

Received June 19, 2021, accepted July 25, 2021, date of publication August 2, 2021, date of current version August 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101871

# Reliable Federated Learning Systems Based on Intelligent Resource Sharing Scheme for Big Data Internet of Things

SA MATH<sup>1</sup>, PROHIM TAM<sup>1</sup>, AND SEOKHOON KIM<sup>1,2</sup>, (Member, IEEE)

<sup>1</sup>Department of Software Convergence, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea

<sup>2</sup>Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do 31538, Republic of Korea

Corresponding author: Seokhoon Kim (seokhoon@sch.ac.kr)

This work was funded by BK21 FOUR (Fostering Outstanding Universities for Research) (no. 5199990914048), and this research was supported by the Bio and Medical Technology Development Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. NRF-2019M3E5D1A02069073). In addition, this work was supported by the Soonchunhyang University Research Fund.

**ABSTRACT** Federated learning (FL) is the up-to-date approach for privacy constraints Internet of Things (IoT) applications in next-generation mobile network (NGMN), 5<sup>th</sup> generation (5G), and 6<sup>th</sup> generation (6G), respectively. Due to 5G/6G is based on new radio (NR) technology, the multiple-input and multiple-output (MIMO) of radio services for heterogeneous IoT devices have been performed. The autonomous resource allocation and the intelligent quality of service class identity (IQCI) in mobile networks based on FL systems are obligated to meet the requirements of privacy constraints of IoT applications. In massive FL communications, the heterogeneous local devices propagate their local models and parameters over 5G/6G networks to the aggregation servers in edge cloud areas. Therefore, the assurance of network reliability is compulsory to facilitate end-to-end (E2E) reliability of FL communications and provide the satisfaction of model decisions. This paper proposed an intelligent lightweight scheme based on the reference software-defined networking (SDN) architecture to handle the massive FL communications between clients and aggregators to meet the mentioned perspectives. The handling method adjusts the model parameters and batches size of the individual client to reflect the apparent network conditions classified by the k-nearest neighbor (KNN) algorithm. The proposed system showed notable experimented metrics, including the E2E FL communication latency, throughput, system reliability, and model accuracy.

**INDEX TERMS** Big data, federated learning, massive Internet of Things, machine learning, software-defined network.

## I. INTRODUCTION

### A. BACKGROUND

Mobile services have become the over-the-top (OTT) application in the current mobile network. The new radio (NR) enhances the future radio stations and strengthens radio power with the millimeter and micrometer wavelength technologies. Network generation mobile network (NGMN) demonstrates many modernities of opportunity and challenge issues in a variety of applications, including Internet of Vehicle (IoV), intelligent wireless sensor networks (WSN), wireless body area network (WBAN), autonomous Internet of Things (AIoT), etc. [1]–[3]. The aforementioned applications

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaoqing Pan<sup>1</sup>.

will generate big data in 5G/6G areas. The critical cloud infrastructures, services, and platforms of mobile cloud computing (MCC) are migrated to a local cloud, namely mobile edge computing (MEC) [4], [5]. MEC is the crucial enabler of technologies for network slicing (NS) and local computing resource for mobile user services.

Moreover, network function virtualization (NFV), software-defined networking (SDN), and machine learning (ML) algorithms share the complementary to enable intent-based edge cloud service and intelligent edge clouds [6]. ML algorithms play an essential function for edge computing in terms of content-based caching, radio resource classification, management, and orchestration, edge network resources optimization, and performs autonomous handling for big data network environments. It is also

critical for next-generation self-organizing networking (SON) and efficient resource configuration for heterogeneous user services. However, due to the privacy constraint of user data, the applied ML algorithms suffer from many challenges since they require computation in a distributed cloud [7]. User data privacy is the core challenge among the other suffering, including model limitation, training time, limitation of computation resource, and model reliability. Federated learning (FL) systems are appropriate for massive user data, can cope with the challenges of data privacy since the user data will not be shared with the distributed or third party [8]–[10]. The training of the local data set of FL models will perform on the local device, while each client has its model to train its dataset. The clients distribute only their local model and model parameters to the aggregation server to define a global model. In the HetIoT environments, big data will be in local areas since the client has computing resource limitations to train the model. Thus, partial training in the local model is required to overcome the overloading of computing resources. Moreover, the radio network reliability will improve influent FL model reliability, while the lower failure in communications network will dramatically lessening the accuracy of the global model [11]. This paper proposed an intelligent resource allocation based on a lightweight ML algorithm, namely k-nearest neighbor (KNN). KNN performs classification tasks to identify communication gateways statuses in congested situations, While the SDN controller utilized the classified metrics to determine forwarding paths.

## B. CONTRIBUTIONS

In this paper, we propose a lightweight approach for effective resource allocation to enhance the reliability of the FL model in big data IoT communication networks. The proposed scheme guarantees E2E communication reliability for model transferring between clients and aggregation servers. In multiple aggregation servers and future edge cloud systems, the resource limitation problem can occur during massive services will be launched to cope with the gigantic user requests. MEC server will slice the physical resource into multiple virtual machines based on NFV architecture. Every aggregation server has a dedicated computing resource that requires monitoring and adjusting from the SDN controller in terms of resource management and orchestration. While the virtual computing will attach with the NR stations for radio service operations, the failures of model transferring can occur whenever radio gateway and attached MEC server resources are overloaded operations. The main contributions of the paper have presented as follows:

- 1) Global network status monitoring by the monitoring module provides the global view of the SDN controller. The MEC server was proposed for storing the gathered radio gateway statuses for processing. Furthermore, the collected network statuses turn to the classification phase for distinguishing distinct conditions of the serving gateways, which is essential for handling processes

of the network resources. An integrated lightweight ML with SDN controller to implement network configuration rules for reliable FL models communication. The model owner selects the local models partially and according to the implemented policy of the controller.

