Communication Efficient Mobile Data Collection for IoT Edge using Federated Learning

[1*]Mario Infant Raj, [2*]Dr. K. Kamali, [3*]Dr. R. Manikandan

[1]Research Scholar , Annamalai University. Chidambaram, Tamilnadu. (corresponding author)
marioinfantrajphd@gmail.com

[2]Assistant Professor, Dept of Computer science,Annamalai University. Chidambaram,, Tamilnadu.
Kamaliaucse2006@gmail.com

[3]Associate Professor, Dept of Computer science,Annamalai University. Chidambaram, Tamilnadu.
rmkmanikandan1111@gmail.com

Abstract

Internet of Things (IoT) has emerged in recent years as a result of substantial advances in computer and communication technology. A new paradigm for networked computing is federated learning (FL). However, utilizing FL in client-edge-computing will result in costly communication overheads, with usual resources used for ordinary communication. The edge server has more effective interactions with the IoT devices as well as with the cloud server. This paper proposes a IoT-edge-computing hierarchical FL system to combine their benefits, backed by a HierFedAI algorithm that enables several edge servers to carry out incomplete method collections. Earlier method training as well as improved communication-computation trade-offs are possible with this method. The proposed approach beats existing algorithms in IoT edge network setup, according to evaluation results, which also show greater precision, recall, and F1 score.

Keywords: Internet of Things (IoT), Federated learning (FL), communication overhead, Client edge cloud, Artificial Intelligence (AI).

1. Introduction

IoT has emerged in recent years as a result of substantial advances in computer and communication technology [1].. Lower energy costs, better use of natural resources, safer cities, and a healthier environment may all be achieved via the usage of IoT solutions.[1]. Artificial Intelligence (AI) and IoT have gained the attention of academics because of their fast increase [2]. IoT networks may save a lot of power by using mobile data collecting. When using a MDC, the biggest problem is determining and arranging the MDC’s course to gather data from nodes. Static techniques of obtaining mobile data only identify a solution to a problem with predetermined variables [3][4].

A broad review is introduced specifying different utilizations of FL in remote organizations and featuring their difficulties and constraints. FL erose communication overhead issues]. In FL, preparing information driven AI models is a demonstration of coordinated effort between numerous clients without requiring the information to be brought to a main issue, subsequently lightening communication and storage costs and giving an extraordinary level of client level security. FL paradigm, specifically provides distributed learning services. The connection between two organization measurements and the FL execution in a various leveled united learning framework is investigated [5].

One drawback of edge-based FL is the constrained client access that each server has, which inevitably reduces training performance. The main objectives of this work include:

- To considerably cut down on the expensive connection to the cloud and augment it with effective client-edge informs, leading to a considerable decrease in local iterations as well as runtime.
- To exceed edge-based FL in method training as more data becomes accessible to the cloud server.

2. Literature Review

This section reviews some of the recent methodologies and frameworks related to FL in networks.

Zhou et al. [8] highlighted a two-layer FL model in this study to take advantage of the distributed end-edge-cloud architecture that is characteristic of a 6G environment, achieve highly effective and accurate training, protect privacy, and reduce communication costs all at the same time. Sirohi et al. [9] discuss the restricted stockpiling and execution bottleneck issues on the unified servers through the implementation of different ML and Deep Learning (DL)-based models. With distributed computing power along with unlimited information, it is possible to use these models to their full potential.

Renda et al. [11] provide a comprehensive description of pertinent 6G use cases, with a effort on environments where V2X depicts a structure to assess the proposed method, including internet preparation, in light of genuine information from live organizations. As a method for realizing seamless obtainability of lightweight, decentralized as well as communication-efficient intelligence, the FL of Explainable Artificial Intelligence (XAI) models is anticipated to offer advantages. Kalapaaking et al. [12] make a proposal for a FL framework based on Convolutional Neural Networks (CNN) that combines Encrypted Inference techniques with Secure Multi-Party Computation (SMPC)-based combination of 6G and IoMT. This work takes into account a number of hospitals that have clusters of edge and mixed Internet Of Medical Things (IoMT) locally trained models encrypt that devices.

3. Proposed Methodology

In this part, the research first discusses the overall learning issue in FL. The main architectural differences between the edge server along with cloud based FL systems are in communication along with client participation numbers. As a result, the two systems are handled as one standard, two-layer FL system. In this study, a suggested FL system called HierFedAI is presented for the CEC hierarchical FL system.

