Dynamic-Fusion-Based Federated Learning for COVID-19 Detection

Weishan Zhang, Tao Zhou, Qinghua Lu, *Senior Member, IEEE*, Xiao Wang, *Member, IEEE*, Chunsheng Zhu, *Member, IEEE*, Haoyun Sun, Zhipeng Wang, Sin Kit Lo, and Fei-Yue Wang

***Abstract*—Medical diagnostic image analysis (e.g., CT scan or X-Ray) using machine learning is an efficient and accurate way to detect COVID-19 infections. However, the sharing of diagnostic images across medical institutions is usually prohibited due to patients' privacy concerns. This causes the issue of insufficient data sets for training the image classification model. Federated learning is an emerging privacy-preserving machine learning paradigm that produces an unbiased global model based on the received local model updates trained by clients without exchanging clients' local data. Nevertheless, the default setting of federated learning introduces a huge communication cost of transferring model updates and can hardly ensure model performance when severe data heterogeneity of clients exists. To improve communication efficiency and model performance, in this article, we propose a novel dynamic fusion-based federated learning approach for medical diagnostic image analysis to detect COVID-19 infections. First, we design an architecture for dynamic fusion-based federated learning systems to analyze medical diagnostic images. Furthermore, we present a dynamic fusion method to dynamically decide the participating clients according to their local model performance and schedule the model fusion based on participating clients' training time. In addition, we summarize a category of medical diagnostic image data sets for COVID-19 detection, which can be used by the machine learning community for image analysis. The evaluation results show that the proposed approach is feasible and performs better than the default setting of federated learning in terms of model performance, communication efficiency, and fault tolerance.**

***Index Terms*—AI, COVID-19, CT, federated learning, image processing, machine learning, X-Ray.**

Manuscript received September 30, 2020; revised December 7, 2020 and January 12, 2021; accepted January 27, 2021. Date of publication February 4, 2021; date of current version October 22, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62072469; in part by the National Key Research and Development Program under Grant 2018YFE0116700 and Grant 2020YFB2104301; in part by the Shandong Provincial Natural Science Foundation (Parallel Data-Driven Fault Prediction Under Online–Offline Combined Cloud Computing Environment) under Grant ZR2019MF049; in part by the Fundamental Research Funds for the Central Universities under Grant 2015020031; in part by the Project "PCL Future Greater-Bay Area Network Facilities for Large-Scale Experiments and Applications" under Grant LZC0019; in part by the Special Project of West Coast Artificial Intelligence Technology Innovation Center under Grant 2019-1-5 and Grant 2019-1-6; in part by the Opening Project of Shanghai Trusted Industrial Control Platform under Grant TICPSH202003015-ZC; and in part by the Project "Beihang Beidou Technological Achievements Transformation and Industrialization Funds" under Grant BARI2005. *(Corresponding authors: Weishan Zhang; Qinghua Lu; Chunsheng Zhu.)*

Weishan Zhang, Tao Zhou, Haoyun Sun, and Zhipeng Wang are with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China (e-mail: zhangws@upc.edu.cn).

Qinghua Lu and Sin Kit Lo are with Data61, CSIRO, Sydney, NSW 2015, Australia, and also with School of Computer Science and Engineering, University of New South Wales, NSW 2052, Australia (e-mail: qinghua.lu@data61.csiro.au).

Xiao Wang and Fei-Yue Wang are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Chunsheng Zhu is with the Institute of Future Networks, Southern University of Science and Technology, Shenzhen 518055, China, and also with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: chunsheng.tom.zhu@gmail.com).

Digital Object Identifier 10.1109/JIOT.2021.3056185

I. Introduction

The COVID-19 pandemic has caused an unprecedented global crisis. Currently, the rapidly increasing number of COVID-19 cases leads to a severe shortage of test kits and calls for a more efficient and accurate way to diagnose COVID-19 infections. To address the COVID-19 diagnosis kits shortage issue, researchers have been applying machine learning technologies, especially deep learning on medical diagnostic image (e.g., CT scan or X-Ray) recognition. However, the deep learning model performance is heavily dependent on the training data set size and diversity. Moreover, data hungriness is a critical challenge due to the concern for data privacy. To protect privacy-sensitive patients' data, the sharing of medical data across medical institutions is not allowed, which causes the issue of insufficient data sets for model training.

The concept of federated learning was first introduced by Google in 2016 as a new machine learning paradigm that produces an unbiased model while preserving data privacy [1], [2]. In each round of training, clients (e.g., organizations, data centers, or mobile/IoT devices) are selected to train a model using local data and the local model updates are sent to a central server for aggregation without transferring any local raw data.