- 2) Balancing the models transferring between the participant devices and model owner servers (MEC servers): the server and client send model updates according to the network configuration role, the adjustment will be applied in any fluctuations situation network.
- 3) The forwarding rules of the SDN controller perform fault tolerance and Load-balancing techniques. SDN controller determines the feasible servers and updates the forwarding flow for the incoming requests. In addition, intelligent resource allocation establishes the cost-adoption loading-balancing that assigns the appropriate tasks to the serving gateways according to the existed resources.
- 4) We provide the system evaluations regarding FL convergence accuracy in various network conditions and E2E communication QoS metrics, including latency, communication rate, and reliability.

Table 1 provides the acronyms and abbreviations for convenient reading. The rest of the paper is organized as follows. Firstly, section II, presenting the federated architecture and its enabler technologies in big data IoT networks. And, section III addresses the proposed architecture and network handling scheme. Next, the system evaluations, results, and

**TABLE 1. Acronyms and abbreviations.**

AI	Artificial intelligence
D2D	Device-to-Device
FL	Federated learning
DT	Decision tree
E2E	End-to-End
EPC	Evolved packet core
HetIoT	Heterogenous IoT
IoT	Internet of Things
KNN	K-nearest neighbor
MCC	Mobile cloud computing
MEC	Mobile edge computing
ML	Machine learning
mmWave	Millimeter wave
NFV	Network function virtualization
NGMN	Next generation mobile network
NS	Network slicing
AIoT	Autonomous Internet of Things
IoT	Internet of Things
PGW	Packet data gateway
QoE	Quality of experience
QoS	Quality of service
RAN	Radio access network
RRH	Radio remote head
SDN	Software-defined networking
SGW	Service gateway
SON	Self-organizing networking
UHCR	Ultra-high communication reliability
ULL	Ultra-low latency
VNI	Virtual network infrastructure
WSN	Wireless sensor network

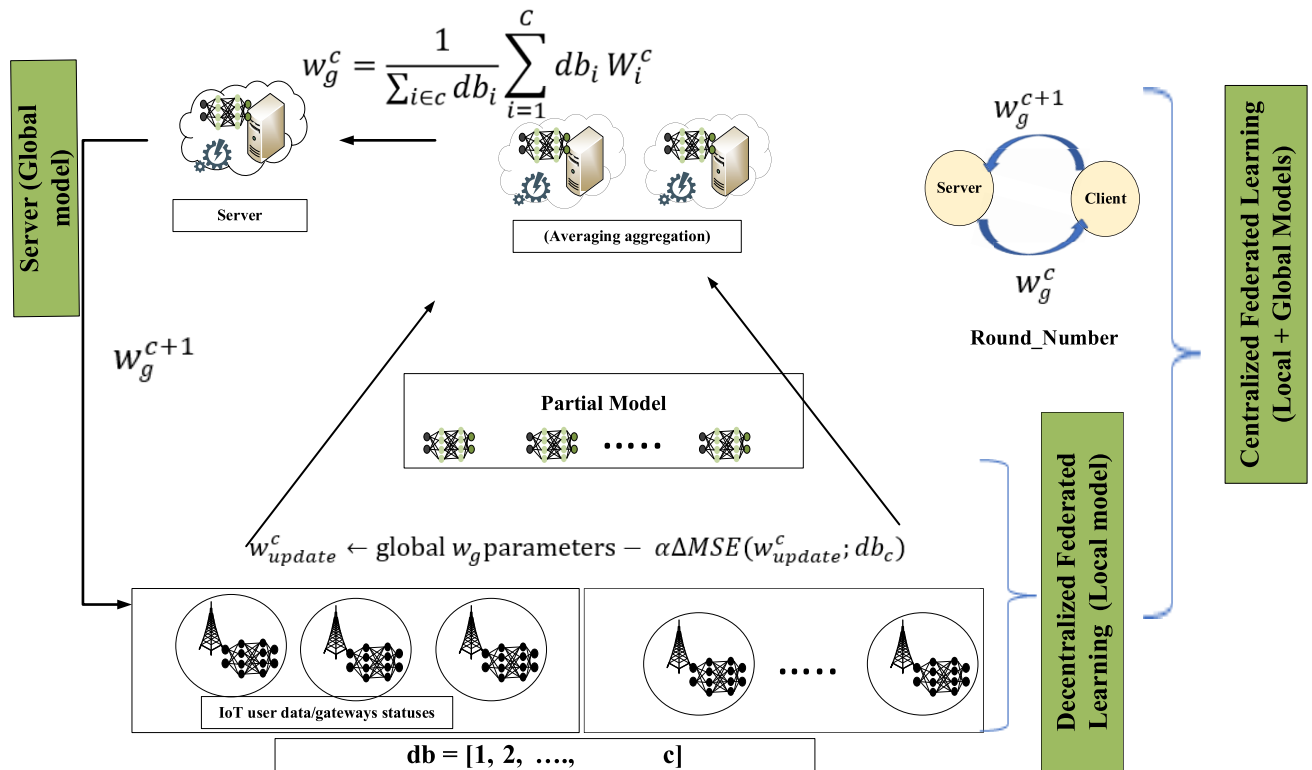


FIGURE 1. The common FL system in mobile communication in future integrated MEC server.

discussion are depicted in section IV. Finally, section V presents the conclusion.