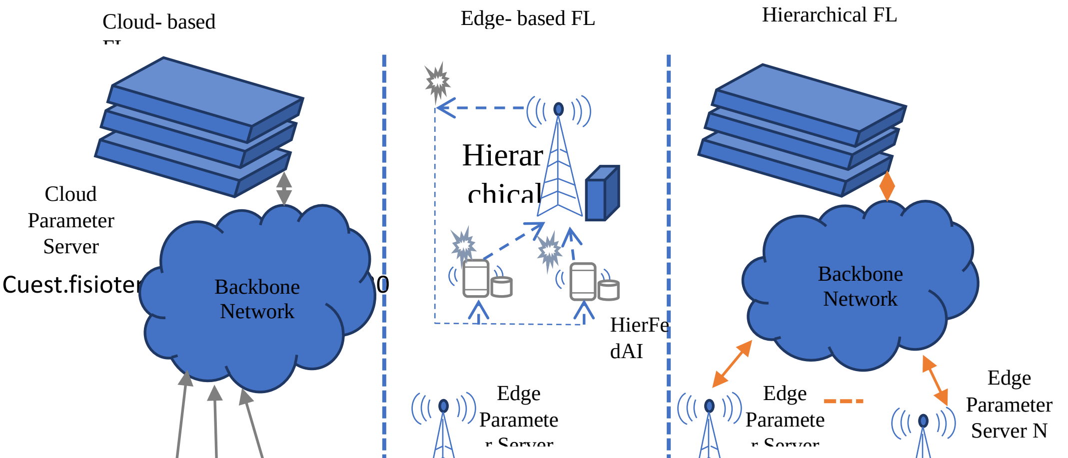


Figure 1. System model of FL

3.1 Learning Problem

The topic of this study is supervised FL. A deep learning CNN model may be trained using the raw data $D = \{d_i\}_{i=1}^{n}$. Assuming that $d_i = (x_i, y_i)$, all local dataset d_i is made up of the raw data x_i and the matching label y_i. The CNN model is completely parametrized by the actual vector ω. A deep learning-based CNN model, $M = h(x, \omega)$, is then trained on the data, D. Locally, this hypothesis $h(x, \omega)$ has been trained as in (1).

$$h(x, \omega) = FC\big(Pool\big(Conv(x_i, \omega_{conv})\omega_{pool}\big)\omega_{FC}\big) \quad (1)$$

Here $Conv$ represents the convolution layer, $Pool$ represents the pooling layer, FC represents the fully connected layer, respectively. A SoftMax classifier predicts the outcome of $h(x, \omega)$ as follows as in (2):

$$y^{predict} = Softmax\big(h(x, \omega)\big) \quad (2)$$

Based on the training dataset, the training procedure aims to minimize the empirical loss $L(\omega)$ is defined as follows in (3):

$$L(\omega) = \frac{1}{|D|}\sum_{i=1}^{|D|} f(x_i, y_i, \omega) = \frac{1}{|D|}\sum_{i=1}^{|D|} f_i(\omega)$$

(3)

The loss function might be convex and is dependent on the DL based CNN model. Gradient descent is frequently used to address difficult learning problems. The method parameters are updated as follows as in (4):

$$\omega(k) = \omega(k-1) - \varphi\nabla L\big(\omega(k-1)\big) \quad (4)$$

Here k denotes the update step index as well as φ denotes the gradient descent step size. The dataset is spread over M clients in FL as follows: $\{D_i\}_{i=1}^{M}$. Therefore, the parameter server is unable to directly access these datasets. As a result, $L(\omega)$, also known as the global loss in Eq. (3), cannot be estimated directly but rather can only be computed as a weighted mean of the loss functions $L_i(\omega)$, on datasets D_i. $L(\omega)$ and $L_i(\omega)$ are specifically equated by (5) and (6):

$$L(\omega) = \frac{\sum_{i=1}^{M} |D_i| L_i(\omega)}{|D|} \tag{5}$$

$$L_i(\omega) = \frac{\sum_{i \epsilon D_i} f_i(\omega)}{|D_i|} \tag{6}$$

3.2 Two-Layer FL

One centralized server and M clients make up the standard two-layer FL system. The method communicates along with aggregates for each k step of gradient descent on each user to minimize the communication costs. The cycle continues until the model achieves the target accuracy or until the finite resources, such as the time budget or communication budget, are exhausted. If $\omega_i(k)$ is the model's input on the ith user, then $\omega_i(k)$ is developed as follows as in (7):

$$\omega_i(k) = \begin{cases} \omega_i(k-1) - \varphi \nabla L_i\big(\omega_i(k-1)\big) & where\ k \neq 0 \\ \frac{\sum_{i=1}^{M} |D_i| \left[\omega_i(k-1) - \varphi \nabla L_i(\omega_i(k-1))\right]}{|D|} & where\ k = 0 \end{cases} \tag{7}$$

