psi_{2,2}\ ...\ \Psi_{2,U} \\ . \\ . \\ \Psi_{B,1}\ \Psi_{B,2}\ ...\ \Psi_{B,U} \end{bmatrix} \tag{15}$$

Now, the candidate cell b can start downloading predicted content that can subsequently be transferred during non-peak hours. As a result, each base station b will have a content caching decision matrix $\mathbf{C}_b$ which stores information of the predicted contents, and it is given as:

$$\mathbf{C} = \begin{bmatrix} C_1 \\ C_2 \\ : \\ C_U \end{bmatrix} = \begin{bmatrix} c_{1,1} & ... & c_{1,F} \\ c_{2,1} & ... & c_{2,F} \\ : & : & : \\ c_{U,1} & ... & c_{U,F} \end{bmatrix} \in \left\{0, \mathbb{Z}^{+}\right\}^{U \times F} \tag{16}$$

where the term $c_{u,f} = 1$ means the content "f" is cached at cell b for the user u. To ensure successful file delivery these $\mathbf{C}_b$ files needed to be transferred to associated users before they are leaving for next cell. Thus, the file transfer time depends upon user u sojourn time t_θ and maximum available downlink rate r_{bu}. Assuming, we have predicted users' demand accurately i.e., of user u, $\lambda_u \subseteq \bar{D}$, at time t which he/she will be demanding at time $t + t^{'}$, user requests for demanding files will be satisfied if following criteria holds which can represented as file satisfaction ratio for user u, and given by:

$$\zeta_u(\lambda) = \frac{1}{N} \sum_{n \in \lambda} \mathbf{P} \left\{ \frac{e_n}{t_{so2} - t_{so1}} \geq r_{b,u} \right\} \tag{17}$$

where e_n is length of nth file and $\mathbf{P}\{...\}$ is indicator function which returns 1 if statement holds otherwise 0. Now, an optimization problem can be formulated in order to maximize the number file (request) satisfaction ratio ζ_u to improve user's QoE, which is given by:

$$\begin{aligned} \max_{\theta_{\mathbf{b},\mathbf{b}',\mathbf{r}_{\mathbf{b},\mathbf{u}}}} \zeta_u(\lambda) = \frac{1}{N} \sum_{n \in \lambda} \mathbf{P} \left\{ \frac{e_n}{t_{so2} - t_{so1}} \geq r_{b,u} \right\} \\ \textbf{subject to:} \quad \mathbf{o} \leq o^{max}, \\ r_{b,u} \leq R^{max}, \end{aligned} \tag{18}$$

where the term o^{max} is the cell content storage capacity, and R^{max} represents maximum file download rate. For each user u even we know the exact sojourn time t_θ and upcoming future requests "λ_u" information, still all the downloaded contents cannot be transferred due to downlink rate r_{bu} and O^{max} cell storage constraints. In such case combinatorial problem, brute-force search solution would be a hard option. To deal with it, we reduced the search space by employing β to consider only those users who's $t_b \leq \beta$. The Algorithm 3 transfer predicted files from cell b to associated users. Thus the Algorithm 3 instructs the cell b to start caching the contents with maximum rating scores and stops if the cell storage capacity has reached maximum limit.

Algorithm 1 gives the implementation steps of the MDPCO scheme. Each BS b initially trains a local mobility prediction model and sends its local updates to fog-model, where global federated averaging is performed. Then the BS b downloads global down updates and predicts user next future cells (line 11-25). The BS b also trains the local future content score prediction model and shares updated to fog-model. After global federated averaging, updated weights are sent to the local models, where they predict future content ratings (line 33-49). Now, using mobility and content prediction models each BS b invokes Algorithm 3 to cache the files with the highest rating scores until the shortage capacity $\mathbf{o}$ is reached.

5. Performance Evaluation and Results

This section covers the performance evaluation of the proposed framework. We assessed the performance of federated mobility and caching models in terms of prediction error and average prediction accuracy. The performance of content offloading is assessed through average caching hits against the caching storage capacity.

5.1. Simulation Setup

We performed network-level simulations to assess the MDPCO framework's performance, assuming that the NFs are BSs deployed by network operators. The proposed framework is evaluated using the available real mobility trace-based dataset from Crawdad Lab's *Context* dataset [42]. The *Context* dataset was collected using a tower logger software operating at mobile devices at Rice community in Houston, Texas, USA, from 10 participants for the duration of 3 to 6 weeks. The data contains information of date, system time (hour, minute, seconds), signal strength and cellular tower identity number i.e., cell-ID. Out of these, we selected cell-IDs from the highest serving cells. In the dataset, the number of data samples are not equal for all the users, and also this dataset contains missing values. The dataset includes the user's cell-IDs and the current date and time. The mobility data is recorded on each minute interval from 10 participants, i.e., the sampling interval is 1 minute. Before feeding the data into the mobility model, we performed necessary pre-processing on the dataset. In this phase, we discarded missing and redundant records from the Context dataset, and the cell-IDs are represented as numeric integers e.g., 522, 512, 724, etc. After necessary data reprocessing, we selected only the top unique cells from all users based on the number of data instances, and we selected only the top 6 unique (distinct) cells. For appropriate class labelling, we replaced cell-IDs with consecutive integer numbers 1, 2, 3, and up-to number of cells in data B. The data has 6 classes and follows a Poisson distribution with a mean of 4.57 and a standard deviation of 2.14. For experiments, 70% of mobility data is used for the training while 30% is used for the model validation.

