A Novel Group Management Scheme of Clustered Federated Learning for Mobile Traffic Prediction in Mobile Edge Computing Systems

Faranaksadat Solat, Tae Yeon Kim, and Joohyung Lee

***Abstract*—This study developed a novel group management scheme based on clustered federated learning (FL) for mobile traffic prediction (referred to as FedGM) in mobile edge computing (MEC) systems. In FedGM, to improve the convergence time during the FL procedure, we considered multiple MEC servers to first be clustered based on their geographic locations and augmented data patterns as references for clustering. In each cluster, by alleviating the straggler impact owing to the heterogeneity of MEC servers, we then designed a group management scheme that optimizes i) the number of groups to be created and ii) the group association of the MEC servers by minimizing their average idle time and group creation cost. For this purpose, we rigorously formulated analytical models for the computation time for local training and estimated the average idle time by applying different frequencies of local training over the MEC servers. The optimization problem was designed using a nonconvex problem, and thus a genetic-based heuristic approach was devised for determining a suboptimal solution. By reducing the average idle time, thereby increasing the workload of the MEC servers, the experimental results for two real-world mobile traffic datasets show that FedGM surpasses previous state-of-the-art methods in terms of convergence speed with an acceptable accuracy loss.**

***Index Terms*—5G/6G, federated learning (FL), genetic algorithm, group management scheme, mobile edge computing (MEC) server, mobile traffic prediction.**

I. INTRODUCTION

WITH the rapid development of artificial intelligence (AI) technologies, AI-driven mobile networks have received increased attention from various industries and academia toward sixth generation (6G) communications for enabling more advanced intelligence into the network management domain. Specifically, by utilizing the data generated from the network functions (NFs), the network data analytic function (NWDAF) was introduced and standardized by 3GPP for leveraging the AI functionality in core networks [1]. Driven by this trend, mobile traffic prediction, which estimates the future volume of traffic data , has emerged as one of the key study items for enabling proactive AI-driven network management by meeting the requirements of heterogeneous services from various devices (e.g., smartphones, sensors, vehicles, drones, and factory machines) on 6G networks [2]–[4].

In this context, most studies on mobile traffic prediction schemes have relied on the centralized learning paradigm through the application of different AI models, which require a significant number of raw data transmitted from each mobile edge computing (MEC) server to a centralized training server [5], [6]. Accordingly, such a centralized approach results in negative impacts on the latency, processing costs at the centralized server, and privacy. Hence, distributed training approaches for mobile traffic prediction are required to address this challenge.

One potential technique for addressing this challenge is FL [7]. In FL, multiple collaborative clients, such as mobile devices, MEC servers equipped with base stations (BSs), or companies, train their local models while retaining their local data. Subsequently, only the model parameters (or gradients) are transferred to a centralized training server for global model aggregation. The updated global model parameters are then delivered back to the clients for the next local training, and the process is repeated until the model converges. In this case, the training speed is limited by the heterogeneity of the clients (e.g., differing numbers of datasets and computing capabilities) who participated in the FL. Thus, many studies have been conducted to improve the convergence speeds of FL considering the clustering of clients, the client selection, and resource management (e.g., bandwidth and CPU/GPU frequency) [8]. Specifically, in [9], a resource-based aggregation frequency control method was proposed that considers different frequencies of local training over the clients for a better convergence speed. In [10], a joint client selection and bandwidth allocation algorithm was developed considering the heterogeneity of the clients for reducing the convergence time. Nevertheless, to the best of our knowledge, most FL studies have yet to be applied to mobile traffic prediction scenarios. Zhang $et\ al.$ [11] introduced clustering-based FL for mobile traffic prediction based on the augmented data pattern and geographic locations of the MEC servers as clustering references. Specifically, because a variety of BSs may have distant traffic patterns that hinder the convergence speeds, MEC servers are clustered, and a model based on the dual attention aggregation mechanism and an aggregation structure based on a hierarchical structure

This work was supported in part by the ICT R&D program of MSICT/IITP. [2022-0-00862, Development of Intelligent 6G Mobile Core Network Technologies] and in part by the Gachon University research fund of 2022 (GCU-202206120001).

Manuscript received November 21, 2022 revised April 14, 2023; approved for publication by Chien-Ming Chen, Division 2 Editor, May 25, 2023.

F. Solat and J. Lee are with the Department of Computing, Gachon Univ., Seongnam 13120, Rep. of Korea. email: {faranak1995, j17.lee}@gachon.ac.kr.

T. Y. Kim is with the Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Rep. of Korea. email: tykim@etri.re.kr.

J. Lee is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2023.000025

1229-2370/23/$10.00 © 2023 KICS

have approved. However, during the clustering, there is no consideration of the heterogeneous computing power or the number of datasets in the MEC servers, thereby affecting the local training speed. Thus, previous approaches still have room for improvement, thus inspiring the present study.

In this study, we jointly optimize the grouping of the MEC servers and group association based on clustered FL for mobile traffic prediction in MEC system (FedGM) aiming for improving the convergence time during the FL procedure. The main contributions of this study are summarized as follows:

- We developed a new clustered FL group management scheme for mobile traffic prediction (referred to as FedGM) in MEC systems to reduce the latency, processing costs, and privacy. All these things will greatly affect the amount of processing time, energy consumption, and convergence time.
- To improve the convergence time during the FL procedure, in FedGM, we considered multiple MEC servers to be first grouped based on the augmented data pattern and the geographic locations of the MEC servers as cluster references to simultaneously capture different clients traffic patterns and keep data privacy.
- Subsequently, in each cluster, reducing the impact of stragglers owing to the heterogeneity of the MEC servers, we designed a group management scheme that optimizes: i) the number of groups to be created and ii) the group association of the MEC servers by minimizing both their average idle time and the cost of creating groups of servers.
- To this end, we rigorously formulated analytical models for the local training computation time and estimated the average idle time by applying various numbers of local updates over the MEC servers.
- The optimization problem is designed from a non-convex problem, and thus, a heuristic approach based on a genetic algorithm was devised to determine a suboptimal solution.
- For practical reasons, the proposed scheme complies with the NWDAF-distributed framework standardized by the 3GPP specifications. Then, rather than waiting for a straggler, by reducing the average idle time, the MEC servers maximize the time spent on useful local training, yielding a faster convergence.
- Extensive experiments are conducted using real-world mobile traffic datasets in two cities to show the effectiveness of our designed FedGM framework. The results show that FedGM outperforms previous state-of-the-art methods in terms of convergence rate with an acceptable loss in accuracy.

The remainder of this paper is organized as follows. Section II summarizes the previous works related to mobile traffic prediction. Section III provides the proposed system model of the FedGM. Section IV provides our proposed FedGM for minimizing the average idle time and group creation cost, and detailed discussions on its implementation. In Section V evaluates the performance of the proposed system. Finally, Section VI concludes the paper.

II. Related Works

In order to manage resources efficiently in a communication network, it is very important to use traffic time series. With the expansion of the use of cellular mobile technologies in the edge space, more attention has been paid to mobile traffic prediction in recent articles. In general, mobile traffic prediction schemes can be analyzed in three groups includes simple methods, parametric methods, and non-parametric methods. Simple methods are used to predict future traffic value based on the historical averaging [12]. The most important application of the simple methods are in the exponentially weighted moving average (EWMA) which is the temporal dynamic sensitivity of the traffic volumes can be controlled by calibrating the exponential smoothing factor in the range of zero and one [13].