3.3 CEC Hierarchical FL

The model aggregation process might be seen as a mechanism for clients to communicate with one another. As a result, cloud parameter server aggregation can include a large number of clients, but communication costs are considerable. Combination at the edge server only includes a few users with significantly lower connection costs. The study takes into account a HierFedAI framework with a cloud server, E edge servers indexed by e, unique user sets $\{C^e\}_{e=1}^{E}$, and M users indexed by i and e, with datasets $\{D_i^e\}_{i=1}^{M}$ to aggregate their benefits. Indicate D^e as the combined dataset underneath edge e. The models from each client's edge server are combined.

All edge server combines the models of its clients once all client receives k_1 local update. The cloud server then combines all of the edge servers' method every k_2 edge model aggregations; thus, communication with the cloud occurs every one and a half local updates. Define k_1 as the count of local updates done, which is represented as $k_1 k_2$, and $w_i^e(k)$ as the local method metrics following the kth local update. Algorithm 1 follows with a presentation of the HierFedAI algorithm's specifics.

The algorithm procedure of HierFedAI algorithm is given below

Algorithm 1: HierFedAI algorithm

1. Procedure HierFedAI
2. Initialize all client parameters ω_0
3. For $k = 1,2,,,,K$ do
4. For each parallel client $i = 1,2,,,,M$ do
5. $w_i^e(k) \leftarrow \omega_i^e(k-1) - \varphi\nabla L_i\big(\omega_i^e(k-1)\big)$
6. End for
7. If $k \mid k_1 = 0$ then
8. For every parallel edge $e = 1,2,,,,E$ do
9. $\omega^e(k) \leftarrow Edgeaggregation(\{w_i^e(k)\}_{i\in C^e})$
10. If $k \mid k_1 k_2 \neq 0$ then
11. For each parallel client $i \in C^e$ do
12. $w_i^e(k) \leftarrow \omega^e(k)$
13. End for
14. End if
15. End for
16. End if
17. If $k \mid k_1 k_2 = 0$ then
18. $\omega(k) \leftarrow Cloudaggregation(\{\omega^e(k)\}_{e=1}^{E})$

19. For every parallel client i= $1,2,,,,M$ do

20. $w_i^e(k) \leftarrow \omega(k)$

21. End for

22. End if

23. End for

24. End procedure

25. Function $Edgeaggregation(e,\{w_i^e(k)\}_{i\in C^e})$

26. $\omega^e(k) \leftarrow \frac{\sum_{i\in C^e}|D_i^e|\omega_i^e(k)}{|D^e|}$

27. return $\omega^e(k)$

28. End function

29. Function $Cloudaggregation(\{\omega^e(k)\}_{e=1}^{E})$

30. $\omega(k) \leftarrow \frac{\sum_{e=1}^{E}|D_i^e|\omega^e(k)}{|D|}$

31. return $\omega(k)$

32. End function

4. Experimental Results

The simulation results for HierFedAI are presented in this part to support the predictions and highlight its benefits. The benefit over the FL system in terms of method correctness is evident in Fig. 2, as demonstrated. Therefore, the comparison with the FL system based on the cloud is the main emphasis of this study.

4.1 Settings

This study assumes that all edge server allows the equal count of clients with the equal amount of training information in this FL system with 5 edge servers, a cloud server along with 50 clients. The standard datasets MNIST and BelgiumTSC are utilized for the ML tasks, which include classification tasks.