Federated learning has the potential to connect isolated medical institutions and train a model for COVID-19 positive case detection while preserving data privacy. Some recent works leverage federated learning to diagnose COVID-19 infection through CT or X-Ray images [3], [4]. However, the above studies adopted the default setting of federated learning that requires massive communication costs of transferring model updates (e.g., massive matrices of weights) and underperforms when data heterogeneity of clients heavily exists.

To improve communication efficiency and model performance, we propose a novel dynamic fusion-based federated learning approach for COVID-19 positive case detection. First, we design a dynamic fusion-based federated learning system architecture for medical diagnosis image analysis to detect COVID-19 positive cases. The proposed architecture

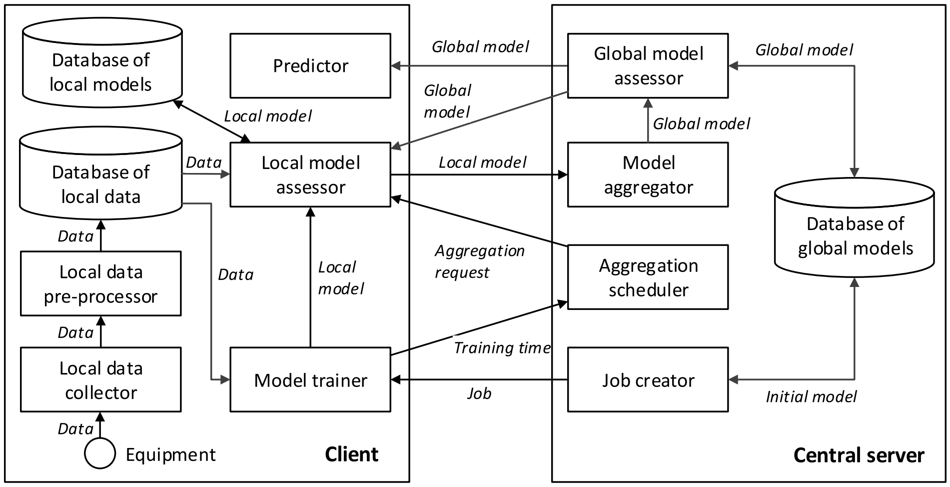


Fig. 1. Architecture of federated learning systems for medical diagnostic image analysis.

provides a systematic view of the components' interactions and serves as a guide for the federated learning system design. Second, we present a dynamic fusion method to decide the participating clients according to their local model performance and schedule the model fusion dynamically, based on the participating clients' training time. Each client assesses the local model trained and only uploads the model updates when it performs better than the previous version. The central server configures the waiting time for each client to send model updates, based on the average training time of the previous round. Additionally, we summarize a category of medical diagnostic image data sets for COVID-19 detection, which can be used by the machine learning community for image analysis research. The evaluation results show that the proposed approach achieves better detection accuracy, fault tolerance, and communication efficiency compared to the default setting of federated learning.

The remainder of this article is organized as follows. Section II presents the approach. Section III evaluates the approach. Section IV discusses the related work. Finally, Section V concludes this article.

II. Dynamic Fusion-Based Federated Learning for COVID-19 Detection

In this section, we present a dynamic fusion-based federated learning approach for CT scan image analysis to diagnose COVID-19 infections. Section II-A provides an overview of the architecture and discusses the components and their interactions. Section II-B discusses a dynamic model fusion method to dynamically decide the participating clients and schedule the aggregation based on each client's training time.

A. Architecture

Fig. 1 illustrates the architecture, which consists of two types of components: 1) central server and 2) clients. The central server initializes a machine learning job and coordinates the federated learning process, while clients train local models specified in the learning job using local data and computation resources.

Each client gathers images scanned by the diagnostic imaging equipment through the client data collector and cleans the data (e.g., noise reduction) via the client data preprocessor and store locally. The job creator initializes a model training job (including initial model code and the number of aggregations) and configures the initial waiting time for each client to return the model updates. Each participating client downloads the job and trains the model via the model trainer. After a set number of epochs, the model trainer completes this round of training and uploads the training time to the central server. The aggregation scheduler updates the waiting time based on the training time received from participating clients.

The local model assessor on each client compares the performance of the current local model with the previous version. If the current local model performs better, the client sends a request for model upload to the central server. Otherwise, the client will request to not upload the model update for this round. All the clients that did not complete the set number of epochs within the waiting time are not allowed to participate in this aggregation round. After the set waiting time, the aggregation scheduler on the central server notifies the clients that have sent the model upload request. Finally, after the aggregation, the global model assessor measures the accuracy of the aggregated global model and sends the global model back to each client for a new training round.