Algorithm 1: Implementation of Federated ML-based Mobility and Demand-Aware Proactive Content Offloading

Input: X, D, U, B, m_{bu},R_{bu}
Output: $\mathbf{C}_b$
/* cell-IDs matrix with date and time X, number of days D, number of users U, number of cells B, users' demands m_{bu}, available resources R_{bu} */
; // Training of Mobility Prediction Model
for *($d \in X$)* **do**
  $J = X(d, :)'\ logical$ = [true, diff(J) = 0, true]
  $counts$ = diff ($find(logical_d)$)
  $X_{new}(d)$=[$concCID$ $counts$]
Initialization of learning rate η_m, local epochs E_m, and data batch size Γ_m
/* Perform Mobility Model Federated Averaging */
for *($t \in T$)* **do**
  for *($b \in B$)* **do**
    Get $\mathbf{W^b_{t+1}} \leftarrow$ Call **Procedure** *MobilityUpdates*
    Get $\mathbf{W^M_{t+1}} \leftarrow \sum_{i=1}^{B} \frac{1}{B} \mathbf{W^b_{t+1}}$
    Get next cell probability using global model weights $\mathbf{W^M_{t+1}}$
/* Training of Content Score Prediction Model */
Get user feature representation vector M_i using Eq. 9
Get file content feature representation vector N_i using Eq. 10
Concatenate $\mathbf{M}_i$ and $\mathbf{N}_i$ using Eq. 11
Initialization of learning rate η_c, local epochs E_c, and data batch size Γ_c
/* Perform Content Model Federated Averaging */
for *($t \in T$)* **do**
  for *($b \in B$)* **do**
    $\mathbf{W^b_{t+1}} \leftarrow$ Call **Procedure** *ContentPredUpdates*
    $\mathbf{W^C_{t+1}} \leftarrow \sum_{i=1}^{B} \frac{1}{B} \mathbf{W^b_{t+1}}$
    Get future file rating scores from $d'_{i,j}$ global model using global model weights $\mathbf{W^C_{t+1}}$
for *($b \in B$)* **do**
  Invoke Algorithm 3 to offload future files to UEs Estimate *PredictionScore*
terminate

Algorithm 2: Mobility Model Updates

Input: X,U,η_m,E_m,Γ_m
Output: $\mathbf{W_m}$
Procedure MobilityUpdates(X,U,η_m,E_m,Γ_m)
for *($i \in E_m$)* **do**
  for *($j \in \Gamma_m$)* **do**
    $\mathbf{W_m} \leftarrow \mathbf{W'_m} \Delta_j(w; v)$
  return $\mathbf{W_m}$ trained local weights

Algorithm 3: Federated Proactive Content Offloading

Input: $\mathbf{C}_b$,**f**,**e**,**o**
Output: **S**
/* *Caching decision matrix* $\mathbf{C}_b$*, caching files* **f***, file size* **e***, cell cache storage vector* **o** */
Initialize $\mathbf{S} \leftarrow \mathbf{0}_{U \times F}$, $\bar{o} \leftarrow \mathbf{0}_{B \times 1}$
for $b = 1, ..., B$ **do**
  Get **q**
  Get files matrix of UE $\|\mathbf{q}\|$, $\mathbf{G} = \mathbf{C}_b(\mathbf{q}, :)$
  $U = \|\mathbf{q}\|$
  for $u = 1, ..., U$ **do**
    $[\mathbf{v}, \mathbf{i}] \leftarrow SORT(g_u)$, // Sorting files in descending order
    $F = \|\mathbf{i}\|$
    for $f = 1, ..., F$ **do**
      $k \leftarrow i_f$, // Gets index of most popular file
      if $e_k + \bar{o}_u \leq o_u$ **then**
        $S_{u,f} \leftarrow 1$, // Sets elements of user cache matrix to 1
        $\bar{o}_u \leftarrow \bar{o}_b + e_f$, // Increase current storage size
        else
          break // Stops if storage reaches max. capacity

Given the unavailability of a realistic cache dataset, we reside to state-of-the-art studies [43] and [44] that have used sparse real-world data *ML100K* for content prediction in a dynamic environment like F-RANs. The *ML100K* dataset is provided by *MoivesLens* and publicly available at [45]. The dataset contains 100,000 ratings between [1, 5] from 943 users on unique 1682 files. Each user has rated at least 20 files. The dataset is recorded during seven months period, from September 1997 to April 1988. It follows normal distribution where most of the ratings centred around 3 and 4. We used 70% dataset for future content rating predictions while 30% of the data is used to assess model performance accuracy. Furthermore, we set the prediction interval $t' = 1$ minute and all simulations are carried out on a 64-bit desktop machine with main memory of 16 GB. Table 3 provides learning parameters settings for mobility and caching models.