Although these methods are highly accurate. However, for these day's complex communication networks, where network dynamics are influenced by various spatial and temporal factors. They are not of much use. For this reason, it is better to use statistical or probabilistic prediction methods in which the mixture of various spatio-temporal factors have a greater impact on traffic dynamics. Parametric methods create probabilistic or statistical prediction models assuming that traffic dynamics are generated from random variables with a specific probability distribution [14]. With the rapid development of mobile communications and the widespread penetration of cellular traffic, many efforts have been constructed to predict mobile traffic. Initially, classical learning methods such as the auto-regressive integrated moving average ARIMA [12] and support vector regression (SVR) [26] are popular for traffic prediction. To start the prediction process, it is necessary to first create a model through calculations or simulation. In [12] they used ARIMA which is able to detect the hidden burstiness and self-similarity hidden in traffic series [15], [16]. In addition to ARIMA, a wide range of probabilistic and statistical applications such as Markov model [17], entropy theory [18], covariance function [19] and alpha-stable model [20] have been investigated for mobile traffic prediction. In the last few years, deep learning approaches have established themselves as strong competitors to traditional statistical models and have become mainstream technologies for solving the mobile traffic prediction problem. According to [21] the authors used the feature-selection method that a prioritization method that prioritizes traffic log data as stated by their contributions to prediction. To solve the problem of a huge capacity of traffic log data, only a part of traffic logs can be used for real-time mobile traffic prediction. Non-parametric methods create prediction models without considering probability distributions. With the emergence of deep learning technology based on artificial intelligence, these methods have been created. Basically, since traffic prediction requires obtaining the temporal dependence that appears in the time-series data set; long short-term memory (LSTM) is usually used due to proper recognition of sequential patterns [11], [22], [23]. Also, [24], proposed the use of data pre-processing before injecting them into an LSTM neural network for time series prediction. Even though, FL was presented to allow model training in distributed manner when data usage is limited to the local domain without data

TABLE I
APPROACHES TO MOBILE TRAFFIC PREDICTION.

Approaches	Ref.	Key Ideas
Simple methods	[12] [13]	Simple methods are used to predict future traffic value based on the historical averaging. EWMA is the temporal dynamic sensitivity of the traffic volumes can be controlled by calibrating the exponential smoothing factor in the range of zero and one.
Parametric methods	[14] [12] [15], [16] [17] [18] [19] [20]	Create probabilistic or statistical prediction models assuming that traffic dynamics are generated from random variables with a specific probability distribution. Use ARIMA to create a model through calculations or simulation. Able to detect the hidden burstiness and self-similarity hidden in traffic series. Markov model Entropy theory Covariance function Alpha-stable model
Non-parametric methods	[11]	Create prediction models without considering probability distributions (with the emergence of deep learning technology).
The feature-selection method	[21]	Use a prioritization method that prioritizes traffic log data as stated by their contributions to prediction. To solve the problem of a huge capacity of traffic log data, only a part of traffic logs can be used for real-time mobile traffic prediction.
Dual attention-based FL (FedDA)	[11]	Try to solve the mobile traffic prediction problem by a distributed architecture and FL.
LSTM method in mobile traffic	[11], [22], [23] [24]	Use the proper recognition of sequential patterns. Use data pre-processing before injecting them into an LSTM neural network for time series prediction.
Federated meta-learning algorithm	[25]	To achieve efficient mobile traffic prediction at the edge they introduced a model-agnostic meta-learning (MAML) algorithm based on the FL framework.

leakage concern. In mobile traffic prediction, FL can be used to avoid increase in transmission of traffic data overhead while ensuring data privacy for various users or applications. In this regard, a dual-attention-based FL (FedDA) scheme was proposed in [11], which effectively collects the contributions of different client models in different BSs to a global model in the central server. This manner was used [27] for the first time in 2016 for collaborative learning in wireless networks, where communication resources, such as transmission power and bandwidth, are restricted and local clients privacy needs to be protected. In [25], authors introduced a model-agnostic meta-learning (MAML) algorithm based on the FL framework to achieve efficient mobile traffic prediction at the edge. To this aim, they trained a sensitive initial model that could adapt very quickly to heterogeneous mobile traffic datasets in various regions. In addition, they used distance-based weighted model aggregation on their proposed scheme and then, compared the results with some other traditional and FL-based algorithms like SVR, random forest (RF), federated averaging (FedAvg), and FedDA, respectively.

III. SYSTEM MODEL

In this section, we first introduce the background of FL and its application to the distributed NWDAF framework in MEC systems for mobile traffic prediction. Then, we present the proposed FedGM, a new clustered FL group management scheme for mobile traffic prediction in MEC systems, and its analytical models.

A. Background of FL and its Application to the Distributed NWDAF Framework

In this subsection, we firstly provide the background of FL and its application to the distributed NWDAF framework for mobile traffic prediction in MEC systems. As shown in Fig. 1(a), we considered multiple MEC servers participating in the FL process, with each MEC server directly connected to the BSs through a wired network such as optical fiber. We define $\mathcal{I}$ as the set of MEC servers, where $|\mathcal{I}| = I$ denotes the total number of MEC servers. For each $i \in \mathcal{I}$, MEC server i has its own local mobile traffic data, denoted as $\mathcal{D}_i$=$\{\mathcal{D}_i^1, \mathcal{D}_i^2, \cdots, \mathcal{D}_i^Z\}$ with Z time intervals, where D_i denotes the size of the mobile traffic data $\mathcal{D}_i$. To realize the FL process on the MEC server, we adopted a distributed NWDAF architecture as in [28]. Note that the NWDAF is a 3GPP-defined function in 5G networks that provides data analytics services and it uses network data to provide insights into network traffic, user behavior, and network performance. FL can be used in conjunction with NWDAF to train models using the data generated by the network and provide insights to the network operators. Specifically, we assume that the root NWDAF is installed at the core network server for global model training and that each leaf NWDAF is installed at the MEC server in a container form for local training. By following the normal procedures of the FL, MEC server i trains the local model at the model training logical function (MTLF) module by utilizing the D_i training dataset stored in the data storage to minimize the loss function $L_i(w_i)$ with training model parameters w_i, which is defined as

$$L_i(w_i(r)) = \tfrac{1}{D_i} \sum_{k \in \mathcal{D}_i} l_k(w_i(r)). \tag{1}$$

Here, r is the round index, and $l_k(w_i(r))$ is a loss function that quantifies the error between the output and ground-truth label. At the tth local iteration, each MEC server i updates its model parameters w_i as follows:

$$w_i(r)^t = w_i(r)^{t-1} - \eta \Delta L_i(w_i(r)^{t-1}), \tag{2}$$

where η denotes the learning rate. After the training is complete, each MEC server i uploads the weights $w_i(r)$ to the aggregation module of the learning module in the root NWDAF using the communication module. It should be

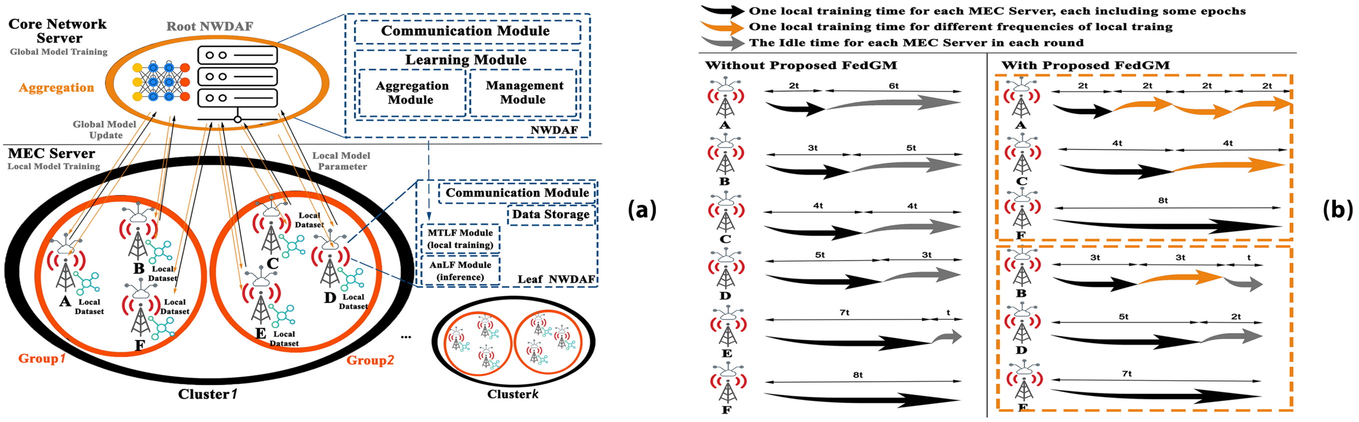

Fig. 1. (a) Proposed FedGM scheme. (b) A round of model training based on grouping.

denote that, throughout this paper, according to [29], due to the high transmit power at the NWDAF core network server and the bandwidth that can be used for data broadcasting between MEC servers and the NWDAF core network server, the communication time is suppose to zero. Then, the core network server obtains the global model w at the aggregation module by aggregating the local weights[1]. The core network server transmits the global model parameters w to all MEC servers. By running these steps iteratively until the global model w converges, the global model w can be utilized in an analytics logical function (AnLF) module for mobile traffic prediction (i.e., inference).