B. Dynamic Fusion

To improve communication efficiency in federated learning, the proposed dynamic fusion method consists of two decision-making points: 1) client participation and 2) client selection. On the client side, each client decides whether to join this aggregation round based on the performance of the newly trained model. On the central server side, the model aggregator selects the participating clients based on the waiting time. The initial waiting time is configured by the platform owner and the waiting time of the current round for a client is calculated by averaging its previous round's training time. If a client does not upload the model update within the waiting time, it will be excluded by the central server for this aggregation round.

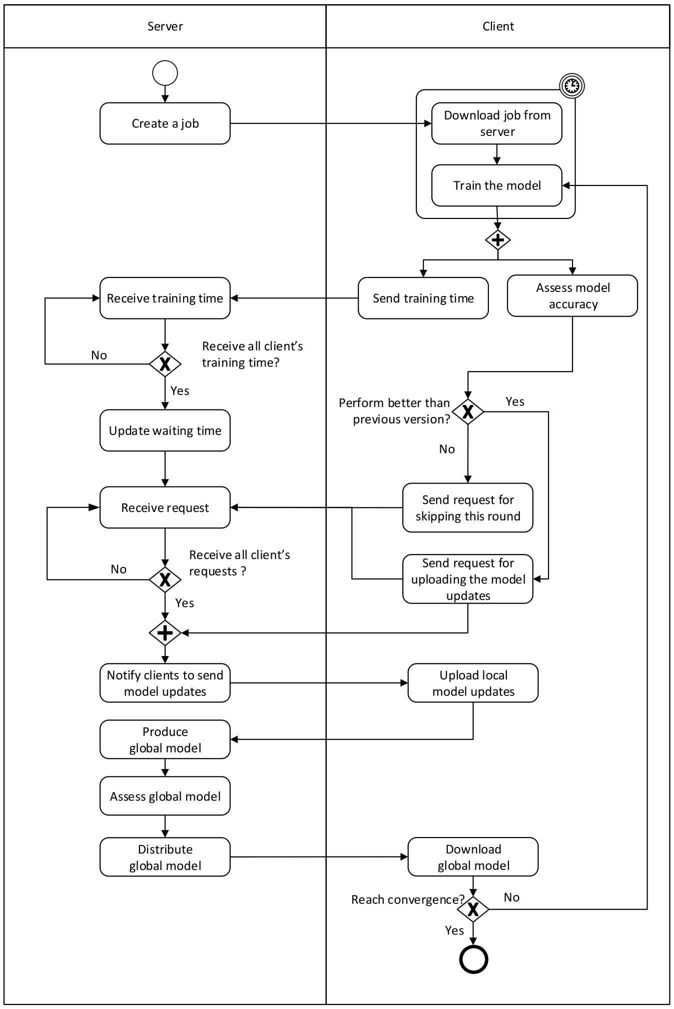


Fig. 2. Dynamic fusion process.

Algorithm 1 Dynamic Fusion Algorithm

/*Client*/
registerClient(*ServerURL*)
$Job \leftarrow$ download(*ServerURL*)
$FusionTimes, Model \leftarrow$ decode(*Job*)
$FedStep \leftarrow 0$
$Lr, BatchSize, Epoch \leftarrow$ initialize()
while $FedStep < FusionTimes$ **do**
    $Acc, TrainingTime \leftarrow$ train($Model, Lr, BatchSize, Epoch$)
    send(*TrainingTime*)
    $MaxAcc \leftarrow$ request(*ServerURL*)
    if $Acc \geq MaxAcc$ **then**
        upload(*Model*)
        $Model \leftarrow$ receiveModel()
    end if
    $FedStep++$
end while

/*Server*/
$WaitingTime, MaxAcc, FusionTimes, Model \leftarrow$ initialize()
$Job \leftarrow$ encode($FusionTimes, Model$)
while true **do**
    $TrainingTime \leftarrow$ receive()
    $WaitingTime \leftarrow$ update(*TrainingTime*)
    $ClientModel \leftarrow$ receiveModel()
    if expired(*WaitingTime*) $==$ *true* **then**
        $Model \leftarrow$ aggregate(*ClientModel*)
        $MaxAcc \leftarrow$ evalate(*Model*)
        dispatch(*Model*)
    end if
end while

Fig. 2 illustrates the process of the proposed dynamic fusion method, and Algorithm 1 describes the detailed process. The process starts by creating a learning job by the central server. All the clients download the job from the central server and set up the local training environment. From the second round onward, a timer is set for each client based on the average training time of all the participating clients for the previous round. If a client does not complete the training within the configured time, the central server proceeds the aggregation without any input from this client for this round. On the other hand, if the model trained by the client for this round performs worse than the previous round, the client sends a request to the central server for skipping this round's aggregation. Otherwise, the client notifies the central server to update the model.