II. SYSTEM ARCHITECTURE

In this section, the FL-based IoT network architecture and enabler technologies including MEC, SDN, NS, and ML are addressed in the following:

A. ENABLER TECHNOLOGIES FOR FL IN BIG DATA IoT

Sensor devices consist of insufficient computing resources for local ML models training, while most wireless sensor devices are targeted to sense and share lightweight information over the networks. Currently, sensor devices rely on high computing devices with sufficient storage and high processing units, including wearable devices, mobile phones, etc. [12], [13]. Due to the perspectives of NGMN objective to deliver remote cloud (MCC) to distribute in RAN areas to enable one user one cloud (OUOC) based on the MEC server. Each wireless sensor device sends its information to the local cloud. The big data will be generated by heterogeneous wireless sensor devices from vast applications, including intelligent healthcare, safety system, smart city, AIoT, IoV, etc. The high computation resource at the distributed cloud and also processing time will be raised according to the volume of the data [14]. To cope with a large volume of user data at the centralized cloud, decentralized computing based on FL models introduces partial computing and parallelism methods for processing in large-scale data environments.

Since MEC servers can be modeled both centralized and decentralized based on the FL architecture, reliable network services are significantly obligated to ensure E2E communications between the local model and aggregation server located in the distributed cloud areas [15]. Therefore, FL systems are suitable for current communication systems, especially in time constraints and privacy constraints applications. Although FL delivers essential opportunities for real-time IoT service, the local device has a small and specific dataset with more minor complicated features required to perform data cleansing and filtering [16]. At the same time, the unfeatured sensor information presents the core challenge for model accuracy and processing time that suffers from the critical application seeking ultra-low latency (ULL) for E2E communications. Moreover, the slicing of model communication between the local devices and aggregate servers will be meaningful for controlling the model transferring on the network. To achieve the model management and adjust the communication resources according to the actual radio network conditions, the complementary SDN and ML algorithms for implementing rules configuration on network resources are compulsory [17].

B. IoT ARCHITECTURE BASED ON FL

As shown in Fig. 1, the FL-based architecture for big data IoT comprises three essential layers, the IoT user layer, network layer, and aggregation layer. These three layers communicate with each other; the model reliability of FL systems and

network QoS will rely upon each layer. The reliability and delay minimization are significantly considered crucial QoS aspects of each layer for E2E model QoS assurance. At the local device (IoT layer), the local client has different data features, different applications, and required various network resources from the network layer depending on its applications. Since the lightweight control protocols are suitable for real-time communications, the heterogeneous IoT applications obligate NS approaches to minimize the complicated system and user data, minimizing the operation time [18]. The MEC servers will place in fronthaul areas for local resources perspectives [19]. FL will take the opportunity for multiple aggregation server deployments based on the offered MEC servers.

Furthermore, the synchronizing servers will handle the mobility resources, which is essential for mobile applications and autonomous IoT applications required for resource handover to the next MEC server in charge. The failure at the handover processes always suffers the communication QoS in the mobile networks. Especially in the multiple aggregation servers, the clients attempt to select the optimal server for transferring its model update [20]. The failures of updated parameters occur during sending in the networks while insufficient radio resources and unavailable aggregation servers fail to respond, as illustrated in Fig. 2. The radio interfaces between client and radio remote head (RRH) require the optimal scheduling method based on specific service requirements. The optimal radio selections based on intelligent consideration enhance the model QoS. In the fronthaul communications, frame alignment processes, radio allocation, channel maintenance, and service mapping will produce communication delay and cause radio access failure in massive client attempts.

On the other hand, the network and aggregation layers require the awareness of the traffic behaviors for efficient traffic engineering methods and the apparent aggregation resources necessary to be classified based on the ML learning algorithms. The identified traffic types and existing aggregation resources will be the key parameters contributing to the centralized handling of the E2E communication of the FL systems. The entire computing times affect the FL system and its user QoS aspect; however, the FL model reliability mainly relies on communication delays. The joined network failures occur when the increased time delay and the local requests are over control.

The local cloud has been attached to the IoT gateways in the federated system to store the sensed data from various sensor devices. Local training is conducted by splitting the local dataset among each client into mini-batches of size  $db$  which are included into the set  $c = \{db_1, db_2, \dots, db_c\}$ . The local trained and updated models are sent to the edge servers for aggregation and modeled as follows.

$$w_{update}^c = \text{global}w_g \text{parameters} - \alpha \Delta \text{MSE}(w_{update}^c; db_c) \quad (1)$$

While  $w_{update}$  is the model parameter update from local clients  $\{w_{update}^1, w_{update}^2, \dots, w_{update}^c\}$ , local data mini-batches

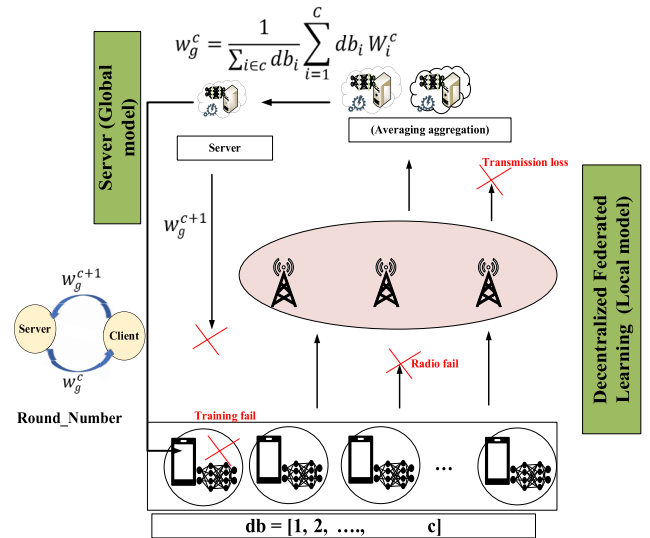


FIGURE 2. The failure of FL model communication (training failure, aggregate failure, network failure, update failure).

of total client  $c\{db_1, db_2, \dots, db_c\}$ , and MSE is the Mean Squared Error as the loss function for deep neural network (DNN). The local client transmits the updated model  $w_{update}^c$  over the wireless networks to aggregation server, which is supposed to locate in edge areas. The global server collects the up-to-date model  $w_{update}^c$  from variety aggregation servers for model summation. The global server will send the average global models to the local client. The global model can be modeled as follows.

$$w_G^t = \frac{1}{\sum_{i \in N} db_c} \sum_{i=1}^N db_c W_i^t \quad (2)$$

where the  $W_i^t$  is the updated model in each time  $t$ .  $W_G^{t+1}$  is the global update summation at time  $t + 1$ , the increasing number of round trip time (RTT) communications between local to the server will boost the global training accuracy.