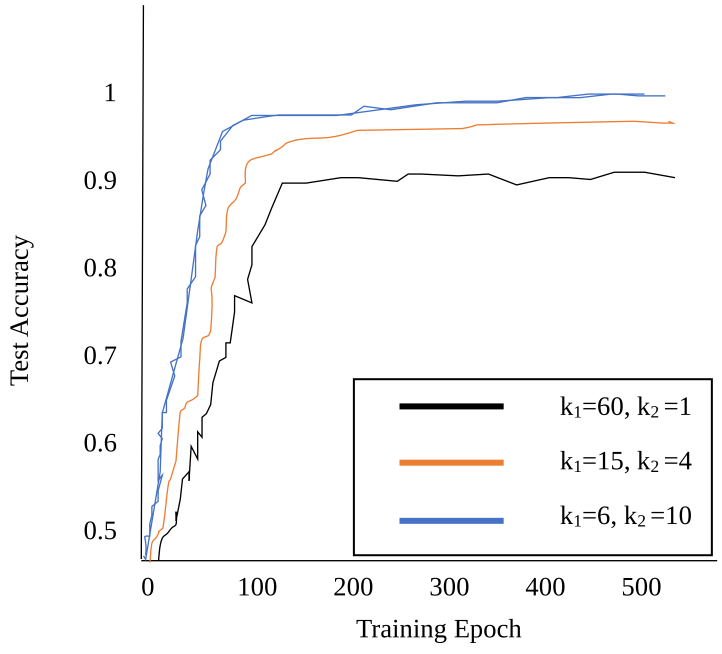
This study uses the CNN with 21840 trainable parameters for the 10-class hand-written digit classification dataset from MNIST and BelgiumTSC. It applies Stochastic Gradient Descent (SGD) with group size 20 along with an original learning rate of 0.01 with a decay rate of 0.995 with each epoch for training. The employed CNN consists of 3 convolutional blocks, 3 activation blocks, and 3 batch normalization blocks. Mini-batch SGD is also used for the local computation of the training, with a batch size of 20, an initial learning rate of 0.1, along with an exponential decrease in learning rate of 0.992 per epoch. This study does not employ momentum in the experiments in order to be consistent with the analyses.

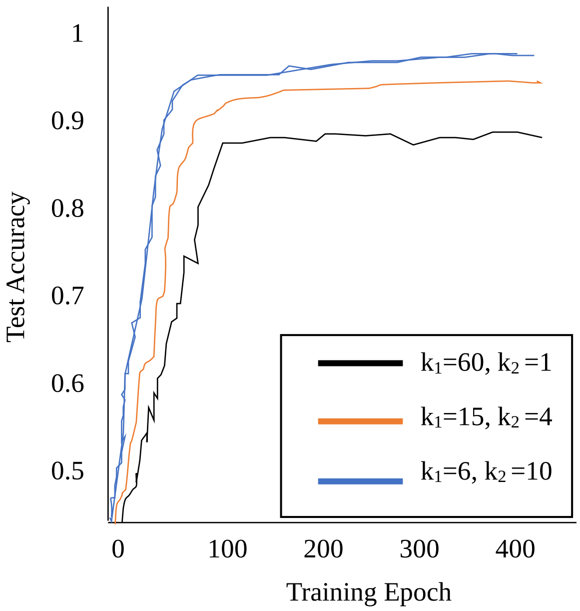
Data pre-processing, which entails the following processes, is done to modify the training as well as testing information to meet the 6G environment in order to adequately assess the detection performance for 6G. Delete the records with incorrect formatting and missing features. To replicate the 6G-supported IoV environment, make the dataset 10 times larger than it was originally. Divide the entire expanded dataset into separate portions and allocate them to FL situations.

4.2 Evaluation Metrics

The parameter configurations for 6G include the peak data rate per device as 1 Tbps, the end-to-end latency as 1 ms, all-out spectral efficiency as 100bps/Hz and maximum frequency as 10 THz. True positive (TP), True negative (TN), False positive (FP) as well as False negative (FN) evaluation indicators are utilized in performance comparisons. The three commonly used metrics are: precision, recall, and F1 score.

The model's capability increases with increased recall and precision values. The model performs more robustly with a higher F1 score. Also, the model accuracy is estimated for the both datasets and is tabulated in figure 3. The finding to be confirmed is that, while the communication incidence with the cloud is fixed as 60 local iterations (i.e., $k_1 k_2$ is fixed), faster training may be accomplished by communicating with the edge more frequently. The value of k_1 is changed. By lowering k_1 for both types of data distribution, the requisite accuracy may be attained with small training epochs, requiring smaller local calculations on the devices. The training process won't be slowed down by cutting back on communication with the cloud server while the datasets between edges along with the communication frequency with the edge server are fixed.




1
0.9
0.8
0.7
0.6
0.5
Test Accuracy
0
100
200
300
400
Training Epoch
k1=60, k2 =1
k1=15, k2 =4
k1=6, k2 =10

(b) B

Figure 3. Testing Accuracy for (a) MNIST dataset (b) BelgiumTSC dataset

In Table 2, the outcomes of other factors are given and contrasted. According to Table 2, it is clear that the proposed HierFedAI produces the best F1 scores, at 0.97 with the MNIST dataset and 0.98 with the Belgium TSC dataset. This suggests that the suggested approach may produce greater accuracy.