III. Evaluation

Table II summarizes a category of medical diagnostic image data sets for COVID-19 detection with 746 CT images and 2960 X-ray images. The 746 CT data set includes 349 images of COVID-19 positive diagnoses and 397 images of negative diagnoses. The chest X-ray images are from two data sets. The first X-ray data set has 2905 images which contain 219 images of COVID-19 positive diagnoses, 1341 images of negative diagnoses, and 1345 images of viral pneumonia (VP) diagnoses. The second X-Ray data set consists of 55 images of positive diagnoses. The proposed approach is evaluated via quantitative experiments using the data sets as shown in Table II.

As shown in Table III, the experiments involve one central server and three clients with different configurations. We selected 3326 images from the collected data sets and divided them into 2800 images for the training set and 526 images for the test set. We set different data set sizes for each client: 600 images, 900 images, and 1300 images, respectively. Considering the difference between CT and X-ray images, we adjusted the ratio of these two types of images while keeping the same total amount for each client, which is shown in Table I. In the test set, there are 71 CT images (31 COVID-19 positive diagnoses, and 40 negative diagnoses), and 455 X-ray images (55 COVID-19 positive diagnoses, 200 negative diagnoses, and 200 virus pneumonia diagnoses). Please note that the CT images are taken from the top, while the X-Ray images are taken from the front.

We conducted experiments using three different models, GhostNet, ResNet50, and ResNet101.

TABLE I
DATA SET CONFIGURATION FOR EACH CLIENT

Client 1	Data Size (MB)	Client 2	Data Size (MB)	Client 3	Data Size	Ratio	Total Data Size (MB)	Amount
600/0	76.8	0/900	391.3	0/1300	545.7	600/2200	1013.8	2800
300/300	168.5	0/900	392.5	0/1300	546.6	300/2500	1107.6	
200/400	196.8	0/900	389.1	0/1300	534.5	200/2600	1120.4	
150/450	209.9	0/900	381.6	0/1300	544	150/2650	1135.5	
200/400	197.4	200/700	318.9	0/1300	557.5	400/2400	1073.8	
200/400	198.6	200/700	317.2	200/1100	497	600/2200	1012.8	

TABLE II
CATEGORY OF MEDICAL DIAGNOSTIC IMAGE DATA SETS FOR COVID-19 DETECTION

Type	Amount	Size	COVID-19	Negative	VP	Github Address
CT	746	92.6M	349	397	0	https://github.com/UCSD-AI4H/COVID-CT
X-ray	2905	1168M	219	1341	1345	https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
X-ray	55	14.2M	55	0	0	https://github.com/agchung/Figure1-COVID-chestxray-dataset

TABLE III
EXPERIMENTAL ENVIRONMENT

Node	GPU	RAM	Python	CUDA
Server	RTX 2080Ti	11G	3.6	10.0
Client1	GTX 1070	8G	3.6	10.1
Client2	GTX 1080	8G	3.6	10.1
Client3	TITAN X(Pascal)	12G	3.7	10.1

A. Accuracy

To evaluate the accuracy of the dynamic fusion-based federated learning (DF_FL), the models are trained with the six groups of data sets listed in Table I. There are 18 groups of experiments in total. We compared the results with the default setting of federated learning (D_FL). The GFL federated learning framework[1] is used in our experiments.

The results are presented in Figs. 3–5, respectively, for each type of model. The results show that in the 18 groups of experiments, there are only four groups in which the dynamic fusion-based federated learning (DF_FL) achieves accuracy lower than the default setting of federated learning (D_FL) (lower by 1.711%, 0.57%, 0.57%, and 1.141%, respectively). The remaining 14 groups of dynamic fusion-based federated learning (DF_FL) achieve accuracy higher than the default setting of federated learning (D_FL). Overall, the proposed dynamic fusion-based federated learning approach achieved higher accuracy compared to the default setting of federated learning. Also, an interference is introduced to the fourth group of the data set for each model, where images of negative diagnosis are marked as positive COVID-19. The model trained by fusion-based federated learning can still achieve relatively steady results and higher accuracy compared to the default setting, which shows that the proposed fusion-based federated learning can ensure fault tolerance and robustness.

[1]https://github.com/GalaxyLearning/GFL

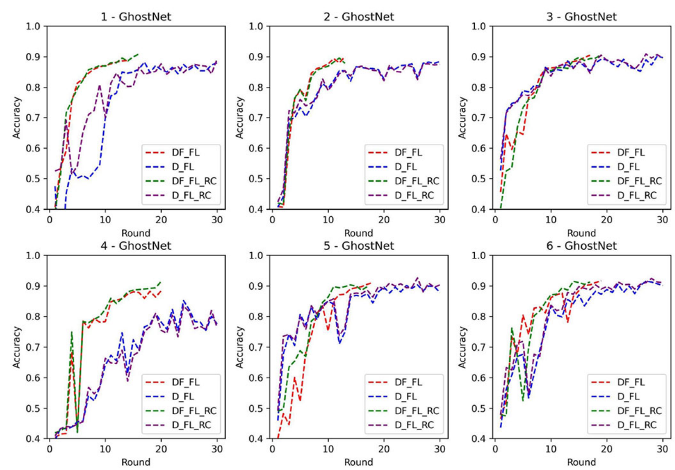


Fig. 3. Accuracy of GhostNet.