5.2. Performance Metrics

We used two performance metrics (i) mobility prediction model's efficiency, and (ii) content caching model's efficiency. The next cell mobility prediction model's efficiency

Algorithm 4: Content Model Updates

Input: X_c,U,η_c,E_c,Γ_c
Output: $\mathbf{W_c}$
Procedure ContentPredUpdates(X_c,U,η_c,E_c,Γ_c)
for $(i \in E_m)$ **do**
  for $(k \in \Gamma_m)$ **do**
    $\mathbf{W_c} \leftarrow \mathbf{W_c'} + \Delta_k(w; v)$
  return $\mathbf{W_c}$ trained local weights

Table 3
Learning parameters settings

Parameter	Specification	
	Mobility Model	Cache Model
Train-test split %	[70,30]	[70,30]
Learning rate	0.01	0.03
Active Func.	Sigmoid	ReLU
Neurons	[12, 8]]	[100, 50, 20, 10]
Layers	2	4

in terms of accuracy metric is defined as:

$$Acc = \frac{\beta_c}{\beta_c + \beta_i} \tag{19}$$

where β_c and β_i is the number of correct and incorrect next cell predictions respectively. The content caching model's efficiency is expressed as data offloading ratio, Ω [46]:

$$\Omega = \left(\frac{\zeta}{\varphi}\right) \tag{20}$$

where ζ and φ is the number of completely offloaded files and the total number of requests respectively. In order to measure the impact of downlink rate as user quality of experience improvement, we used metric downlink rate distribution τ, a probability that the amount of users received higher downlink rate greater than the predefined threshold ϱ and is defined as:

$$\tau = Pr\left[R_{bu} > \varrho\right] \tag{21}$$

5.3. Results and Discussion

5.3.1. Federated Next Cell Mobility Prediction

Firstly, a federated mobility model is developed via aggregated averaging of locally trained mobility models. Each local mobility model is hosted at each BS. The local model captures users' local mobility data and trained using the method explained in section 4.1. All local models send their gradient updates to the global model at each communication round t. The global model performs federated weights aggregation. After that, global model updates are sent to local models where the next training phase is about to start.

Figure 5 shows the mobility prediction model's error graph for the user's next cell prediction in each communication round. The graph shows both instantaneous loss in blue and the average loss over 50 rounds in red. Federated mobility error slowly decreases in the successive communication rounds highlighted through reduction in the average loss. The global trajectory model learned by the federated algorithm predicts the next cell prediction of the users. As shown in Figure 5, the average next cell prediction accuracy is greater than 79%, impacting the file satisfaction ratio.

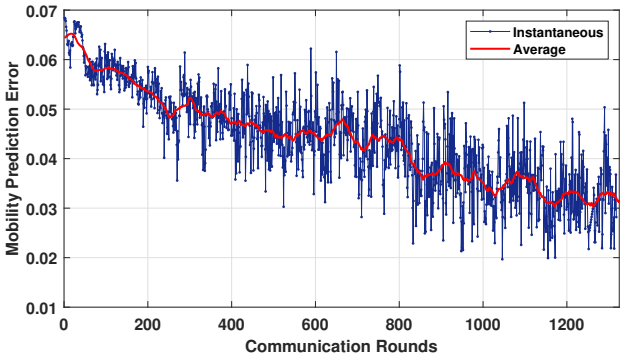

Figure 5: Federated mobility prediction model's error (loss) in each respective communication round

5.3.2. Federated Future Content Prediction

Now, the fog-server develops global future file prediction model via aggregated averaging of locally trained future content prediction models. Each local cache model is hosted serving BS. The local content prediction model is trained using DNN-based model as explained in section 4.2. At each communication round t, each local model send its gradient updates to global model that is hosted nearby fog-server. The global model is then formulated via aggregated averaging of local model updates. And in the next round, the global model updates are sent to each associated local model for the next train phase. Figure 6 shows prediction error plot for user future files prediction in each respective communication round. The federated model converges successfully and predicts users' future next possible content for proactive caching.

5.3.3. Local vs. Federated vs. Cloud Model Performance Analysis

Figure 7 shows the performance comparison of local, federated, and cloud server-based content score prediction models. The cloud-based model outperforms in terms of future file prediction with an average of 94.0% and the least prediction score is observed in of local model i.e., 67.8%. The federated model outperforms as compared to local model and yields an accuracy of 87.3% which is close to the performance of the cloud model. Cloud-based have shown to outperform federated and local models since it has access to the data from all the nodes. However, the FL-based model has shown promising performance despite utilizing less data. On the other hand, the local model lacks overall generalization since it only exploits local data samples to develop the model, and subsequently under-performs during the test-