B. Proposed Clustered FL Framework

In this subsection, we provide the proposed FedGM as a new clustered FL group management scheme for mobile traffic prediction, which is a novel and optimized approach aiming for reduced computation and convergence time while ensuring privacy and accuracy. To achive this goal, based on the framework suggested in the previous subsection, we provide the novel function in the management module at the root NWDAF. Specifically, in this management module, to improve the convergence speeds, we first clustered the multiple MEC servers based on the augmented data pattern and geographic locations of the servers as clustering references, as in [11][2]. Then, as in [9], we assume that for each $i \in \mathcal{I}$, MEC server i has a different number of datasets D_i and different CPU frequency e_i. Owing to the heterogeneity of the MEC servers, as shown in Fig. 1(b), the local training time for each MEC server can vary, and the idle time over the MEC servers caused by stragglers reduce the workload of the MEC servers. This indirectly hinders a fast convergence. Thus, in the proposed FedGM, by introducing different frequencies of local training over the MEC servers and a group management scheme in the management model[3], we minimize the weighted sum of the average idle time and group creation cost over the MEC servers by determining the optimal number of groups to be created and the group association of the servers. Then, rather than waiting for the straggler, the MEC servers maximize the time spent on useful local training, which yields a faster convergence. The right side of Fig. 1(b) depicts the impact on the idle time over the MEC servers with the proposed FedGM. Here, the average idle time over the MEC servers, depicted by the gray arrow, is dramatically reduced compared with that without the proposed FedGM.

C. Analytical Models

In this subsection, we provide analytical models by formulating i) the computational load and time for local training, and ii) the idle time for each MEC server i.

1) Computational load and time for local training: To model the computational time for local training, we first defined the computational load W_i (i.e., the number of CPU cycles for local training at the server) for MEC server i, which is proportional to D_i. Then, W_i is obtained by

$$W_i = N_{e_i}(\frac{D_i}{B_s})CB_s = N_{e_i}CD_i, \tag{3}$$

where N_{e_i} denotes the total number of epochs for each MEC server i in the cluster. In addition, B_s denotes the mini-batch size (training data size of one iteration), and the constant C is the number of CPU cycles required for training 1-bit of data. The computation time t_i of MEC server i with CPU frequency e_i is then given by the following:

$$t_i = \frac{W_i}{e_i}. \tag{4}$$

2) Idle time: In the proposed FedGM, the management module can adjust the number of groups m to be created in the cluster and the group association rule $b_{i,j}$, where $b_{i,j}$ is a binary variable that represents whether the MEC server i is associated with group j in the specified cluster. For instance, if it is associated, $b_{i,j} = 1$; otherwise, $b_{i,j} = 0$. With the obtained

[1] Various techniques such as FedAvg and FedDA can be utilized, as in [11].

[2] As in [11], for privacy concerns, instead of sending raw mobile traffic data, each MEC sends its augmented data and geographic location information. We compute the statistical average value for each time point to obtain the augmented data, which follows the steps described in Section IV. [11]. Then, the size of the augmented data is smaller than that of the raw data, while preserving high similarities.

[3] In the group level aggregation, we consider a weak synchronization, which is widely used for a hierarchical FL model, as in [9].

TABLE II
VARIABLES AND FUNCTIONS TABLE.

Name	Instruction
$\mathcal{I}$	The set of MEC servers
$\lvert\mathcal{I}\rvert = I$	The total number of MEC servers
$\mathcal{D}_i$	The size of the mobile traffic data
w_i	The training model parameters
$L_i(w_i)$	The loss function
r	The round index
$l_k(w_i(r))$	A loss function between the output and ground-truth label
η	The learning rate
e_i	The CPU frequency
N_{e_i}	The total number of epochs for each MEC server i
B_s	The mini-batch size
C	A constant number of CPU cycles for training 1-bit of data
t_i	The computation time t_i of MEC server i
$b_{i,j}$	A binary variable represents MEC server i is associated with group j
$t_{i,j}$	The computation time for MEC server i in group j
max_j	The maximum computation time obtained by the straggling MEC server in group j
$t_{idle_{i,j}}$	The idle time for MEC server i in group j
α	The weight factor of m compared to the average idle time over the MEC servers
m	The total number of groups to be created between 1 and M_{max}
M_{max}	The maximum number of groups
CH	A chromosome in the genetic algorithm
POP	A population set is called a generation
r_{select}	The selection ratio
Ch_{pi}	A parent chromosome

association rule $b_{i,j}$, the computation time for MEC server i in group j denoted by $t_{i,j}$ can be extended from (4), which is given by $t_{i,j} = t_i b_{i,j}$. Then, the maximum computation time obtained by the straggling MEC server in group j max_j is calculated as

$$max_j = \max(t_{1,j}, t_{2,j}, \cdots, t_{I,j}). \tag{5}$$

By applying different frequencies of local training over the MEC servers, the frequency of local training at MEC server i in group j can be obtained by $F_{i,j} = \lfloor max_j/t_i \rfloor$. Finally, the idle time for MEC server i in group j is given by the following:

$$t_{idle_{i,j}} = max_j - F_{i,j}t_i = max_j - \lfloor \frac{max_j}{t_i} \rfloor t_i. \tag{6}$$

IV. PROBLEM FORMULATION AND PROPOSED SCHEME

In this section, we introduce the proposed FedGM scheme, which jointly optimizes the grouping of the MEC servers and group association based on clustered FL for mobile traffic prediction in MEC system aiming for improving the convergence time during the FL procedure. We provide in Table II a list of the major symbols that we define and use in this paper.

A. Problem Definition

In our design of the FedGM scheme, we aimed at minimizing both the average idle time and the group creation cost over the MEC servers in the cluster.

It should be noted that in this study, the number of clusters is assumed to be fixed while the number of groups within each cluster is optimized using a genetic algorithm. Thus, a multi-objective optimization problem is involved in balancing different types of metrics. To characterize this trade-off, we adopt the weighted linear sum method [10] to define the cost function as

$$C(b_{i,j}, m) = \sum_j \sum_i \frac{t_{idle_{i,j}}}{I} + \alpha \cdot m, \tag{7}$$

where $\alpha \geqslant 0$ is the weight factor of m compared to the average idle time over the MEC servers, which represents the group creation cost (e.g., memory, processing resources, and increased complexity). Here, α is controllable such that with a larger value, an excessive increase in the number of groups can be prevented. As different federated learning tasks have different preferences on the average idle time and group creation cost, we define a non-negative parameter α to adjust the preference in the objective function. A larger α indicates that the root NWDAF is not particularly concerned about average idle time and vice versa. On the other hand, as the α increases, optimal number of m decreases due to the increased cost of m. Furthermore, the number of m* also increases with the increased number of MEC servers. In [30], the authors defined the objective function as two parts: The average time cost and the the average energy consumption as the weighted sum of cost as $\alpha T + E$, where α is the weight of quality of experience (QoE), $\alpha > 0$.