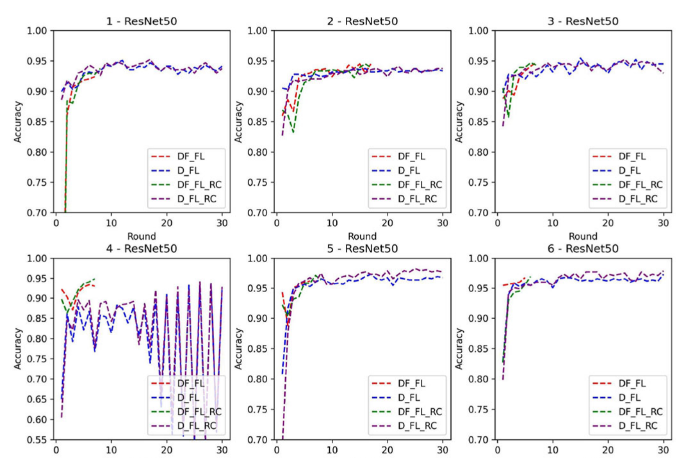


Fig. 4. Accuracy of ResNet50.

In addition, we measured the accuracy of each type of model using the test set which was processed by random cropping. The results are also shown in Figs. 3–5. Similarly, the results demonstrate that the proposed dynamic fusion-based federated

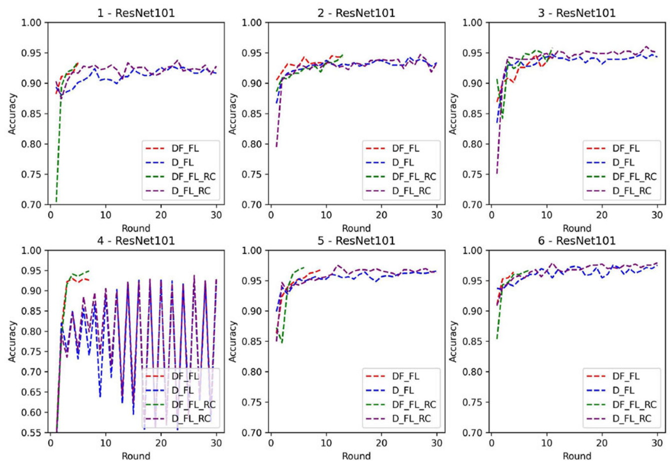


Fig. 5. Accuracy of ResNet101.

learning (DF_FL) achieves higher accuracy than the default setting of federated learning (D_FL) in 14 groups of experiments. For the rest, DF_FL have lower accuracy compared to D_FL for 0.57%, 1.331%, 0.951%, and 1.141%, respectively The results show that the proposed fusion-based federated learning performs better in real-world data sets than the default setting of federated learning.

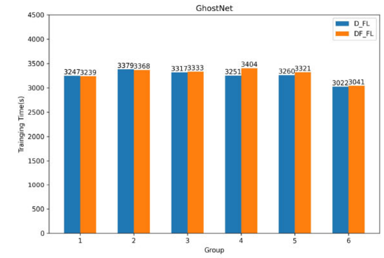
B. Training Time

To evaluate the training efficiency of the proposed dynamic fusion-based federated learning, we recorded the training time of the experiments. The training epochs of the clients are set to 90 and the maximum network speed is configured to 10 MB/s for model upload/download (10 MB/s). The recorded training time is illustrated in Fig. 6. The results show that in GhostNet, the proposed dynamic fusion-based federated learning does not reduce the training time, but there is an apparent reduction on ResNet50 and ResNet101. The training time of ResNet50 is reduced by 8–10 min, while the training time of ResNet101 is decreased by 25–30 min.

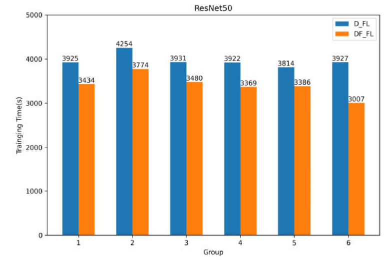
Since we found out that the proposed dynamic fusion-based federated learning cannot reduce the training time of the GhostNet network in the above experiments, we further study the influence factor. After measuring the single-model transmission time, we observed that the GhostNet has fewer parameters compared to the other two networks. Thus, the GhostNet costs less time for model transmission (which is 2.2 s on average), which results in no change in GhostNet training time. In contrast, ResNet50 and ResNet101 have more parameters that take more time to transmit the model updates. Thus, there is an apparent improvement in these two networks in terms of communication efficiency. We can conclude that applying the proposed dynamic fusion-based approach can significantly reduce the training time when the network is poor and the model has large amounts of parameters.