### III. PROPOSED SYSTEM

This section presents the proposed optimal gateways selection and resource adjustment between client and service communication on the network interfaces.

#### A. RESOURCE ADJUSTMENT

Fig. 3 demonstrates the proposed network architecture, which is composed of CP and DP. Besides the monitoring and classification network statuses, the CP will manipulate the flow configuration. So, the forwarding flow will be made based on the outputs of the inspected network interfaces. The forwarding flow has to be updated whenever the networks and aggregation server statuses have changed. The CP and DP separate works that benefit from the CP resource offloading times, learning, and forwarding flows configuration. The updated periods are not influent real-time communication at the DP. At the same time, the MEC servers are integrated at

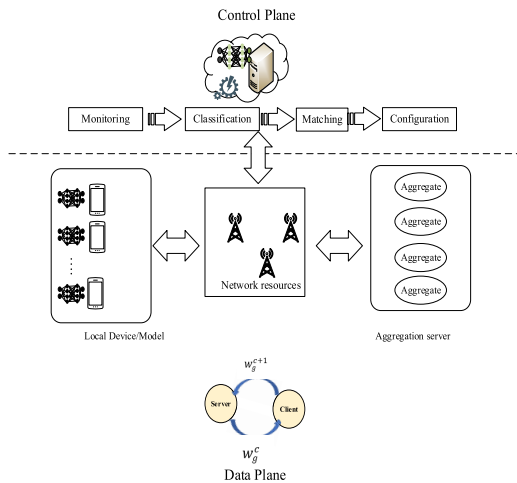


FIGURE 3. The proposed network architecture.

the centralized controller for global storage of the DP network data.

Moreover, MEC can be applied to store the massive forwarding flows which the SDN controller has configured. The flow updates are based on the inspected network resources. Therefore, if the incoming traffic is stable, the forwarding rules will also keep the stability without updating communication paths. The DP of the edge network consists of local client, network, and aggregation server, while the CP composes of four main modules, including monitoring, classification, resource adjustment, and configuration modules as explained in the following:

- 1) The controller server has a global monitoring module conscious of the oscillated network loading (local device/model, network resources, and aggregation servers interfaces). The information updates of local client  $w_{update}^c; db_c$  (e.g., update parameters, device statuses, etc.) and network resources with  $k$  gateways (the network interface between client to RRH and between RRH to aggregation server, and vice versa) are essential to be monitored. In addition, the changes are caching for performing centralized computing. The monitoring of the interface between RRH and the aggregation server will perceive real-time interface conditions. The client sends the updated models and the updated parameters  $db_c$  to the aggregator and aggregate server will return the updated global model  $w_g$  to the local, and uplink and downlink statuses are monitored. Additionally, the model sharing between client and server was adjusted, as shown in Algorithm 1.
- 2) The cached information will be utilized for the classification processes to differentiate the statuses of each entity of the DP. In this paper, the classification module was applied to classify the network resources in both interfaces, client to RRH and RRH to aggregation servers. Furthermore, the KNN was utilized to perform the classification of the individual RRH from both interfaces, as mentioned earlier. This module returns

### Algorithm 1 Pseudocode for Training Local, Global Model, and the Gateway Adjustment

**Require:**  $Data = \{x_1, x_2, \dots, x_n, y_m\}$  denotes  $n$  of general features as  $X$  with  $y_m$  target feature in the global dataset, and the network resource statuses classification  $c_k$  on  $k$  gateways.

**Ensure:** Optimal global learning model to classify into targets of  $Y = \{y_1, y_2, y_3 \dots, y_m\}$  associating with  $k$  gateways

- 1: Initialize the synchronous  $c, n_{epoch}, w_{update}$  parameters,  $\alpha, w_g$  for [Global Server]
- 2: [Local Client  $c$ ]
- 2: **for** each  $db_c$  in ( $Data$ ) **do**
- 3: Input  $\alpha$  hyperparameter,  $c_k$  network resource statuses on  $k$  gateways,  $w_g$ , and  $w_{update}$  parameters
- 4: Initialize empty  $gw_{replay}$  list for local
- 5: Define class **lowDNN**(self,  $\alpha, w_g$ ):
- 6: **for** each client in  $c$  **do**
- 7:  $w_{update}^c = \text{global } w_g \text{ parameters} - \alpha \Delta \text{MSE}(w_{update}^c; db_c)$
- 8: **end for**
- 9: **for** each epoch in  $\text{range}(n_{epoch})$  **do**
- 10: **if** random (0,  $k$ ) gateway has bad condition  $c_k$  **then**
- 11: Restrict on model updating  $w_{update}$  on **lowDNN**()
- 12: Append the gateway to  $gw_{replay}$  and suggest gateway not in  $gw_{replay}$  list
- 13: **pass**
- 14: **else**
- 15: Increase model updating  $w_{update}$  to maximum  $w_{update}^c$  on **lowDNN**()
- 16: Update selected gateway resource condition  $c_k$  for local experience
- 17: **end if**
- 18: **end for**
- 19: **end for**
- 20: [Global Server]
- 21: Define class **highDNN**():
- 22: **for** each epoch in  $\text{range}(n_{epoch})$  **do**
- 23: Import **lowDNN**() from  $c$  clients via  $k$  gateways
- 24: Aggregating the model  $w_g$  for next epoch by using FedAvg algorithm [21]
- 25: Update gateway resource condition  $c_k$  after received
- 26: **for** each gateway in  $\text{range}(k)$  **do**
- 27: Update  $gw_{replay}$  list for local by modifying resource statuses
- 28: Update selected gateway resource condition  $c_k$  for global experiences
- 29: **end for**
- 30: **end for**

the inspected results of each interface representing the network conditions. The inspected outcomes are

**Algorithm 2** Network Path Selections

---

```

[Input]
Network observations
Caching metrics pools
1  While communication is not terminated, do
2    Compare network observed conditions
3    If  $\rightarrow$  Network stability = true, then
4      Selection of the cached metrics
5      Flow configuration
6    Else  $\rightarrow$  Network fluctuation = true, then
7      Optimal path prediction
8      Flow configuration
9    End if
10   Update caching metrics
11  End while

```

---

essential for SDN controllers regarding local devices gateways and optimal aggregation server selections.