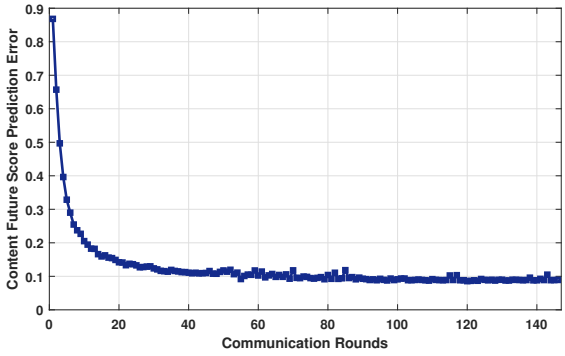

Figure 6: Federated content rating prediction model's error (loss) in each respective communication round

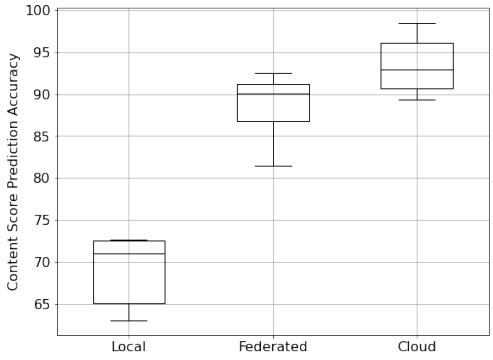

Figure 7: Comparison of prediction accuracy of local, federated and cloud based future content score prediction model

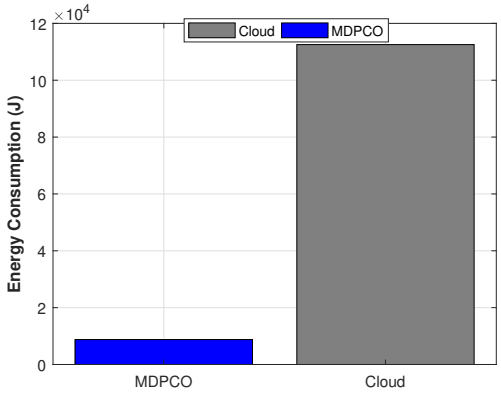

Figure 8: Energy consumption comparison MDPCO-federated and cloud-based model

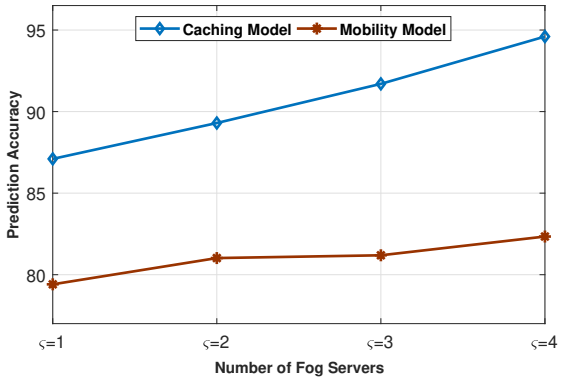

Figure 9: Federated mobility and cache model accuracy under varying the number of fog-servers $\varsigma = [1, 4]$

ing phase. However, FL-based models generalize better and have been shown to perform on par with cloud-based models and provide data privacy as an added advantage. Thus, it can be a viable option for future content score learning in F-RANs with reasonable prediction accuracy and data privacy conservation.

5.3.4. *Energy Consumption in Federated vs. Cloud Schemes*

Figure 8 shows the energy consumption plot for FL-based MDPCO and cloud-based content offloading schemes. The cloud scheme consumes higher energy than the proposed federated approach. The federated model exchanges only weight updates rather than training on data samples; that is why it requires lower energy consumption during federated weight averaging. We carried out these simulations using a system with the specification of Intel core $i7$ processor with the frequency of $3.4GHz$ and 84 Watts. From Figure 8 cloud model consumes 112536.84 Joules of energy where as FL-based MDPCO model consumes 8761.31 Joules of energy. In short, the cloud-based scheme consumes 12.84% more energy as compared to MDPCO scheme.

5.3.5. *Impact of Number of Fog Severs*

In order to study the generalization impact of federated mobility and future content prediction model, we increased the number of fog servers from 1 to 4 and each fog-server is associated with 3 local BSs. Figure 9 shows the performance of federated next cell mobility prediction model and federated future content score prediction model under varying the fog-servers ς= [1, 4]. Under these settings when $\varsigma = 1$ we can see that the mobility model accuracy is 79.5% and it improves up-to 82.34%. Similarly, the future content prediction accuracy increases from 87.3% to 94.3% if we increase the number of fog servers ς from 1 to 4. The content prediction score is dependent on the number of fog servers in the network. This is evident from the results shown in Figure 9, that clearly demonstrates the improvement in prediction accuracies for caching and mobility models, as we increase the number of servers.