Table 2. Performance Comparison

Dataset	Algorithm	Precision	Recall	F1 score
MNIST	CNN	0.88	0.82	0.85
	RegionNet	0.89	0.91	0.90
	Random Forest (RF)	0.74	0.85	0.79
	TFL-CNN	0.90	0.96	0.93
	Proposed	0.98	0.98	0.97
BelgiumTSC	CNN	0.86	0.81	0.84
	RegionNet	0.87	0.90	0.88
	Random Forest (RF)	0.75	0.83	0.76
	TFL-CNN	0.89	0.95	0.92
	Proposed	0.97	0.96	0.98

This research presents the outcomes of every client on average based on the provided data sets to estimate the communication expenses throughout the FL process. The degree of communication optimization will vary significantly depending on the size of the distinct data sets. Since the data is uploaded simultaneously with each FL-CNN iteration, the communication cost during the learning process is minimized to a greater extent in this situation.

5. Conclusion

This study proposed a HierFedAI collaborative training method and a CEC hierarchical FL architecture. HierFedAI analysis was given, and this resulted in several qualitative design recommendations. In studies, it was also demonstrated that, when compared to conventional cloud-based FL, it can simultaneously lower the method's training time along with the communication overhead of the edge devices. Three convolutional blocks, three activation blocks, and three batch normalization blocks are used in FL CNN training. Future research will be required to properly define and optimize these crucial factors, even though our work highlighted trade-offs in choosing the values of important HierFedAI algorithm parameters.

References

[1] Liu, Su, Jiong Yu, Xiaoheng Deng, and Shaohua Wan. "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks." IEEE Transactions on Intelligent Transportation Systems 23, no. 2 (2021): 1616-1629.

[2] Al-Quraan, Mohammad, Lina Mohjazi, Lina Bariah, Anthony Centeno, Ahmed Zoha, Kamran Arshad, Khaled Assaleh, Sami Muhaidat, Mérouane Debbah, and Muhammad Ali Imran. "Edge-native intelligence for 6G communications driven by federated learning: A survey of trends and challenges." IEEE Transactions on Emerging Topics in Computational Intelligence (2023).

[3] Barbieri, Luca, Stefano Savazzi, Mattia Brambilla, and Monica Nicoli. "Decentralized federated learning for extended sensing in 6G connected vehicles." Vehicular Communications 33 (2022): 100396.

[4] Math, Sa, Prohim Tam, and Seokhoon Kim. "Reliable federated learning systems based on intelligent resource sharing scheme for big data internet of things." IEEE Access 9 (2021): 108091-108100.

[5] Khan, Mohammad Bariq, Xueli An, and Chenghui Peng. "Towards Native Support for Federated Learning in 6G." In 2023 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-6. IEEE, 2023.

[6] Wu, Jiajun, Steve Drew, Fan Dong, Zhuangdi Zhu, and Jiayu Zhou. "Topology-aware Federated Learning in Edge Computing: A Comprehensive Survey." arXiv preprint arXiv:2302.02573 (2023).

[7] Zhang, Tuo, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and A. Salman Avestimehr. "Federated learning for the internet of things: applications, challenges, and opportunities." IEEE Internet of Things Magazine 5, no. 1 (2022): 24-29.

[8] Zhou, Xiaokang, Wei Liang, Jinhua She, Zheng Yan, I. Kevin, and Kai Wang. "Two-layer federated learning with heterogeneous model aggregation for 6g supported internet of vehicles." IEEE Transactions on Vehicular Technology 70, no. 6 (2021): 5308-5317.

[9] Sirohi, Deepika, Neeraj Kumar, Prashant Singh Rana, Sudeep Tanwar, Rahat Iqbal, and Mohammad Hijjii. "Federated learning for 6G-enabled secure communication systems: a comprehensive survey." Artificial Intelligence Review (2023): 1-93.

[10] Khowaja, Sunder Ali, Kapal Dev, Parus Khowaja, and Paolo Bellavista. "Toward energy-efficient distributed federated learning for 6G networks." IEEE Wireless Communications 28, no. 6 (2021): 34-40.

[11] Renda, Alessandro, Pietro Ducange, Francesco Marcelloni, Dario Sabella, Miltiadis C. Filippou, Giovanni Nardini, Giovanni Stea et al. "Federated Learning of Explainable AI Models in 6G Systems: Towards Secure and Automated Vehicle Networking." Information 13, no. 8 (2022): 395.

[12] Kalapaaking, Aditya Pribadi, Veronika Stephanie, Ibrahim Khalil, Mohammed Atiquzzaman, Xun Yi, and Mahathir Almashor. "SMPC-Based Federated Learning for 6G-Enabled Internet of Medical Things." IEEE Network 36, no. 4 (2022): 182-189.