- 3) The SDN controller selects the optimal gateway based on inspected metrics. The gateway with high loading metrics will be considered the high-risk gateway that can cause network failure. The reliability gateways and aggregation server will be considered on the available resource with the possibility of providing low latency. According to the adjustment policy, the matching module will guarantee model communication based on the throttling process as depicted in Algorithm 2.

Additionally, the client will be restricted from sending updated information  $w_{update}^c$  and its mini-batches. In the local clients [Local Client  $c$ ], the controller restricts the sending update of the model parameter concerning the network resource statuses. The throttling method reduces the limited network gateway and suggests an optimal gateway with sufficient serving resources for carrying the model parameters. Furthermore, a network gateway will be configured to increase the delivery ratio of local models (reduce the restriction) whenever the gateway statuses are not under the limited threshold. The classified network resources presented particular gateway statuses and distinguished each gateway's capability, which is an effective utilization for resource configuration from client to server and vice versa.

## IV. SYSTEM EVALUATION

### A. DATA AND SIMULATION ENVIRONMENTS

The utilizing data for model evaluation comprised two categories, E2E FL model reliability evaluation and network QoS aspects. The opened dataset `tff.simulation.datasets.emnist.load_data()` [22] are loaded from the federated EMNIST. The EMNIST dataset was sliced to meet the number of clients for testing by using the google platform. Each client has each slice of the dataset (individual dataset) and training model, and the aggregate server is utilized the FedAvg function offered by TensorFlow Federated [22]. And network dataset was generated by utilizing the python software program. KNN was used to perform classification processes for the

generated network gateways resources. The reliability of the FL model was evaluated by utilizing FedAvg of TensorFlow by google. And the network's QoS evaluation system was simulated by two different scenarios, including the conventional scheme (random and equal-cost based methods) and proposed network resources adjustment. ML algorithms' training and testing model were conducted using the opened ML library, sci-kit learn [23]. And the real-time network simulation was conducted using the discrete network simulation version 3 (NS3) [24]. MEC gateway configured default priority first in first out (pFIFO) queue to buffer the incoming traffic. The network resources are generated according to 4 RRH, 4 MEC gateways, 40 user devices, and 200 seconds of simulation periods.

### B. RESULTS AND DISCUSSION

In this section, the experimental results and discussion are provided. The FL model reliability evaluations are compared based on the vital QoS in terms of E2E packet lost ratio, E2E communication reliability, average communication throughput of the systems, and E2E communication jitters. The simulated results are compared with the equal-cost, random, and proposed algorithm. Due to the real-world network environments, traffic handling is commonly based on random handling. The incoming traffic can be selected as the serving gateway aimlessly. So, the selection of inappropriate serving gateways will be made. The optimal approaches for appropriate serving gateways selection are required to increase the E2E network fault tolerance for enhancing communication reliability. In real-world FL model communications, massive local devices in the networks consist of heterogeneous serving gateways with different capacities, including the server busy, delay at each server, serving bandwidth, and other conditions, based on the apparent environments. Each serving gateway is required to handle the incoming traffic that is adjusted with its capacity. Moreover, suppose the incoming requests exceed the available serving capacity. In that case, the incoming requests can be queued for an extended period, and the local model will be dropped during the waiting times are expired.

Four conditional simulations of the federated model experiments are conducted to evaluate the system performances. C1, C2, and C3 present the 50%, 30%, and 10% loss of aggregation from clients, respectively, in ineffective network resources handling. C4 denotes the optimal network adjustment in the proposed model. C4 also presents the selected gateway with appropriate network resources. Fig. 4 illustrates the summation of errors in four different conditions, namely Loss-C1, Loss-C2, Loss-C3, and Loss-C4. The proposed model presented in Loss-C4 by minimizing the loss function's metric through an efficient optimization method with adequate aggregation capabilities, which outputs the average reduction of 1.0723, 0.7243, and 0.4315 loss values compared to Loss-C1, Loss-C2, and Loss-C3, respectively, within 99 round communications for global model averaging and aggregation. Fig. 5 presents the precision accuracy of

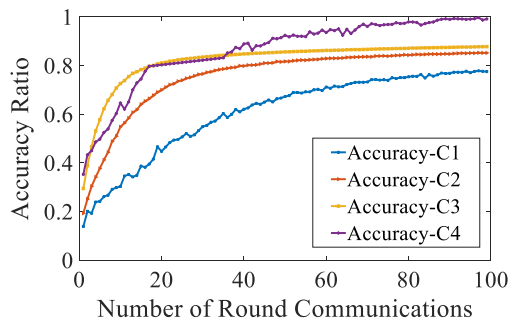


FIGURE 4. The model lost comparisons.