5.3.6. *Impact of User Mobility and Data Offloading Ratio*

To study this impact we further extended the experiments for the impact of users' cell sojourn times on the data offloading ratio. We compared the performance of the proposed MDPCO scheme with the state of the art caching schemes such as popular content caching (PopCaching) [47] and ran-

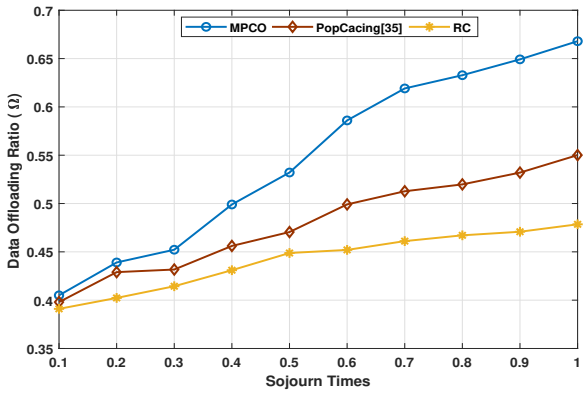

Figure 10: Data offloading ratio with respect to users' cell sojourn times

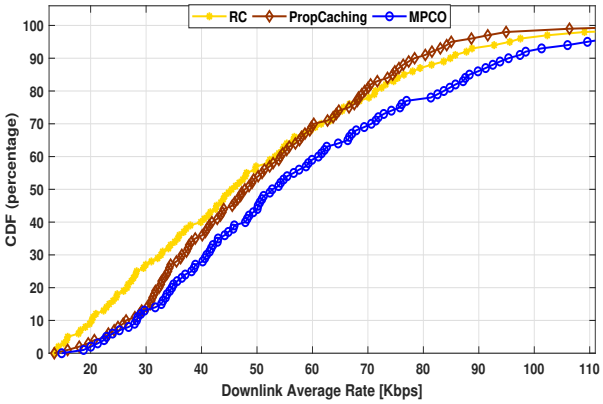

Figure 11: CDF plot of average user downlink rate under MDPCO, RC and PropCaching schemes

dom caching (RC). The following settings are used for RC and PopCaching schemes. In the RC scheme, each file f is cached randomly independent of a user and content statistics whereas in the case of PopCaching the future contents are fetched based on the previous content popularity history only. Figure 10 shows the results containing data offloading ratio Ω with respect to users' respective cell sojourn times under MDPCO, RC and PopCaching schemes. Initially, the proposed MDPCO approach behaves very much similar to PC and PopCaching schemes because of users' lower cell sojourn times. From Figure 10, we can observe that with higher cell sojourn times MDPCO performance improves and it outperforms as compared to RC and PopCaching schemes. The proposed approach exploits users' mobility information to cache the contents at the future locations, whereas RC and PopCaching schemes neglect mobility information. PopCaching only considers the content popularity score and ignores users' content likeliness/dislikeliness statics. In addition to incorporate mobility statics, the proposed MDPCO scheme learns the contextual information between the user and its previously requested contents; therefore, it yields a 9.8% higher data offloading ratio as compared to RC and PopCaching schemes.

5.3.7. Average User Downlink Rate

Now further, we studied user quality of experience (QoE) in terms of average user downlink (DL) rate under MDPCO, RC, and PopCaching schemes. We followed Eq. 2 and 21 to calculate cell load-aware downlink rates of the associated users. We exploited the cumulative distribution function (CDF) of DL rates to compare the performance of MDPCO, RC, and PopCaching schemes. Figure 11 shows CDF of average user DL rates. Current cell load and future content offloading at corresponding cells affect user downlink rates. Under RC and PopCacing average user is getting 471 and 493 KBPs whereas under the proposed MDPCO scheme each user is receiving 557 KBPs. In the case of MDPCO a user is able to receive 1.18 and 1.15 times higher DL rates as compared to RC, and PopCaching respectively. The higher DL rate of MDPCO is because the overall active cell loads are lower as compared to both schemes. As a result, MDPCO enables the network to fetch fewer files from the core server and thus, more channel bandwidth is available to the severing users. These results highlight the effectiveness of the MDPCO framework in terms of fairer user QoE and DL rates.

6. Analysis and Insights

In the proposed MDPCO framework, we trained local mobility and caching models using local data only, which reduced the overall data exchange overheads. FL-empowered global weight aggregations helped MDPCO Framework to capture global mobility and content demand context and provide generalized predictions. In contrast, cloud-based approaches required large date volumes for model training and resulted in higher data exchange overheads. The local learning-based schemes also trained their models based on local data and offered reduction in data communication overheads, however, due to local training these schemes could not capture data generalizations, and resulted in the lower file score prediction accuracy i.e 67.8%. Whereas, the proposed MDPCO framework provided a higher file score prediction accuracy of 87.3% (Figure 7) that reflects the ability to capture overall data context and, therefore, yield better generalization as compared to local training. In addition to data communication overhead reduction and global context learning, the MDPCO scheme avoids users' privacy breaches by restricting users' data at BS level. In the proposed MDPCO scheme, federated models transmit weight updates rather than raw data samples, which is why our scheme is 12.84% more energy-efficient than the cloud-based scheme.