Based on the cost function (7), we define the multi-objective optimization problem as

$$\min_{b_{i,j}, m} C(b_{i,j}, m) = \sum_j \sum_i \frac{t_{idle_{i,j}}}{I} + \alpha \cdot m, \tag{8}$$

$$s.t. : \sum_j b_{i,j} = 1, \tag{9}$$

$$b_{i,j} \in \{0, 1\}, \tag{10}$$

$$1 \leqslant m \leqslant M_{max}, \tag{11}$$

where in (9) and (10), $b_{i,j}$ should be a binary value such as a 0 or 1. In addition, the MEC server should be associated with a single group rather than with multiple groups. In addition, m is the total number of groups to be created between 1 (the same as with the cluster without a group) and M_{max}. Here, the maximum value M_{max} should be less than or equal to $I/2$ because we assume that in each group, there should be at least two MEC servers.

B. Proposed FedGM Scheme

In this subsection, we propose an FedGM scheme that solves the above optimization problem. Because the optimization problem has the form of a binary integer variable and a

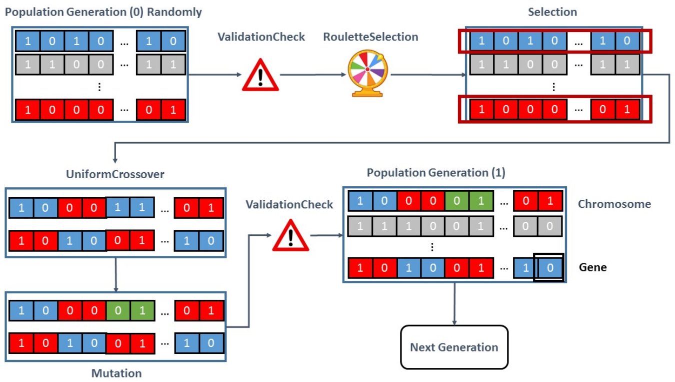

Fig. 2. Genetic process.

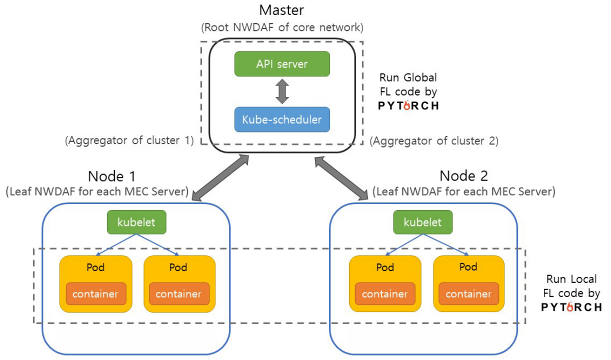
Master
(Root NWDAF of core network)
API server
Kube-scheduler
Run Global
FL code by
PYTORCH
(Aggregator of cluster 1)
(Aggregator of cluster 2)
Node 1
(Leaf NWDAF for each MEC Server)
Node 2
(Leaf NWDAF for each MEC Server)
kubelet
Pod
container
Run Local
FL code by
PYTORCH

Fig. 3. Implementation guideline of the proposed FedGM on Kubernetes platform.

floor function, the formulated problem is a challenging non-convex problem. In our formulated problem, the value of the objective function changes by varying m and $b_{i,j}$, which can be converted into multiple potential solution spaces. As shown in Fig. 2, a genetic algorithm can be considered a suitable heuristic algorithm for this non-convex problem with multiple potential solution spaces [31], [32].

To achieve an efficient design, by knowing that the search space for m is relatively small, we provided an exhaustive search-based genetic algorithm. The entire procedure of the genetic algorithm is summarized in Algorithm I. With m given in line 1 for an exhaustive search, several chromosomes $ValidationCheck$ (CH) indicating $b_{i,j}$ are combined to form a population POP; this population set is called a generation in line 2. To satisfy the constraints in the optimization problem, the chromosomes should be validated in lines 3 and 15 using a CH. Over the generations in line 5, the offspring chromosomes are generated by combining the two parent chromosomes of the current generation to optimize the fitness function value. The specifications are as follows.

- **Fitness function value**: The negative of cost function at the optimization problem (8) becomes the fitness function value F in the Algorithm I.
- **Selection**: The selection procedure is conducted to produce the next generation of offspring by selecting a genetically superior parent (line 6). Here, we utilize the roulette selection as a stochastic selection approach, wherein the probability for the selection of an individual is proportional to its fitness value. Thus, the $RouletteSelection$ function is a function of CH, F, POP, and r_{select}, where r_{select} $(0 \leqslant r_{select} \leqslant 1)$ is the selection ratio. Thus, the number of chromosomes in $\widehat{CH}$ is controlled by $POP \cdot r_{select}$.
- **Crossover**: The crossover procedure produces better offspring Ch_0 for the next generation by combining two parent chromosomes (Ch_{p1}, Ch_{p2}) (lines 9–11). Here, we utilize the uniform crossover, wherein each bit is chosen from either parent with equal probability.
- **Mutation**: To avoid the local optimum, if the condition in line 13 is satisfied, the genetic algorithm performs a mutation operation on the chromosome.

1) Summary of the general procedure for the proposed algorithm: Based on this procedure as shown in Fig. 2, our proposed genetic algorithm is conducted on the management module in the root NWDAF section of core network. By implementing the binary variable as input values of the genetic algorithm based on group association of the MEC servers $(b_{i,j})$. Each binary variable is a gene in each chromosome, we generate the initial population (0) randomly as represented (line 2). Then in (line 3) we apply the ValidationCheck on the generated population and after getting acceptance, by running the RouletteSelection with the prioritized fitness value in (line 6); we select two chromosomes as parents (lines 8–10). By applying the UniformCrossover on the parents, we conduct the mutation step (line 11). Then, we conduct the ValidationCheck again on the chromosomes before generating the population (1) (line 15). Finally, the first round ends and we send the chromosomes to the second round to create a new generation (lines 18–21). This general process will be iterated until the maximum number of the optimal fitness value for chromosomes (optimal association rule) is reached (lines 22–24).

Finally, after the algorithm converges, the suboptimal chromosomes $Ch_{opt,m*}$ and $m*$ are returned as output S.

C. Discussion on Implementation of the Proposed FedGM on Virtualization Platform : Kubernetes

For practical concern, this subsection provides discussions on how to implement the proposed FedGM on Docker based Kubernetes platforms. Similar to our previous study of [33], the proposed FedGM can be implemented on Kubernetes platform which includes a master and two nodes. Fig. 3 shows the master represented as the root NWDAF of core network, and the two other nodes represented as two aggregator nodes of two clusters of MEC servers with leaf NWDAF for each MEC server which represents for local training. The master is responsible for operating the two other nodes using Kubernetes, and after receiving the local updates from clusters, first, aggregates and then, updates the global model. Additionally, each of two nodes is responsible for running pods. Inside a pod, there is a container which has the FedGM global model that receives from the master. Therefore, the pod conducts local training by running the FedGM on the container.