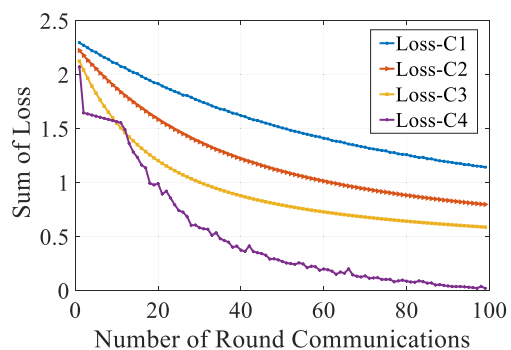


FIGURE 5. The model accuracy at each condition comparison.

the proposed model on Accuracy-C4 and other three experimental scenarios, including Accuracy-C1, Accuracy-C2, and Accuracy-C3, on conventional gateway selections. Accuracy evaluation is a significant metric for performance analytics on ultra-reliable communication requirements. The proposed scheme reached an accuracy of 99.6707%, which was 21.8689%, 14.4414%, and 11.9040%, 17.6995% higher than Accuracy-C1, Accuracy-C2, and Accuracy-C3, respectively. By determining the congestion circumstance critically and selecting the gateway with efficient communication and computation resources, the proposed scheme can significantly advance promising ultra-reliable low-latency communication (URLLC) requirements. In each round of communication, a loss is measured through the backpropagation process with mean squared error (MSE).

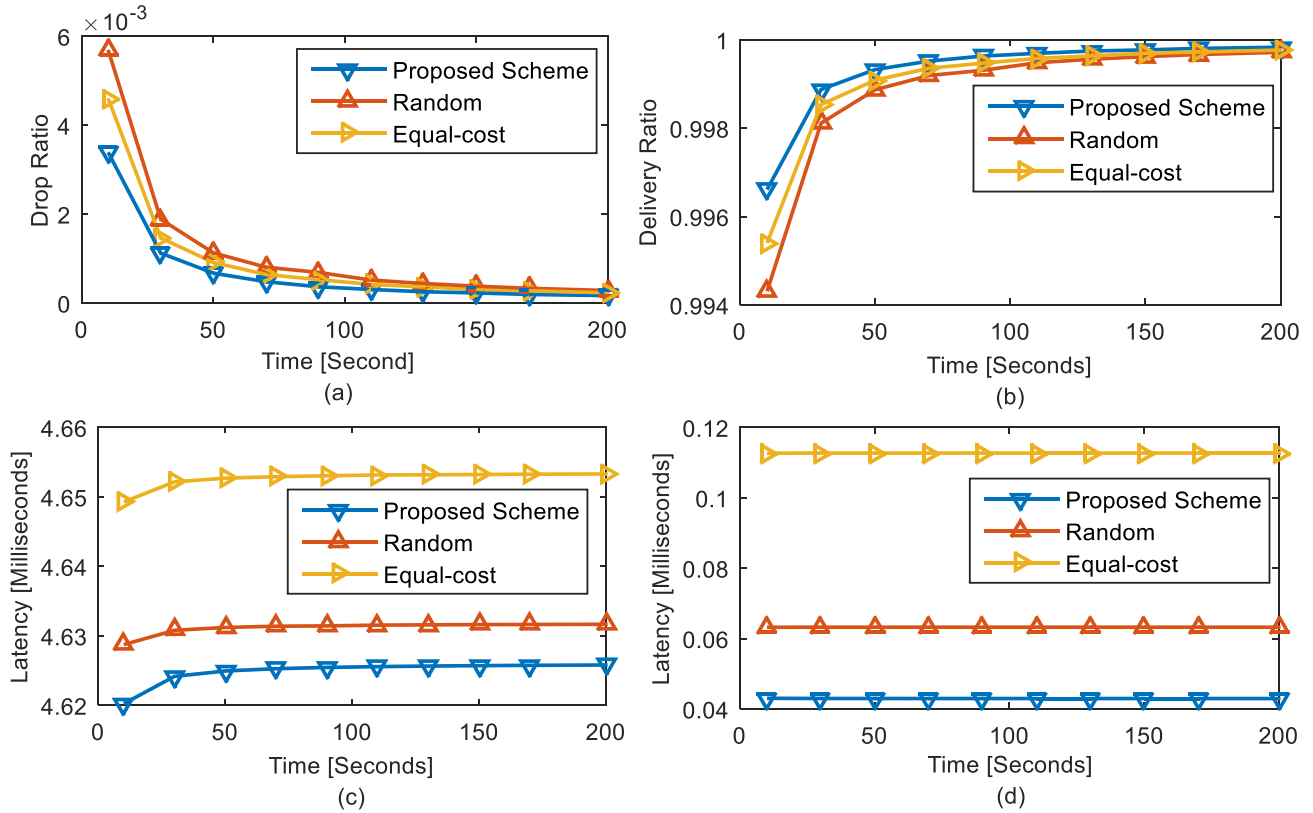
Fig. 6(a) compares the total packet drop ratio with different schemes between the random handling method, equal-cost load balancing, and the proposed algorithm. The proposed scheme shows a remarkable outperformance compared to the two conventional approaches. Based on the graphs, the proposed scheme achieved the minimum packet drop ratio during the communication. Therefore, the proposed scheme representation of the model exchanges between the local and global servers (E2E communication) can be adjusted and reliable. For the conventional schemes, the equal-cost technique has lower packet drop ratios than the random method. The equal-cost handling method provides the equal-cost load balancing for traffic handling. The incoming traffic will be divided equally for each network interface

(e.g., RRH and aggregation server gateways), which gets the exact incoming requests to be served. However, this method can be suffered from inadequate gateway capacity with limited resources, which cannot handle the massive requests. Then, the user requests will be discarded during under-control periods.