Further, we studied the performance sensitivity of the proposed MDPCO framework in terms of the future score prediction model, data offloading ratio and average user's downlink rate. From simulation results, we observed that the prediction accuracy of federated models is less than the cloud models but significantly higher than the local models. It is also observed that the prediction accuracy of the future content model substantially improves as we increase the

number of fog servers. This allows the model to adopt better generalization over one fog sever. The MDPCO framework demonstrates scalability through extensive simulations by varying the number of fog servers. The results reveal that model accuracy improves when we increase the number of fog servers from 1 to 4. However, in case of an outage or other fog server discovery faults, one or more fog servers becomes unavailable for federated weight averaging; it leads to reduction in the global model accuracy. In future, the MDPCO framework can be extended to make it more resilient to cell/fog server outages.

The proposed MDPCO framework delivered higher downlink rates as compared to RC and Popcachig to the fact that overall active user load is reduced because MDPCO enables the network to fetch fewer files from the core server and thus, more channel bandwidth is available for severing users. The proposed MDPCO system, in its present configuration, is suitable for proactive content caching and delivery of files/jobs that are delay-tolerant and totally downloadable, such as software updates, movies, and music files. In fact, these apps demand a significant amount of bandwidth while being sent to end users. The lightly loaded cells may download a big quantity of such data and send it to users prior to their movement to the congested cells and as result users' QoE improve under the overall network. Further, the MDPCO framework avoids user privacy violations by restricting users' data at the BS level and also possesses the benefits of centralized training through federated weight averaging. These promising features of the proposed MDPCO framework advocate its applicability in the emerging F-RAN-based cellular networks. In short, the building blocks of the MDPCO scheme, i.e. DNN-based mobility and LSTM-based content caching models, are kept simplistic in deep-learning architecture to support minimum computation load and higher energy efficiency.

7. Conclusion

This paper proposed a novel FL-based MDPCO framework for F-RANs that proactively offload users' future contents at local caches based on the next cell mobility statistics. The proposed MDPCO framework contains global mobility and content prediction models developed from the locally trained mobility and caching models. We carried out extensive simulations, and the accuracy of future content score prediction under local, federated, and cloud models is 67.8%, 87.3%, and 94.0% respectively. We observed that the federated model's accuracy increases from 87.3% to 94.3% as we increase the number of fog-servers, highlighting the better generalization ability of federated models. Moreover, the federated mobility and future content score prediction model shows an average next cell predicting accuracy of 84.0%, file score prediction accuracy of 87.3% and consumes 12.84% lesser energy than the cloud-based models. We observed that the MDPCO yields 9.8% higher data offloading ratio and offers 1.18 times higher DL rates as compared to RC and PopCaching approaches.

We demonstrated that the proposed MDPCO framework for F-RANs can proactively predict and offload future contents at cells with higher accuracy under dynamic network conditions by utilising users' mobility and content demand information. The proposed MDPCO scheme can be integrated into current 5G networks to learn and predict mobility and cache demands. In the future, we aim to study the feasibility of the MDPCO framework for D2D supported F-RANs and also enhance performance in terms of higher energy efficiency.

References

[1] W. P. Cisco, "Cisco, visual networking index: Global mobile data traffic forecast update, 2013–2018 white paper," *Document ID*, vol. VI, USA, Mar, 2017.

[2] W. Paper, "Cisco VNI forecast and methodology, 2015-2020," *Document ID*, vol. VI, USA, Jun, 2017.

[3] T. Ahsan, Z. Iqbal, M. Ahmed, R. Alroobaea, A. M. Baqasah, I. Ali, M. A. Raza *et al.*, "Iot devices, user authentication, and data management in a secure, validated manner through the blockchain system," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[4] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Network*, vol. 30, no. 4, pp. 46–53, 2016.

[5] F. R. Costa, R. da Rosa Righi, C. A. da Costa, and C. B. Both, "Nuoxus: A proactive caching model to manage multimedia content distribution on fog radio access networks," *Future Generation Computer Systems*, vol. 93, pp. 143–155, 2019.

[6] R. Wang, M. Li, L. Peng, Y. Hu, M. M. Hassan, and A. Alelaiwi, "Cognitive multi-agent empowering mobile edge computing for resource caching and collaboration," *Future generation computer systems*, vol. 102, pp. 66–74, 2020.

[7] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1710–1732, 2018.

[8] Y. Liu, Z. Ma, Z. Yan, Z. Wang, X. Liu, and J. Ma, "Privacy-preserving federated k-means for proactive caching in next generation cellular networks," *Information Sciences*, vol. 521, pp. 14–31, 2020.

[9] S. Manzoor, S. Mazhar, A. Asghar, A. Noor Mian, A. Imran, and J. Crowcroft, "Leveraging mobility and content caching for proactive load balancing in heterogeneous cellular networks," *Transactions on Emerging Telecommunications Technologies*, p. e3739, 2019.

[10] Y. Shi and Q. Ling, "An adaptive popularity tracking algorithm for dynamic content caching for radio access networks," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 5690–5694.

[11] G. S. Rahman, M. Peng, S. Yan, and T. Dang, "Learning based joint cache and power allocation in fog radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4401–4411, 2020.