Algorithm 1 Proposed FedGM

```
Input: $I$, $M_{max}$, $t_i$, $POP$, $G$
Output: Best chromosome $S(b^*_{i,j}, m^*)$
for $m \leftarrow 1$ to $M_{max}$ do
    $CH \leftarrow \{Ch_i | \forall i \in \{1, \cdots, POP\}\}$ randomly generated
    $CH \leftarrow ValidationCheck(CH)$
    $F \leftarrow \{\varepsilon(Ch_i) | \forall Ch_i \in CH\}$
    for $g \leftarrow 1$ to $G$ do
        $\widehat{CH} \leftarrow Rouletteselection(CH, F, POP, r_{select})$
        $CH \leftarrow \widehat{CH}$
        for $o \leftarrow 1$ to $POP \cdot (1 - r_{select})$ do
            $Ch_{p1} \in \widehat{CH}$
            $Ch_{p2} \in \widehat{CH}/Ch_{p1}$
            $Ch_o \leftarrow UniformCrossover(Ch_{p1}, Ch_{p2})$
            if $U(0,1) \leqslant r_{mutate}$ then
                $Ch_o \leftarrow Mutation(Ch_o)$
            end if
            $Ch_o \leftarrow ValidationCheck(Ch_o)$
            $CH \leftarrow CH \cup \{Ch_o\}$
        end for
        $F \leftarrow \{\varepsilon(Ch_i) | \forall Ch_i \in CH\}$
    end for
    $(F_{opt,m}, Ch_{opt,m}) \leftarrow (\max F, \arg\max_{Ch} F)$
end for
$(F_{opt}, m^*) \leftarrow (\max F_{opt_m}, \arg\max_{m} F_{opt,m})$
$S \leftarrow (Ch_{opt,m*}, m^*)$

return S
```

In the following, we will describe the FedGM implementation process that works with the FL approach. First, the master sends the initial global model to the two nodes (clusters). Then, the MEC servers located in the different groups of each node also save the initial global model (The details of the process have already been described in the previous sections). After updating the local models based on the initial global model by MEC servers and sending back to the aggregation nodes of the master; aggregation node of each cluster aggregates the updated local models received from the group in each cluster. Finally, the master updates the global model and sends new global model to all nodes. This process will be repeated until the final global model converges to the target accuracy.

V. Performance Evaluation

This section provides the simulation results verifying the effectiveness of the proposed FedGM under various parameter settings. We set up several benchmark algorithms: 1) a clustered FL without group management, denoted as benchmark 1, and 2) two variants of a clustered FL with group management using different combinations of m and $b_{i,j}$, denoted as benchmarks 2 and 3. Specifically, in benchmark 2 (benchmark 3), the values of m and $b_{i,j}$ can be M_{max} (the optimal value) and a random value (the optimal value), respectively.

A. Evaluation of Proposed Cost Model

Fig. 4(a) depicts how Algorithm 1 for the proposed FedGM converges to the optimum fitness value for minimizing the objective function according to r_{select} over the generations. The constant settings of the proposed fitness model were derived using a Monte Carlo simulation with 20 random samples, where f_{min} and f_{max} were 1 and 5 GHz, respectively. We can observe that the proposed FedGM converges to a specific value as the generation is iterated, regardless of r_{select}. Here, if r_{select} is small (e.g., 0.10), the cost value is large because it may fall into the local optimum. This is because a small value of r_{select} indicates the selection of few parental chromosomes. Fig. 4(b) shows the effect of the proposed scheme with different values of α (0.01, 0.1, 0.5, and 0.9) in a cluster considering different number of MEC servers (6, 10, 20). As the α increases, optimal number of m decreases due to the increased cost of m. Furthermore, the number of m^* also increases with the increased number of MEC servers.

Fig. 5(a) demonstrates the effect of the proposed FedGM on the basis of the proposed cost model. The evaluation of the proposed cost model (C) was conducted using a Monte Carlo simulation with 20 random samples (i.e., random settings of [1 MB, 5 MB]). As shown in Fig. 5(a), the proposed FedGM outperforms the benchmarks: Compared to benchmark 1 (without group management), significant gain can be achieved in terms of C, as shown by the dashed line. This result is attributable to the fact that by introducing the proposed grouping management with an optimal number of groups to be created and a group association of the MEC servers, which minimizes the average idle time and the group creation cost over such servers, the MEC servers are able to maximize the time spent on useful local training. As mentioned in benchmark 2, the number of groups to be created is maximum (half of the MEC servers), which is obvious that the cost is lower than benchmarks 1 and 3, respectively. The cost of benchmark 2 is especially less than benchmark 3 because of the maximum number of groups. However, compared with the proposed scheme, benchmark 2 has much cost value because the $b_{i,j}$ is random, while the proposed scheme optimizes $b_{i,j}$ to obtain an optimal value. Correspondingly, Fig. 5(b) shows the impact of the performance of proposed FedGM model on the average idle time considering the different number of MEC servers (6, 10, 20) based on iterations. The average idle time in benchmark 1 has the highest value compared to other cases since there is no group management in the cluster. And, since benchmark 2 and 3 are not jointly optimized in terms of both m and $b_{i,j}$, the proposed FedGM achieves the lowest average idle time. The performance gap compared to the benchmarks increases as the number of MEC servers increases. Fig. 5(c) demonstrates how optimal m^* is obtained according to the number of MEC servers. Here, the proposed FedGM, by alleviating excessive number of m, suitable m is obtained to balance between the average idle time and the cost of m.

B. Evaluation of Accuracy

In this subsection, we describe the accuracy the evaluation conducted to address the communication rounds versus the

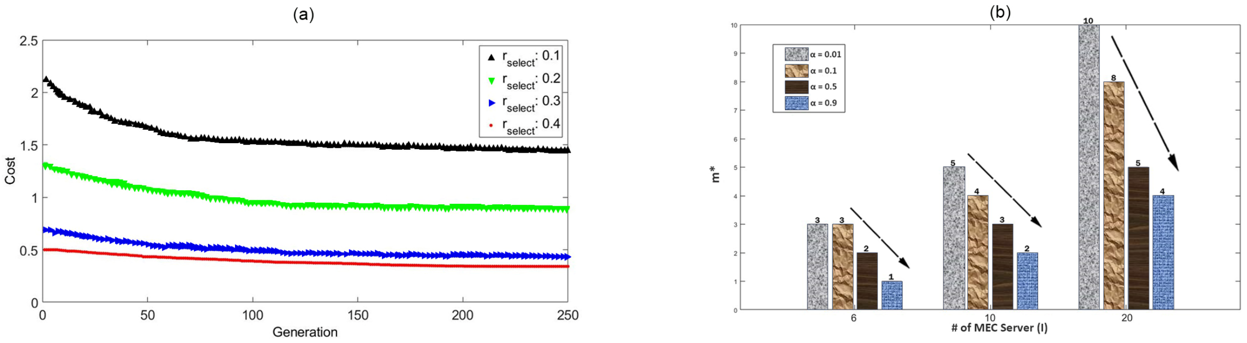

Fig. 4. (a) Convergence analysis of the proposed scheme according to $r_{rselect}$. (b) Performance of the proposed scheme according to α.