Additionally, in a traditional network, the servers with higher capacity will receive only a tiny amount of the requests, while the remaining serving capacity will be unutilized. The user requests possible select the unreliable gateway with poor or high loading metrics for the random selection method. When a vast amount of local clients send their update models to the lowest capacity gateways, the packet will be waiting in a queue that can be discarded when the waiting period has expired. The poor optimal gateway selections will increase the network failure ratio. Consequently, conventional schemes are incompatible for future FL environments, especially the FL systems for time constraint applications, including ultra-high reliability communication and ultra-low latency systems. The proposed scheme provides the optimal distributed edge cloud gateways selection based on the integration of ML algorithms with SDN controllers. Thus, the recommendation based on the ML approach significantly offered mechanisms to balance the network gateways in fluctuation communication systems.

The proposed scheme outperformed the conventional approaches in E2E communication reliability, as shown in Fig. 6(b). The graphs show that the random and equal-cost-based handling methods showed poor transmission reliability, while the proposed scheme provided the highest E2E reliable communication. IoT networks required efficient traffic handling to improve the E2E QoS, especially communication reliability, which evolves with ubiquitous and other lightweight sensor devices with limitations of capacity. Furthermore, lightweight IoT systems communicate over the UDP communication protocol, which has poor communication reliability. Hence, the proposed schemes can be meaningful for lightweight IoT systems in enhancing communication reliability.

Moreover, in the FL environments, the primary concern is communication reliability since the global server aggregates the average model based on the update parameters from the local devices. In future edge network infrastructure, the aggregation servers will be distributed in various RAN areas and significantly obligated the reliable model with global monitoring of the statuses for massive aggregation servers. The URLLC is required to enhance the QoS and QoE of real-time IoT communications. Regarding the drop and delivery ratios demonstration in Figures 6(a) and 6(b), the maximum number of received model updates and parameters from local clients occurred in the proposed scheme. Since the proposed scheme enhances communication reliability, the aggregation server can collect sufficient models from various clients to establish a reliable global model with satisfying accuracy. Based on the given graphs in Fig. 4, the summation of training loss in each condition reflects



**FIGURE 6.** The comparison of E2E communications QoS between proposed gateway resource adjustment and conventional schemes in terms of packet drop ratio, delivery ratio, delay, and jitters are shown in (a), (b), (c), and (d), respectively. The proposed scheme was applied both at radio and aggregation gateways.

the model drop ratio expressed in Fig. 6(a). Whenever the local model is discarded in the communication, the insufficient accurate model will occur at aggregation servers. At the same time, Fig. 6(b) reflects model accuracy expressed in Fig. 5. The maximum E2E reliability of model communication between client and server brings the maximum model accuracy and representing the model reliability. The random and equal-cost-based gateways selection have the lowest and second-lowest communication reliability (minimum numbers of the receiving models, parameters, respectively), representing the lower model accuracy in particular network conditions.

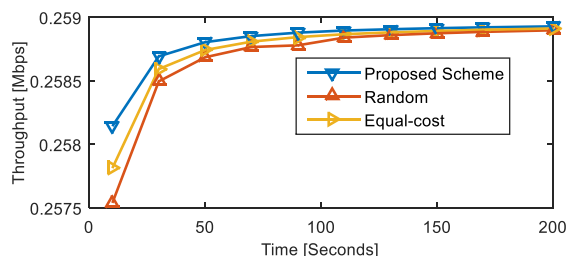
In future FL communication systems, real-time IoT users will suffer from the loading delay during MAC scheduling in the radio networks. Consequently, to guarantee E2E QoS, a real-time IoT network is required to handle both congestions at radio gateways and MEC gateways. The proposed scheme systematically supervises the massive traffic to match the existing communication resource (RRH gateways and MEC gateways). Fig. 6(c) illustrates that the proposed method surpasses the random and equal-cost-based approaches in E2E communication delay. However, IoT communicates over 5G communications; the mission-critical IoT applications will suffer from highly queuing periods at network interfaces that raise the network loading

metrics and lead to failure transmission for time-critical applications.

Furthermore, the gateway with high loading metrics is highly vulnerable to fail the model exchange between client and servers, especially in lightweight WSN devices, which is a kind of resource constraint device that shares information over unreliable transmission protocols. Hence, the ultimate traffic adjustment for E2E networks and will be significant for future HetIoT applications. Regarding the proposed scheme consists the tiniest drop ratio, the URLLC for lightweight systems is performed. The E2E communication latency can be lessened when the poor conditions at bottleneck areas can be handled. Due to the limitation of radio resources and edge gateways, the stability of the RAN environments will be insufficient for massive IoT traffic. The proposed scheme solved these significant issues, which relied on ML algorithms for classifying DP resources and recommending optimal gateway selection for serving the incoming traffic according to identified network loading metrics. Consequently, the E2E optimal resource utilization overcomes the communication latency for both communication delay and jitter, as elaborated in Fig. 6(c) and Fig. 6(d), respectively.

Typically, the communication throughput performances rely on the communication delay. The average communication throughput is shown in Fig. 7, and the proposed scheme





**FIGURE 7. The comparison of Average communication throughput comparison between proposed and conventional schemes.**

consists of higher potential in forwarding client models to global servers. The proposed method offers a remarkable outperformance compared to the conventional approaches. Due to the optimal steering with the appropriate MEC gateway statuses, the fluctuation of improper resource computing was reduced. The conventional schemes were handling the communication based on splitting the incoming requests equally for each network gateway. As the illustrated results, the proposed systems suited the adequate balance between IoT requests and serving entities' available resources; the proposed scheme achieved QoS in E2E communications reliability, delay, jitter, and throughput.

## V. CONCLUSION

This paper presented an intelligent network resource adjustment by integrating the SDN controller with a lightweight ML algorithm to enhance E2E FL communication reliability for massive real-time IoT applications in future edge cloud servers. The proposed schemes deliver systematic resource adjustment and outperform the conventional approaches in terms of crucial QoS aspects. According to the communication gateways, the local client and its traffic will be controlled to meet the network conditions. Moreover, the proposed scheme improves the main critical factors of user QoS in reducing E2E communication delays and jitters, increasing communication reliability, and enhancing communication throughput; therefore, the stability of the communication systems will be significantly improved. These mentioned vital outcomes implied the reliability of the E2E model in the massive FL clients. The paper is dedicated to the lightweight methods which meet the perspective of fronthaul MEC network infrastructure and real-time IoT applications. The proposed scheme is mainly suitable for lightweight computation systems that require short periods of computation and resource-constrained systems. Future research will integrate the autonomous SDN rules implementation for software-defined routing approaches based on the network loading prediction to enhance model reliability for federated mobility systems in big data-sharing networks.