C. Communication Efficiency

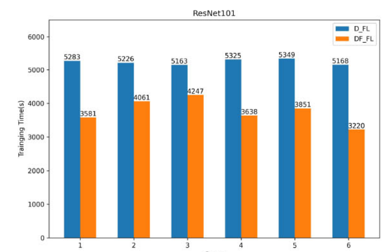
To evaluate the effect of dynamic fusion on communication, we measure the upload number and the upload time, which are



(a)



(b)



(c)

Fig. 6. Training time. (a) GhostNet. (b) ResNet50. (c) ResNet101.

shown in Figs. 7 and 8, respectively. Here, the collected upload number and time are the total number of three clients, which is 30 times for each client and 90 times in total. In comparison with the default setting of federated learning (D_FL) for the GhostNet, the upload number of dynamic fusion is reduced by an average of 61, matching to a reduction of 110–160 s of the upload time (1/3 of the D_FL time). For ResNet50, the upload number of dynamic fusion is decreased by an average of 80, matching to a reduction of 900–1200 s of the upload time (1/10 of the D_FL time). For ResNet101, the upload number of dynamic fusion is decreased by an average of 78, matching to a reduction of 3200–4200 s on the upload time (1/16 of the D_FL time).

Based on the results, we can conclude that dynamic fusion is capable to reduce the communication overhead through fewer model uploading. For models with a simple structure and fewer parameters like the GhostNet, the reduction is not significant (only 1/3 of D_FL). Nevertheless, dynamic fusion has more obvious effects in treating complicated models with more parameters (ResNet50 and ResNet101), which are scaled down to 1/10 and 1/16 of the D_FL time.

IV. RELATED WORK

The concept of federated learning is first proposed by Google in 2016 [1], which initially focuses on cross-device

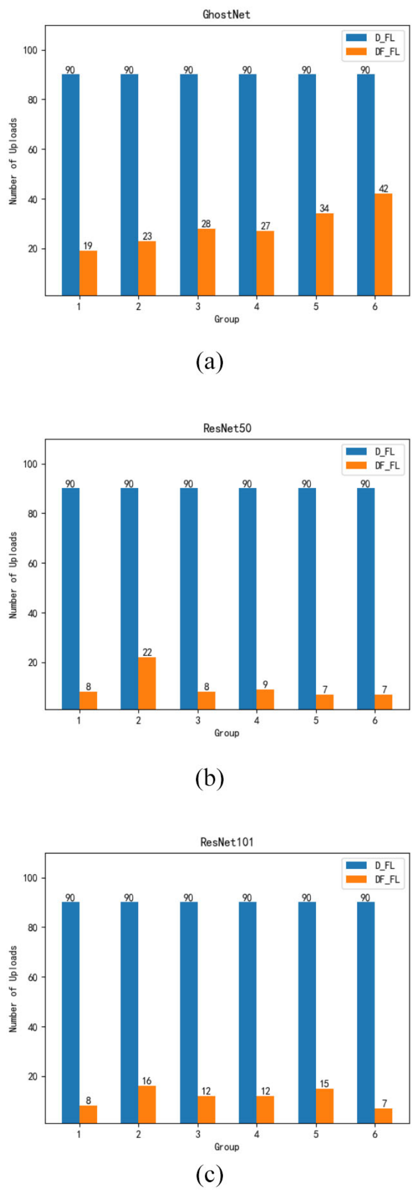


Fig. 7. Upload number. (a) GhostNet. (b) ResNet50. (c) ResNet101.