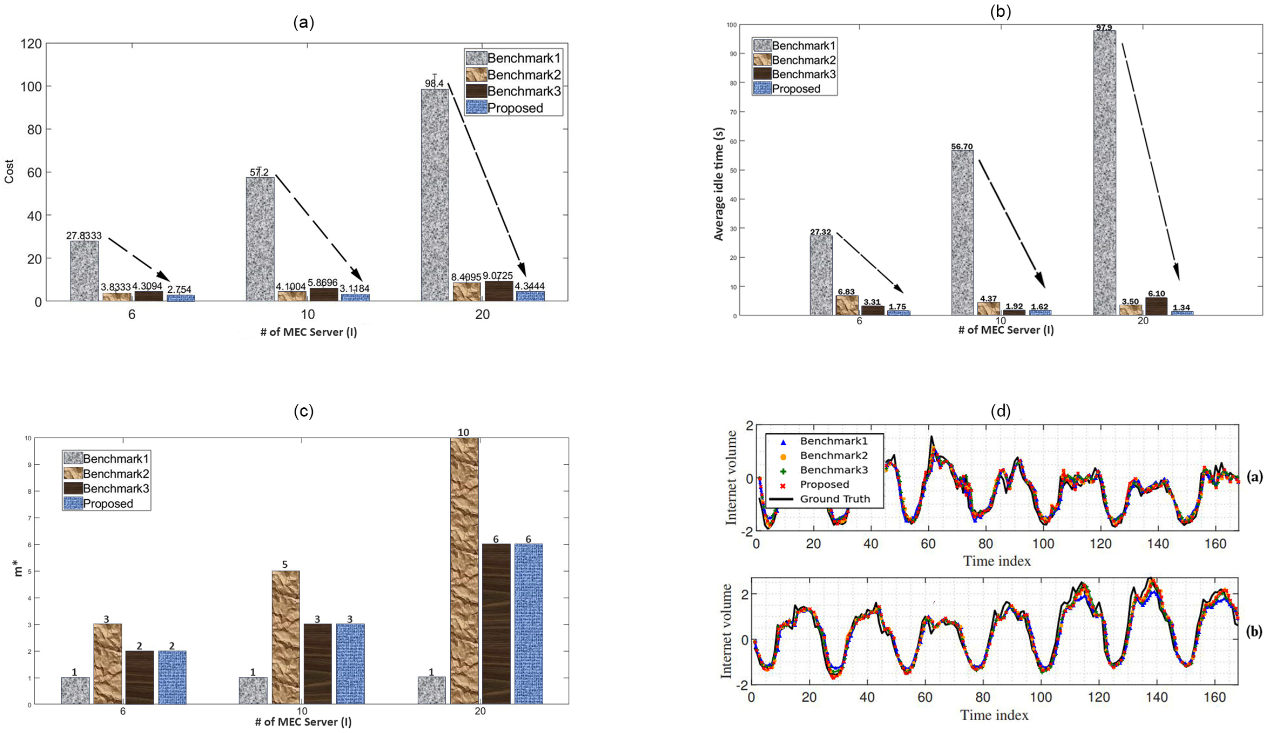

Fig. 5. (a) Performance of proposed cost model (considering different number of MEC servers). (b) Performance of the proposed scheme according to average idle time. (c) Performance of the proposed scheme according to $m*$. (d) Prediction results on the (a-upper) Milano dataset and (b-lower) Trento dataset.

TABLE III
PREDICTION PERFORMANCE COMPARISONS AMONG DIFFERENT METHODS IN TERMS OF MSE AND MAE ON TWO DATASETS.

Methods	Milano		Trento	
	MSE	MAE	MSE	MAE
Lasso	0.4380	0.5475	5.9121	1.5391
SVR	0.1036	0.2220	5.9080	1.0470
LSTM(Distributed)	0.1697	0.2936	4.6976	1.1193
LSTM(Centralized)	0.3681	0.4588	6.4016	1.5196
FedAvg	0.1096	0.2319	4.7988	1.0668
FedDA	0.1033	0.2211	2.4473	0.7471
FedGM(Proposed)	0.1026	0.2207	2.4068	0.6050

accuracy of the proposed FedGM. The proposed FedGM and all benchmarks were implemented based on FedDA [11], which is the latest FL implementation adopted for mobile traffic prediction. In our practical evaluation, as in [11], we utilized the dataset for the mobile traffic prediction simulation, which comes from the *Big Data Challenge* launched by Telecom Italia and is mainly composed of call detail records (CDR) [34]. Specifically, there are two datasets of two areas in Italy: Milano (a grid with 10,000 cells) and Trento (a grid with 6,575 cells). In the dataset, we only utilized mobile traffic corresponding to Internet services. We randomly selected 20 MEC servers located in the same cluster from each dataset and conducted experiments on them. We used the first 7 weeks' worth of traffic to train the prediction models and applied the traffic from the last week for the test round with 8,750 and 5,753 training data and 1,250 and 822 testing data for Milano and Trento, respectively. The experimental results of various prediction methods are presented in Table III.

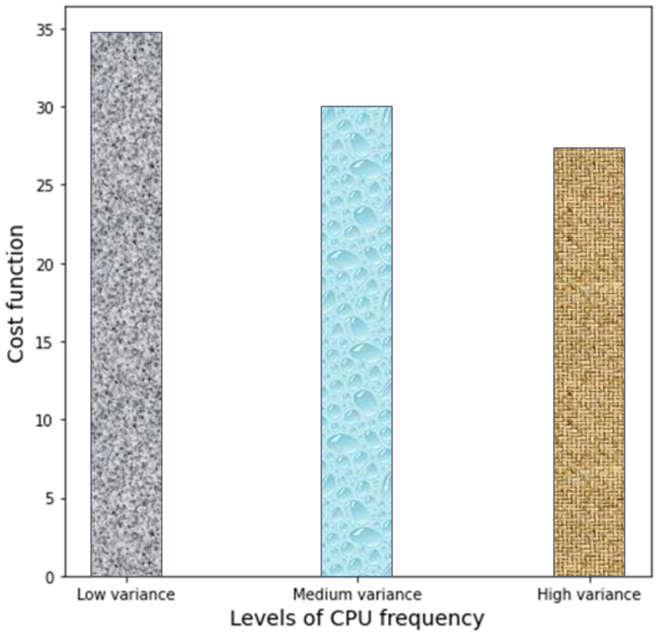

Fig. 6. Performance of proposed cost model (considering different levels of CPU frequency over MEC servers).

We implemented Lasso, SVR, LSTM, FedAvg, FedDA, and FedGM (the proposed scheme) and compared their results in Table III to provide a clear comparison between the different methods. (Overall, it can be observed that the proposed FedGM method outperforms all the other methods in terms of both MSE and MAE on both datasets. Specifically, the FedGM method achieves an MSE of 0.1026 and 0.2207 and an MAE of 0.6050 and 2.4068 on Milano and Trento datasets, respectively. In contrast, the other methods, such as Lasso, SVR, and LSTM models, show higher MSE and MAE values, indicating lower prediction accuracy. Therefore, the proposed FedGM method seems to be a promising approach for predicting outcomes on these datasets.) Additionally, it is worth mentioning that the Trento dataset exhibited a higher variance between data in most parts of the Internet section compared to the Milano dataset, despite similar settings. This observation suggests that the two datasets have different distributions. However, due to the higher number of data points in the Milano dataset, better correlations and results were obtained for the Internet section of the Milano dataset. Overall, the proposed scheme demonstrated superior performance compared to other algorithms, which can be attributed to its optimal number of groups and group creation cost.

In Fig. 5(d), prediction results of the Internet traffic volume of randomly selected MEC servers on the (a-upper) Milano dataset and (b-lower) Trento dataset are given. By observing Fig. 5(d), we can tell that the proposed FedGM obtains a consistent and similar prediction performance as the benchmarks while predicting the ground truth well. Furthermore, Fig. 6 displays the minimum and maximum CPU frequencies of 6 MEC servers in the test dataset from the 'Internet' section of Milano and Trento datasets. The results indicate that when the variance is in the range (Min.: 42432.16 & Max.: 402265.69), the variance value in each case was as follows:

- Low variance: 14675.32
- Medium variance: 22716.87
- High variance: 94736.53

Based on the data, it can be concluded that as the variance of the CPU frequencies increases, the cost function decreases. This implies that a higher variance in CPU frequencies results in a more efficient allocation of tasks to MEC servers. Moreover, it is important to note that this factor affects latency (idle time). Additionally, as shown in Fig. 7, we tested the R-squared score for the prediction accuracy of the proposed FedGM and the benchmarks with respect to the communication rounds, where the R-squared score represents the exactness by reflecting how well the ground-truth values are anticipated as a promising statistical measure [35]. Specifically, as shown in Figs. 7(a) and (b), we can observe that the proposed FedGM achieves a higher accuracy on both datasets compared to the other benchmarks. Its advantages are clearer on the Trento dataset. More importantly, the proposed FedGM requires fewer communication rounds to achieve a certain prediction accuracy than the other benchmarks. In Fig. 7(a), after 10 communication rounds, the proposed FedGM can achieve and accuracy of 0.92 for Internet traffic. For benchmarks 2, 3, and 1 during this round, the accuracies for achieved Internet traffic were 0.86, 0.83, and 0.81, respectively. In addition, as shown in Fig. 7(b), the proposed FedGM has a better R-squared score than other benchmarks throughout the communication rounds.