## REFERENCES

- [1] "5G end-to-end architecture framework (phase-3)," NGMN Alliance, White Paper, 2020. [Online]. Available: [https://www.ngmn.org/wp-content/uploads/201117-NGMN\\_E2EArchFramework\\_v4.31.pdf](https://www.ngmn.org/wp-content/uploads/201117-NGMN_E2EArchFramework_v4.31.pdf)
- [2] M. M. Alqarni, A. Cherif, and E. Alkayal, "A survey of computational offloading in cloud/edge-based architectures: Strategies, optimization models and challenges," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 3, pp. 952–973, 2021, doi: [10.3837/tiis.2021.03.008](https://doi.org/10.3837/tiis.2021.03.008).

- [3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.
- [4] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.
- [5] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.
- [6] P. Tam, S. Math, and S. Kim, "Intelligent massive traffic handling scheme in 5G bottleneck backhaul networks," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 3, pp. 874–890, 2021, doi: [10.3837/tiis.2021.03.004](https://doi.org/10.3837/tiis.2021.03.004).
- [7] A. H. Sodhro, S. Pirbhulal, and V. H. C. de Albuquerque, "Artificial intelligence-driven mechanism for edge computing-based industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4235–4243, Jul. 2019.
- [8] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu, "EdgeFed: Optimized federated learning based on edge computing," *IEEE Access*, vol. 8, pp. 209191–209198, 2020.
- [9] L. U. Khan, M. Alsenwi, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "Resource optimized federated learning-enabled cognitive Internet of Things for smart industries," *IEEE Access*, vol. 8, pp. 168854–168864, 2020.
- [10] X. Lu, Y. Liao, P. Lio, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48970–48981, 2020.
- [11] P. Shantharama, A. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione, "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing," *IEEE Access*, vol. 6, pp. 57545–57561, 2018.
- [12] W. Saeed, Z. Ahmad, A. I. Jehangiri, N. Mohamed, and A. I. Umar, "A fault tolerant data management scheme for healthcare Internet of Things in fog computing," *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 1, pp. 35–57, 2021, doi: [10.3837/tiis.2021.01.003](https://doi.org/10.3837/tiis.2021.01.003).
- [13] J. H. Kwak, "A study on the evolution of post-smartphone technologies in the 5G technology environment," *KSII Trans. Internet Inf. Syst.*, vol. 14, no. 4, pp. 1757–1772, 2020, doi: [10.3837/tiis.2020.04.019](https://doi.org/10.3837/tiis.2020.04.019).
- [14] G. Harerimana, B. Jang, J. W. Kim, and H. K. Park, "Health big data analytics: A technology survey," *IEEE Access*, vol. 6, pp. 65661–65678, 2018, doi: [10.1109/ACCESS.2018.2878254](https://doi.org/10.1109/ACCESS.2018.2878254).
- [15] O. A. Wahab, A. Mourad, H. Otok, and T. Taleb, "Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1342–1397, 2nd Quart., 2021.
- [16] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.
- [17] E. Kim and S. Kim, "An efficient software defined data transmission scheme based on mobile edge computing for the massive IoT environment," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 2, pp. 974–987, 2018.
- [18] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, 2020.
- [19] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1227–1240, Jun. 2020.
- [20] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.
- [21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54, 2017, pp. 1273–1282.
- [22] M. Abadi et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [23] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
- [24] G. F. Riley and T. R. Henderson, "The NS-3 network simulator," in *Modeling and Tools for Network Simulation*, K. Wehrle, M. Güneş, and J. Gross, Eds. Berlin, Germany: Springer, 2010.



**SA MATH** received the B.E. degree from the Department of Telecommunication and Electronic Engineering, Royal University of Phnom Penh, Phnom Penh, Cambodia, in 2018. He is currently pursuing the Ph.D. degree with the Department of Software Convergence, Soonchunhyang University, Asan, Republic of Korea. His research interests include 5G distributed, core networking, the Internet of Things, quality of service, software defined mobile edge computing, and machine learning.



**PROHIM TAM** received the B.S. degree from the Department of Management of Information Systems, Paragon International University, Cambodia, in 2019. He is currently pursuing the Ph.D. degree with the Department of Software Convergence, Soonchunhyang University, Republic of Korea. His research interests include future access networks, software-defined networking, artificial intelligence, bigdata transmission, mobile edge computing, and the Internet of Things.



**SEOKHOON KIM** (Member, IEEE) received the B.E. and Ph.D. degrees in computer engineering from Kyunghee University, Republic of Korea, in 2000 and 2004, respectively. From 2004 to 2006, he was with IPOne, Inc., Seoul, Republic of Korea, where he led various projects as a Research Engineer. From 2006 to 2009, he was a Research Engineer at Neowave, Inc., Anyang, Republic of Korea, where he developed Mobile WiMAX (IEEE 802.16) devices. He was an Assistant Professor with the Department of Mobile Communications Engineering, Changshin University, Changwon, Republic of Korea. Since March 2016, he has been with the Department of Computer Software Engineering, Soonchunhyang University, Asan, Republic of Korea, where he is currently an Assistant Professor. His research interests include cloud and mobile edge computing, the Internet of Things, heterogenous Internet of Things, software-defined networking and network function virtualization, mobile system and communications, and machine learning based on bigdata.

...