[12] L. Hu, Y. Miao, J. Yang, A. Ghoneim, M. S. Hossain, and M. Alrashoud, "If-rans: Intelligent traffic prediction and cognitive caching toward fog-computing-based radio access networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 29–35, 2020.

[13] H. Feng, Y. Jiang, D. Niyato, F.-C. Zheng, and X. You, "Content popularity prediction via deep learning in cache-enabled fog radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[14] F. Jiang, Z. Yuan, C. Sun, and J. Wang, "Deep q-learning-based content caching with update strategy for fog radio access networks," *IEEE Access*, vol. 7, pp. 97 505–97 514, 2019.

[15] Y. Jiang, X. Cui, M. Bennis, F.-C. Zheng, B. Fan, and X. You, "Cooperative caching in fog radio access networks: a graph-based approach," *IET Communications*, vol. 13, no. 20, pp. 3519–3528, 2019.

[16] Y. Sun, M. Peng, and S. Mao, "A game-theoretic approach to cache and radio resource management in fog radio access networks," *IEEE*

Transactions on Vehicular Technology, vol. 68, no. 10, pp. 10 145–10 159, 2019.
[17] S. Yan, M. Jiao, Y. Zhou, M. Peng, and M. Daneshmand, "Machine-learning approach for user association and content placement in fog radio access networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9413–9425, 2020.
[18] Z. Zhao, C. Feng, H. H. Yang, and X. Luo, "Federated-learning-enabled intelligent fog radio access networks: Fundamental theory, key techniques, and future trends," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 22–28, 2020.
[19] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.
[20] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.
[21] G. S. Rahman, M. Peng, K. Zhang, and S. Chen, "Radio resource allocation for achieving ultra-low latency in fog radio access networks," *IEEE Access*, vol. 6, pp. 17 442–17 454, 2018.
[22] Y. N. Shnaiwer, S. Sorour, T. Y. Al-Naffouri, and S. N. Al-Ghadhban, "Opportunistic network coding-assisted cloud offloading in heterogeneous fog radio access networks," *IEEE Access*, vol. 7, pp. 56 147–56 162, 2019.
[23] Y. Jiang, A. Peng, C. Wan, Y. Cui, X. You, F.-C. Zheng, and S. Jin, "Analysis and optimization of cache-enabled fog radio access networks: Successful transmission probability, fractional offloaded traffic and delay," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5219–5231, 2020.
[24] X. Wang, C. Wang, X. Li, V. C. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, 2020.
[25] L. Cui, X. Su, Z. Ming, Z. Chen, S. Yang, Y. Zhou, and W. Xiao, "Creat: Blockchain-assisted compression algorithm of federated learning for content caching in edge computing," *IEEE Internet of Things Journal*, 2020.
[26] T. Xiao, T. Cui, S. Islam, and Q. Chen, "Joint content placement and storage allocation based on federated learning in f-rans," *Sensors*, vol. 21, no. 1, p. 215, 2021.
[27] Z. Xue, P. Zhou, Z. Xu, X. Wang, Y. Xie, X. Ding, and S. Wen, "A resource-constrained and privacy-preserving edge computing enabled clinical decision system: A federated reinforcement learning approach," *IEEE Internet of Things Journal*, 2021.
[28] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
[29] R. Cheng, Y. Sun, Y. Liu, L. Xia, D. Feng, and M. Imran, "Blockchain-empowered federated learning approach for an intelligent and reliable d2d caching scheme," *IEEE Internet of Things Journal*, 2021.
[30] C. Li, J. Liu, Q. Zhang, and Y. Luo, "Efficient cooperative cache management for latency-aware data intelligent processing in edge environment," *Future Generation Computer Systems*, vol. 123, pp. 48–67, 2021.
[31] L. Xiao, X. Lu, T. Xu, X. Wan, W. Ji, and Y. Zhang, "Reinforcement learning-based mobile offloading for edge computing against jamming and interference," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6114–6126, 2020.
[32] Y. Qi, M. S. Hossain, J. Nie, and X. Li, "Privacy-preserving blockchain-based federated learning for traffic flow prediction," *Future Generation Computer Systems*, vol. 117, pp. 328–337, 2021.
[33] 3rd Generation Partnership Project, " 3GPP, "Architecture enhancements for 5G System to support network data analytics services," TS 23.288, 2019, version 16.0.0. [Online]. Available: http://www.3gpp.org/DynaReport/23288.htm," March 2019.
[34] D. Kakadia, J. Yang, and A. Gilgur, "Evolved universal terrestrial radio access network (eutran)," in *Network Performance and Fault Analytics for LTE Wireless Service Providers*. Springer, 2017, pp. 61–81.
[35] Y. Zeng, J. Xie, H. Jiang, G. Huang, S. Yi, N. Xiong, and J. Li, "Smart caching based on user behavior for mobile edge computing," *Information Sciences*, vol. 503, pp. 444–468, 2019.
[36] T. Zhou, Y. Huang, L. Fan, and L. Yang, "Load-aware user association with quality of service support in heterogeneous cellular networks," *IET Communications*, vol. 9, no. 4, pp. 494–500, 2015.
[37] J. Tadrous, A. Eryilmaz, and H. El Gamal, "Proactive content download and user demand shaping for data networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 6, pp. 1917–1930, 2014.
[38] I. FG-ML5G, "Focus group on machine learning for future networks including 5g."
[39] Y.-J. Ku, D.-Y. Lin, C.-F. Lee, P.-J. Hsieh, H.-Y. Wei, C.-T. Chou, and A.-C. Pang, "5g radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 46–52, 2017.
[40] D. Kingma and J. Ba., "Adam: A method for stochastic optimization." *arXiv preprint*, vol. 1412.6980, pp. –., 2014.
[41] Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated learning based proactive content caching in edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.
[42] L. Z. A. Rahmati. Crawdad data set rice/context (v. 2007-05-23). [Online]. Available: http://crawdad.cs.dartmouth.edu/rice/context
[43] Y. Zhang, Y. Li, R. Wang, J. Lu, X. Ma, and M. Qiu, "Psac: Proactive sequence-aware content caching via deep learning at the network edge," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2145–2154, 2020.
[44] S. Khanal, K. Thar, and E.-N. Huh, "Dcol: Distributed collaborative learning for proactive content caching at edge networks," *IEEE Access*, vol. 9, pp. 73 495–73 505, 2021.
[45] http://grouplens.org/datasets/movielens/, "Movielens Dataset," March 2018.
[46] C. Ying, X. Wang, and Y. Luo, "Optimization on data offloading ratio of designed caching in heterogeneous mobile wireless networks," *Information Sciences*, vol. 545, pp. 663–687, 2021.
[47] S. Li, J. Xu, M. Van Der Schaar, and W. Li, "Popularity-driven content caching," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