Moreover, we also tested the root mean squared error (RMSE) for the prediction accuracy of the proposed FedGM and the benchmarks, where the RMSE is a useful metric for evaluating the overall performance of a regression model, taking into account both accuracy and precision. Fig. 8(a) presents the RMSE results for benchmarks 1 to 3 and the proposed scheme on the Milano dataset. The proposed scheme achieved the lowest RMSE of 0.0971, outperforming benchmarks 1 to 3 which achieved RMSE values of 0.1020, 0.0983, and 0.0998, respectively. Similarly, in Fig. 8(b) on the Trento dataset, the proposed scheme achieved the lowest RMSE value of 0.1384, while benchmarks 1 to 3 achieved RMSE values of 0.1495, 0.1407, and 0.1478, respectively. These results demonstrate the superior predictive performance of the proposed scheme compared to the existing benchmarks on both datasets.

Finally, we can conclude that the proposed FedGM achieves better convergence speeds while providing a similar or better prediction accuracy, depending on the dataset, by introducing optimal group management into the cluster.

C. Discussion

It is possible that if stragglers have meaningful data that gets excluded, this could lead to a loss of accuracy. Nevertheless, our proposed clustered FL group management scheme aims to minimize the impact of stragglers by optimizing the group association of MEC servers and reducing their average idle time. This approach ensures that the contributions of all participating devices, including stragglers, are efficiently utilized during the

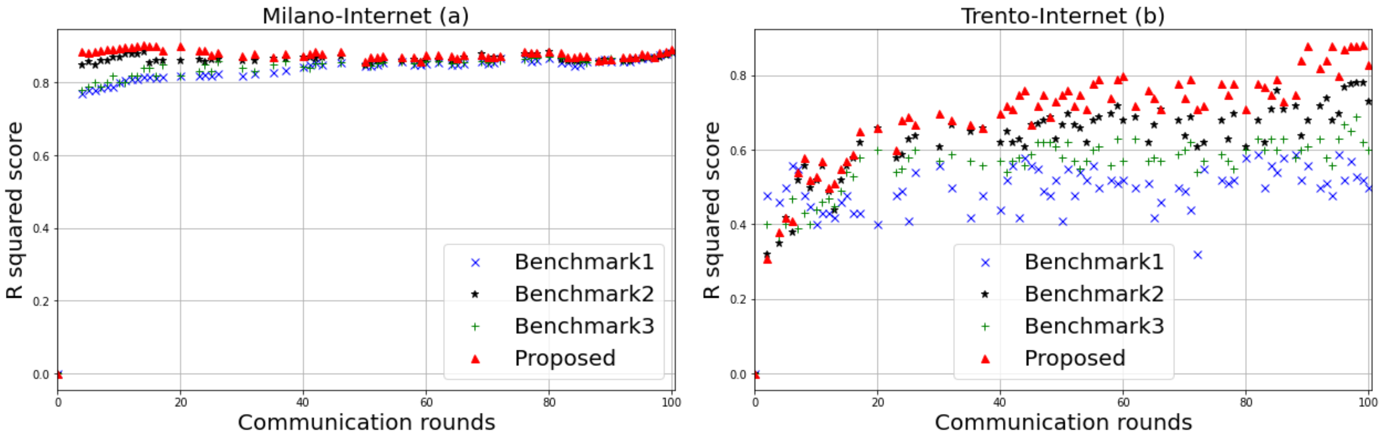

Fig. 7. Prediction accuracy versus communication rounds: (a) Milano and (b) Trento.

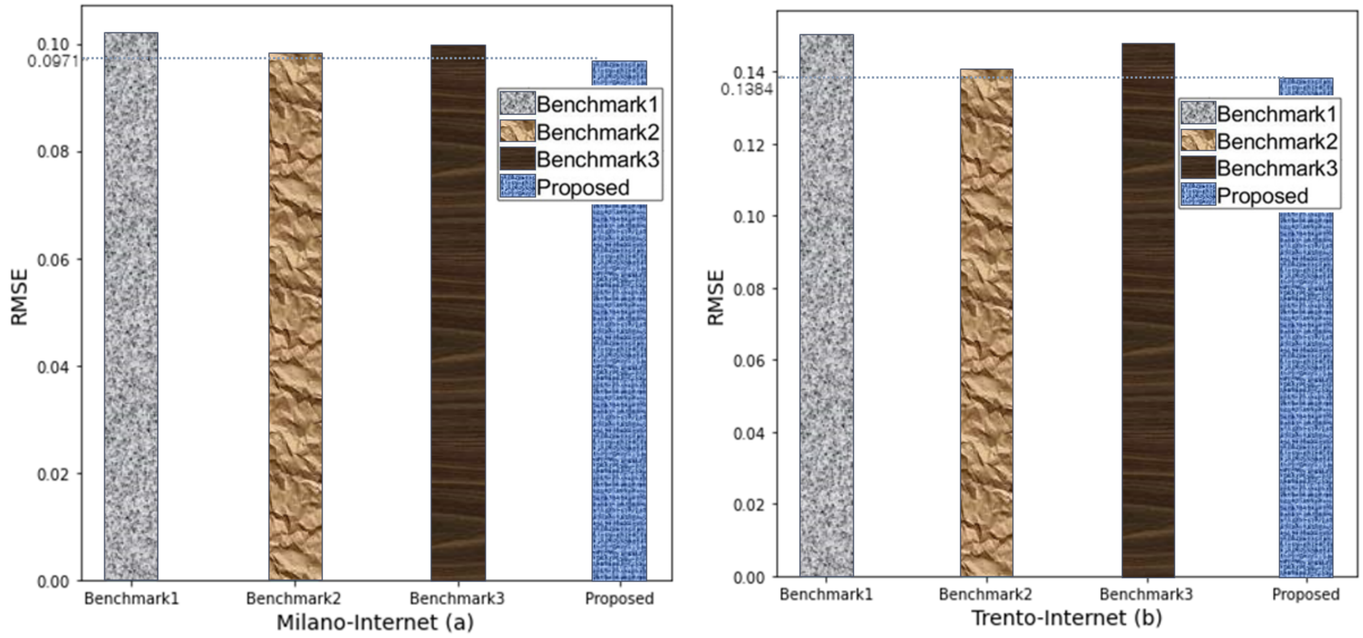

Fig. 8. Comparing prediction accuracy across multiple benchmarks: (a) Milano and (b) Trento.

local training process. By optimizing the group association of MEC servers, the proposed scheme also reduces the likelihood of idle time, which is an important factor in minimizing the impact of stragglers. Moreover, the proposed scheme allows for flexibility in the number of groups created, which can help mitigate the impact of stragglers. By creating multiple groups, the scheme can balance the workload across different groups and reduce the likelihood of any one group being heavily impacted by stragglers. This, in turn, can improve the overall convergence time and accuracy of the FL model.

VI. Conclusion

In this study, to improve the convergence speeds for mobile traffic prediction by alleviating the straggler impact due to the heterogeneity of MEC servers, we developed a FedGM scheme that conducts group management through the optimal control of i) the number of groups to be created and ii) the group association of MEC servers. Our simulation results for real-world mobile traffic datasets show that the convergence time of the proposed FedGM is lower than that of the benchmarks. The optimization problem was designed using a non-convex problem, and a genetic-based heuristic approach was proposed for determining a suboptimal solution. By reducing the average idle time to increase the workload of the MEC servers under two real-world mobile traffic datasets, the experimental results showed that FedGM outperforms previous state-of-the-art methods in terms of convergence speed with an acceptable loss of accuracy.