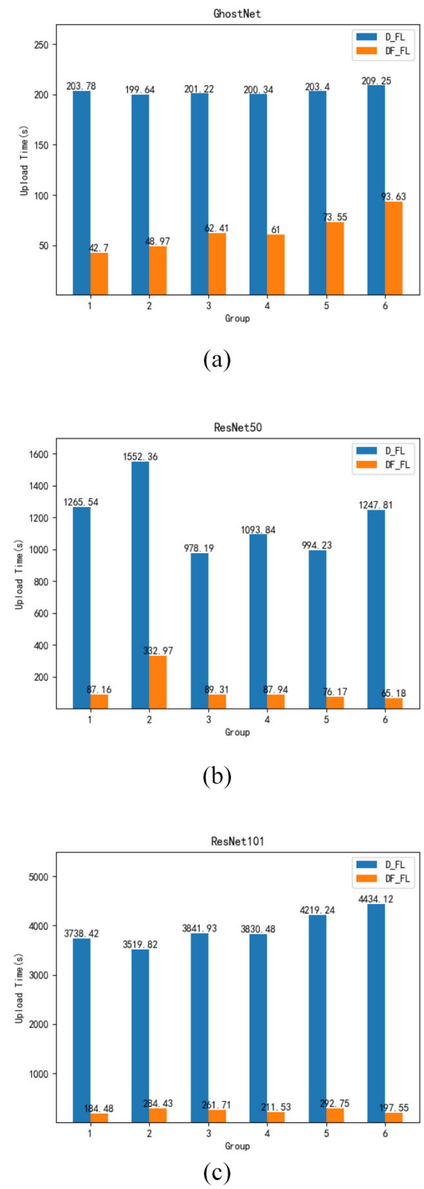


Fig. 8. Upload time. (a) GhostNet. (b) ResNet50. (c) ResNet101.

learning. Google initially adopted federated learning to predict search suggestions, next words and emojis, and the learning of out-of-vocabulary words [5]–[7]. The scope of federated learning is then extended to cross-silo learning, e.g., for different organizations or data centers [2], [8], [9]. For example, Sheller *et al.* [10] built a segmentation model using brain tumor data from different medical institutions. Furthermore, the concept of federated learning can also be adopted to edge computing, such as task scheduling process in Internet of Vehicles [11]–[14].

Although communication efficiency can be improved by only sending model updates instead of raw data, federated learning systems require multiple rounds of communications during training to achieve model convergence. Many researchers work on the methods to reduce communication rounds [15], [16]. One way is through aggregation, e.g., selective aggregation [17], aggregation scheduling [18], asynchronous aggregation [19], temporally weighted aggregation [20], controlled averaging algorithms [21], iterative round reduction [15], and shuffled model aggregation [22]. Furthermore, model compression methods are utilized to reduce the communication cost that occurs during the model parameters and gradients exchange between clients and the central server [23]. Additionally, communication techniques are introduced to improve communication efficiency, e.g., over-the-air computation technique [24] and multichannel random access communication mechanism [25].

Federated learning can address statistical and system heterogeneity issues since models are trained locally [26]. However, challenges still exist in dealing with non-IID data. Many researchers have worked on training data clustering [27], multistage local training [28], and multitask learning [26]. Also, some works [29], [30] focus on incentive mechanism design to motivate clients to participate in the machine learning jobs.

Federated learning has been recently adopted in CT or X-Ray image processing for COVID-19 positive case detection [3], [4]. However, the above studies do not consider the communication efficiency and model accuracy issues of federated learning. Hence, our research work proposed a dynamic fusion-based approach to improve communication efficiency and model performance.

V. CONCLUSION

This article proposes a novel dynamic fusion-based federated learning approach to improve accuracy and communication efficiency while preserving data privacy for COVID-19 detection. The evaluation results show that the proposed approach is feasible and performs better than the default setting of federated learning in terms of model accuracy, fault tolerance, robustness, and communication efficiency.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. Y. Arcas. (2016). *Federated Learning of Deep Networks Using Model Averaging*. [Online]. Available: http://arxiv.org/abs/1602.05629

[2] S. K. Lo, Q. Lu, C. Wang, H. Paik, and L. Zhu, "A systematic literature review on federated machine learning: From a software engineering perspective," 2020. [Online]. Available: arxiv.abs/2007.11354

[3] B. Liu, B. Yan, Y. Zhou, Y. Yang, and Y. Zhang, "Experiments of federated learning for COVID-19 chest *x*-ray images," 2020. arXiv:2007.05592.

[4] R. Kumar *et al.*, "Blockchain-federated-learning and deep learning models for COVID-19 detection using CT imaging," 2020. arXiv:2007.06537.

[5] T. Yang *et al.* (2018). *Applied Federated Learning: Improving Google Keyboard Query Suggestions*. [Online]. Available: http://arxiv.org/abs/1812.02903

[6] M. Chen, R. Mathews, T. Ouyang, and F. Beaufays, "Federated learning of out-of-vocabulary words," 2019. arXiv:1906.04329.

[7] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," 2019.

[8] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019. [Online]. Available: arXiv:1912.04977

[9] W. Zhang *et al.*, "Blockchain-based federated learning for device failure detection in industrial IoT," *IEEE Internet Things J.*, early access, Oct. 20, 2020, doi: 10.1109/JIOT.2020.3032544.

[10] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation," in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, A. Crimi, S. Bakas, H. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum, Eds. Cham, Switzerland: Springer Int., 2019, pp. 92–104.