SANAULLAH MANZOOR is Ph.D. fellow in the Department of Computer Science at Information Technology University, Lahore, Pakistan. He received MS degree in computer system engineering from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Swabi, Pakistan. His main research areas include applied machine learning for cellular networks mainly focuses on user mobility modelling, proactive resource management, network performance analysis, load balancing and content caching polices.

ADNAN NOOR MIAN is professor at Information Technology University, Lahore, Pakistan and a senior associate in the International Centre for Theoretical Physics (ICTP), Trieste, Italy in the field of Internet of Things (IoT). He received his Ph.D. in Computer Engineering from Sapienza University of Rome, Italy in 2009 and postdoc from the

same university. In 2018-2019 he has been a visiting scholar in the Department of Computer Science and Technology, the Computer Laboratory, University of Cambridge, UK. He has published more than 40 papers in refereed conferences and journals and serving in a number of technical program committees of international conferences and reviewer of many international journals. His research interests include wireless sensor and ad hoc networks, distributed algorithms, and mobile and distributed systems.

AHMED ZOHA received the Ph.D. degree in electrical and electronic engineering from the 5G Innovation Centre, University of Surrey, U.K., and the M.Sc. degree in communication engineering from the Chalmers University of Technology, Sweden. He has more than 12 years of experience in the domain of artificial intelligence, big data-enabled self-organizing networks for wireless communication, healthcare technologies, and smart energy monitoring. He is currently a Lecturer with the School of Engineering, University of Glasgow, U.K. His research work is centered around a broad range of machine learning applications spanning 5G network optimization, human behavior modeling for clinical interventions, non-intrusive load monitoring, and he strongly advocates the use of AI for social good. His research has been cited by national and international bodies, regulators, and the media. He has also received the two IEEE best paper awards.

MUHAMMAD ALI IMRAN received the M.Sc. (Hons.) and Ph.D. degrees from Imperial College London, London, U.K., in 2002 and 2007, respectively. He is currently the Vice Dean of the Glasgow College, UESTC, and a Professor of communication systems with the School of Engineering, University of Glasgow. He is also an Affiliate Professor with the University of Oklahoma, Norman, OK, USA, and a Visiting Professor with the 5G Innovation Centre, University of Surrey, Guildford, U.K. He has more than 20 years of combined academic and industry experience, working primarily in the research areas of cellular communication systems. He has authored or coauthored more than 400 journals and conference publications and has been a Principal or Co-Principal Investigator on more than 10 million in sponsored research grants and contracts. He has supervised more than 40 successful Ph.D. graduates. He holds 15 patents. He is a Fellow of IET and a Senior Fellow of the Higher Education Academy (SFHEA), U.K. He was a recipient of the Award of Excellence in recognition of his academic achievements, conferred by the President of Pakistan, the IEEE ComSocs Fred Ellersick Award, in 2014, the FEPS Learning and Teaching Award, in 2014, and the Sentinel of Science Award, in 2016. He was twice nominated for the Tony Jeans Inspirational Teaching Award. He was a shortlisted finalist for The Wharton-QS Stars Awards, in 2014, the QS Stars Reimagine Education Award, in 2016, for innovative teaching, and the VCs Learning and Teaching Award at the University of Surrey.