References

[1] A. Chouman, D. M. Manias, and A. Shami, “Towards supporting intelligence in 5G/6G core networks: NWDAF implementation and initial analysis,” *arXiv preprint arXiv*, p. 2205.15121, 2022.

[2] S. Zhang, H. Zhang, and L. Song, “Beyond D2D: Full dimension UAV-to-everything communications in 6G,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6592–6602, Jun. 2020.

[3] L. M. M. Zorello *et al.*, "Baseband-function placement with multi-task traffic prediction for 5G radio access networks," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 5104–5119, Dec. 2022.

[4] U. Paul, J. Liu, S. Troia, O. Falowo, and G. Maier, "Traffic-profile and machine learning based regional data center design and operation for 5G network," *J. Commun. Netw.*, vol. 21, no. 6, pp. 569–583, Dec. 2019.

[5] X. Qiao, Y. Huang, S. Dustdar, and J. Chen, "6G vision: An AI-driven decentralized network and service architecture," *IEEE Internet Comput.*, vol. 24, no. 4, pp. 33–40, Apr. 2020.

[6] D. Andreoletti *et al.*, "Network traffic prediction based on diffusion convolutional recurrent neural networks." in *Proc. IEEE INFOCOM*, Apr. 2019.

[7] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1–24, Sep. 2021.

[8] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards efficient synchronous federated training: A survey on system optimization strategies," *IEEE Trans. Big Data*, vol. 9, no. 2, pp. 437–454, Apr. 2023.

[9] L. Yang, Y. Gan, J. Cao, and Z. Wang, "Optimizing aggregation frequency for hierarchical model training in heterogeneous edge computing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 7, pp. 4181–4194, Jul. 2023.

[10] J. Kim, D. Kim, J. Lee, and J. Hwang, "A novel joint dataset and computation management scheme for energy-efficient federated learning in mobile edge computing," *IEEE Wireless Commun. Lett.*, vol. 11, no. 5, pp. 898–902, May 2022.

[11] C. Zhang, S. Dang, B. Shihada, and M. S. Alouini, "Dual attention-based federated learning for wireless traffic prediction," in *Proc. IEEE INFOCOM*, May 2021.

[12] R. J. Hyndman and G. Athanasopoulos, "Forecasting: Principles and practice," *Scientific data*, vol. 2, 2018.

[13] J. S. Hunter, "The exponentially weighted moving average," *J. Quality Technol.*, vol. 18, no. 4, pp. 203–210, Apr. 1986.

[14] D. R. Cox, "Principles of statistical inference," *Cambridge university press*, 2006.

[15] Y. Shu, M. Yu, O. Yang, J. Liu, and H. Feng, "Wireless traffic modeling and prediction using seasonal Altiparmak models," *IEICE Trans. Commun.*, vol. 88, no. 10, pp. 3992–3999, Oct. 2005.

[16] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, "Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, Jun. 2019.

[17] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," *ACM SIGMETRICS Performance Evaluation Review*, vol. 39, no. 1, pp. 265–276, Jan. 2011.

[18] R. Li, Z. Zhao, A. X. Zhou, J. Palicot, and H. Zhang, "The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 234–240, Jun. 2014.

[19] X. Chen, Y. Jin, A. S. Qiang, W. Hu, and K. Jiang, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *Proc. IEEE ICC*, Jun. 2015.

[20] R. Li *et al.*, "The learning and prediction of application-level traffic data in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3899–3912, Jun. 2017.

[21] Y. Yamada, R. Shinkuma, T. Sato, and E. Oki, "Feature-selection based data prioritization in mobile traffic prediction using machine learning," in *Proc. IEEE GLOBECOM*, Dec. 2018.

[22] J. Wang *et al.*, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE INFOCOM*, May 2017.

[23] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, "Spatio-temporal wireless traffic prediction with recurrent neural network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, Apr. 2018.

[24] M. L. Hachemi, A. Ghomari, Y. Hadjadj-Aoul, and G. Rubino, "Mobile traffic forecasting using a combined FFT/LSTM strategy in SDN networks," in *Proc. IEEE HPSR*, Jun. 2021.

[25] L. Zhang, C. Zhang, and B. Shihada, "Efficient wireless traffic prediction at the edge: A federated meta-learning approach," *IEEE Commun. Lett.*, vol. 26, no. 7, pp. 1573–1577, Apr. 2022.

[26] N. I. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, Feb. 2009.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *In Proc. AISTATS*, Apr. 2017.

[28] Y. Jeon *et al.*, "A distributed NWDAF architecture for federated learning in 5G," in *Proc. IEEE ICCE*, Jan. 2022.

[29] Z. Yang, M. Chen, W. Saad, C. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2020.

[30] Y. Dong, S. Guo, J. Liu, and Y. Yang, "Energy-efficient fair cooperation fog computing in mobile edge networks for smart city," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7543–7554, May 2019.

[31] M. Gen, F. Altiparmak, and L. Lin, "A genetic algorithm for two-stage transportation problem using priority-based encoding," *OR spectrum*, vol. 28, no. 3, pp. 337–354, Mar. 2006.

[32] S. Lee, J. Lee, H. S. Park, and J. K. Choi, "A novel fair and scalable relay control scheme for internet of things in lora-based low-power wide-area networks," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5985–6001, Jun. 2020.

[33] J. Kim, D. Kim, and J. Lee, "Design and implementation of Kubernetes enabled federated learning platform," in *Proc. ICTC*, Oct. 2021.

[34] G. Barlacchi *et al.*, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Scientific data*, vol. 2, no. 1, pp. 1–15, Jan. 2015.

[35] A. Cameron and F. A. G. Windmeijer, "R-squared measures for count data regression models with applications to health-care utilization," *J. Business Economic Statistics*, vol. 14, no. 2, pp. 209–220, Feb. 1996.

Faranaksadat Solat received the B.S degree in Industrial Engineering from the Alzahra University, and M.S degree in Information Technology Engineering from the Iran University of Science and Technology (IUST), Tehran, Iran, in 2017 and 2019 respectively. She is currently a student pursuing a Ph.D. in the School of Computing, at Gachon University, Seongnam, South Korea, in 2022. Her interested research fields are system optimization, machine learning, FL, 5G/6G network management, and reinforcement learning.

Tae Yeon Kim received his BS and MS degree in computer engineering from the Chung-Ang University, Seoul, Rep. of Korea, in 1990 and 1992 respectively, and his PhD in Network Engineering from the Chungbuk National University, Cheongju, Rep. of Korea, in 2008. He is currently leading the Network Intelligence Research Section at the ETRI, Rep. of Korea. He has been working in the area of programmable networks with network functions virtualization and software defined networking technologies since 2012. His interests include AI-native networking technologies for 6G networks.

Joohyung Lee received the B.S., M.S., and Ph.D. degrees from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2008, 2010, and 2014, respectively. From 2012 to 2013, he was a Visiting Researcher with the Information Engineering Group, Department of Electronic Engineering, City University of Hong Kong, Hong Kong. From 2014 to 2017, he was a Senior Engineer with Samsung Electronics. He is currently an Associate Professor with the School of Computing, Gachon University, South Korea and also a Visiting Fellow with the Department of Electrical and Computer Engineering, Princeton University. He has contributed several articles to the International Telecommunication Union Telecommunication (ITU-T) and the 3rd Generation Partnership Project (3GPP). His current research interests include resource allocation, optimization and protocol design, with a focus on resource management for machine learning, including federated learning, 6G networks, cloud/edge computing, smart grids, augmented reality, virtual reality, and network economics. He received the Best Paper Award at the Integrated Communications, Navigation, and Surveillance Conference, in 2011, and Award for outstanding contribution in reviewing at Elsevier Computer Communications. He has been an IEEE Senior member since 2019, and a Technical Reviewer for several conferences and journals.