[11] Z. Ning *et al.*, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 463–478, Feb. 2021.

[12] Z. Ning *et al.*, "Intelligent edge computing in Internet of Vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TITS.2020.2997832.

[13] Z. Ning *et al.*, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Trans. Mobile Comput.*, early access, Sep. 18, 2020, doi: 10.1109/TMC.2020.3025116.

[14] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Trans. Mobile Comput.*, early access, Jul. 28, 2020, doi: 10.1109/TMC.2020.3012509.

[15] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5986–5994, Jul. 2020.

[16] S. Silva, B. A. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi, "Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 270–274.

[17] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Apr. 2020.

[18] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[19] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," 2019. arXiv:1903.03934.

[20] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[21] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," 2019. [Online]. Available: arXiv:1910.06378

[22] B. Ghazi, R. Pagh, and A. Velingker, "Scalable and differentially private distributed aggregation in the shuffled model," 2019. arXiv:1906.08320.

[23] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 954–964.

[24] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[25] J. Choi and S. R. Pokhrel, "Federated learning with multichannel aloha," *IEEE Wireless Commun. Lett.*, vol. 9, no. 4, pp. 499–502, Apr. 2020.

[26] L. Corinzia and J. M. Buhmann, "Variational federated multi-task learning," 2019. [Online]. Available: arXiv:1906.06268

[27] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multi-task optimization under privacy constraints," 2019. [Online]. Available: arXiv:1910.01991

[28] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan, "Improving federated learning personalization via model agnostic meta learning," 2019. [Online]. Available: arXiv:1909.12488

[29] Y. J. Kim and C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *Proc. 20th Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS)*, Sep. 2019, pp. 1–4.

[30] H. Kim, J. Park, M. Bennis, and S. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

Weishan Zhang received the Ph.D. degree in mechanical manufacturing and automation from Northwestern Polytechnical University, Xi'an, China, in 2001.

He is a Full Professor and the Deputy Head for Research with the Department of Software Engineering, School of Computer and Communication Engineering, China University of Petroleum, Qingdao, China. His research interests include big data platforms, pervasive cloud computing, service-oriented computing, and federated learning.

Tao Zhou is currently pursuing the postgraduate degree with the College of Computer Science and Technology, China University of Petroleum (East China), Qingdao, China.

His research interests include computer vision and federated learning.

Qinghua Lu (Senior Member, IEEE) received the Ph.D. degree from the University of New South Wales, Sydney, NSW, Australia, in 2013.

She is a Senior Research Scientist with Data61, CSIRO, Sydney. Before she joined Data61, she was an Associate Professor with China University of Petroleum, Qingdao, China. She formerly worked as a Researcher with National ICT Australia, Sydney. She has published over 100 academic papers in international journals and conferences. Her recent research interests include architecture design of blockchain systems, blockchain for federated/edge learning, self-sovereign identity, and software engineering for machine learning.

Xiao Wang (Member, IEEE) received the Ph.D. degree in social computing from the University of Chinese Academy of Sciences, Beijing, China, in 2016.

She is currently an Associate Professor with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing. She has published more than a dozen SCI/EI articles and translated three technical books (English to Chinese). Her research interests include social transportation, cybermovement organizations, artificial intelligence, and social network analysis.

Dr. Wang has served the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and *ACM Transactions on Intelligent Systems and Technology* as a Peer Reviewer with a good reputation.

Chunsheng Zhu (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2016.

He is an Associate Professor with the Institute of Future Networks, Southern University of Science and Technology, Shenzhen, China. He is also an Associate Researcher with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen. He has authored more than 100 publications published by refereed international journals, magazines, and conferences. His research interests mainly include Internet of Things, wireless sensor networks, cloud computing, big data, social networks, and security.

Haoyun Sun is currently pursuing the postgraduate degree with China University of Petroleum (East China), Qingdao, China.

His main research areas are big data processing, computer vision, and software engineering. He has published two papers in these areas.

Zhipeng Wang is currently pursuing the postgraduate degree with China University of Petroleum (East China), Qingdao, China.

His main research areas are computer vision and federated learning.

Sin Kit Lo is currently pursuing the Ph.D. degree with the University of New South Wales, Sydney, NSW, Australia, and CSIRO, Sydney.

His research interests include federated learning, blockchain, and software architecture.

Fei-Yue Wang received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the University of Arizona, Tucson, AZ, USA, in 1990 and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory for Management and Control of Complex Systems. His current research focuses on methods and applications for parallel intelligence, social computing, and knowledge automation.

Prof. Wang is a Fellow of INCOSE, IFAC, ASME, and AAAS. He is currently the President of CAA's Supervision Council, IEEE Council on RFID, and the Vice President of the IEEE Systems, Man, and Cybernetics Society.