A Federated Learning-based Solution for Pneumonia Diagnosis in Remote and Low-Income Areas

Mateus R. da Cruz, Hyago V. L. B. Silva, Felipe A. P. de Figueiredo and Samuel B. Mafra

***Abstract*— The digital transformation in healthcare is a recurring theme around the world. However, the continental extension of Brazil makes it difficult to offer basic healthcare services in remote locations. Due to the scarcity of diagnosis services in these areas, this article proposes leveraging Federated Learning as a way to reduce costs and mitigate these problems, helping with pre-diagnosis. The experiments carried out showed that just using the federated approach can increase the model's predictive capacity and reduce training time. The model developed using Federated Learning increased the model's accuracy by 14%, while managing to reduce the model's loss in the validation set by 1.0354.**

***Keywords*— Federated Learning, Internet of Things, Radiography, Deep Learning.**

I. INTRODUCTION

Technological advancements in the medical field have greatly improved the diagnosis and treatment of various diseases, making them more effective, affordable, and less burdensome for patients and clinicians. The emergence of new medical technologies offers opportunities for the early detection of diseases through data that are used to analyze the patient's health [1]. Traditionally, in the field of radiology, medically trained professionals were responsible for visually assessing medical images and identifying, characterizing, and classifying diseases. However, Computer Vision (CV), a subfield of Artificial Intelligence (AI), has emerged as a promising alternative, capable of automatically recognizing complex patterns in data images and providing quantitative assessments, as opposed to human assessments of radiography results [2], [3].

Simultaneously, the rise of devices dedicated to patient health underscores the emergence of the AI of Med Things (AIoMT) as an approach to implementing AI at the edge. The growing integration between advances in IoT and emerging demands in healthcare reflects a trend toward more accessible, personalized, and proactive services through AIoMT. In this context, AIoMT is becoming fundamental in healthcare management, playing a significant role in optimizing care and promoting a more effective approach to the well-being of patients [4]. However, a few more technologies are needed for implementation and collaboration between hospitals and healthcare institutions in the area of Big Data. One of these technologies is Federated Learning (FL), a technology that enables the joint development of AI models [5].

Distributed training, such as FL, has emerged as a promising approach, allowing collaborative training of models by distributing the learning process across multiple devices. This overcomes challenges such as privacy, data heterogeneity, and communication efficiency while maintaining the security and anonymity of patient data. These approaches have been successfully applied to medical image analysis and tackle important issues such as privacy preservation and fairness [6], [7].

This article proposes an AIoMT solution that integrates CV and FL into an IoT framework that can be easily installed in hard-to-reach places. CV aims to diagnose numerous images quickly and efficiently using the Squeezenet V1.0 architecture. FL, on the other hand, enables better collaboration between hospitals and institutions that adopt the approach. It guarantees the best model for everyone involved and the non-disclosure of sensitive data. Finally, the IoT framework makes installing the application in remote and low-resource areas possible, ensuring that emerging countries can access it. Not only that, but the multiple experiments that have been carried out prove that the use of FL in model training can help increase the model's predictive capabilities in SqueezeNet and MobileNet architectures. Therefore, the methodology presented for training models can be applied to different databases, scenarios, and contexts for effective training. The article is structured as follows: Section II introduces related literature. Section III describes the employed methodology. Section IV presents the results obtained, along with their qualitative discussion. Finally, Section V concludes the study and outlines future research directions.

II. RELATED WORKS

The systematic review presented in [8] revealed significant findings that point to the potential use of emerging technologies such as Edge Computing Models (ECMs), Unmanned Aerial Vehicles (UAVs), IoT, cloud sensor networks, and Machine Learning (ML). These tools can potentially improve efficiency in disaster visualization, analysis, and prediction, as well as empower healthcare professionals to deal with emergencies more effectively. In contrast to the previous study, which focused on comparing and identifying a specific model for chest disease, this work suggests a more complex approach, proposing different models for each case and aiming for a more accurate analysis that is adaptable to the circumstances. The implications of this research are vast, including the possibility of improving preparedness and response strategies in disaster medicine through the implementation of advanced remote technologies such as IoT, ML, and virtual and augmented reality.

Meanwhile, a study on the VGG-19 model for detecting chest diseases focuses on analyzing multiple X-ray images, emphasizing DL and knowledge transfer. On the other side, a systematic review of emerging technologies to improve disaster efficiency shows the importance of IoT, ML, and virtual reality, suggesting different models for each case. In addition, an article on the use of blockchain for ensuring the security of health systems based on AI emphasizes the protection of health data. It proposes solutions to mitigate vulnerabilities against adversary attacks, pointing to blockchain as a promising solution to strengthen the security of health systems. The work [9] focuses on the classification of pneumonia in chest X-ray images using TF, specifically using Convolutional Neural Networks (CNNs). The study uses a pre-trained ResNet50 model, which is centrally trained. Their model achieves an accuracy of 91.8 %.

The study presented in [10] analyzes the performance of various AI architectures, detailing the use of processing, the number of parameters in each model, and their accuracy using the Image-Net database. It is important to note that the performance of these models can vary depending on the type of database. In turn, the present article proposes the adoption of models with fewer parameters and processing to reduce the costs of IoT devices, achieving greater accuracy than the one attained by the first article, [11], with other FL models, emphasizing the influence of performance in relation to the diversity of databases in different architectures, as evidenced in [10].

Although several researchers have developed models, methodologies, and frameworks for producing AI models for the medical field, none focus on applying them to low-income or remote locations with constrained resources. Also, the potential of FL to improve the model during the training process has not been explored, a result shown in this article through the experiments carried out. Therefore, this study aims to develop an FL-based solution employing distributed IoT devices to serve remote and low-income regions. It brings these regions closer to urban centers through a service integrated with these devices, providing preliminary diagnostic information for patients and health professionals. A comparison is made between a centralized model with local training and an embedded model trained with FL, i.e., distributed training, revealing a significant improvement in accuracy and loss minimization. This proposes the distributed capacity of FL over locally trained models, focusing on data privacy. It is important to note that none of the models used in this work employed TF, i.e., they were trained from scratch.

III. Model Development and Training

This section presents the main details and how the proposed solution works, showing the main software and hardware components.

A. Application Architecture

The application can be separated into three layers: I) IoT, II) Local FL, and III) Server. The first layer is where the system demonstrates a tangible result for the user, being the layer where the clients will have contact with the application. This result is presented through effective and efficient models for classifying diseases. This means rapid image processing for diagnosis. The main actors in this layer are the doctors, patients, and other health professionals responsible for diagnostic imaging, communication with other systems and sectors responsible, such as the finance sector, diagnostic imaging sector, and others, using Hypertext Transfer Protocol (HTTP), client-server, with communication over the internet. In other words, the embedded device will receive the image after the X-ray examination through a photograph or scan so that the inference can be made and the pre-diagnosis can be reported.

Currently, the application has no graphical interfaces, so the image must be inserted into the model using commands. Therefore, the professional interested in examining the image through the application must first transfer the image to the server and then make the inference through the trained model using commands in the terminal. This is not an intuitive practice, but it must be borne in mind that each page must be designed exclusively for the context of the application, something done later and on demand.

The second layer is responsible for local training of the model, where FL is first applied. Each hospital, clinic, or institution of interest trains its model in isolation using the same CNN architecture in this layer. A data pre-processing stage is necessary at this layer to use homogeneous data for training the model since it expects images with certain height, width, and channel characteristics. The second stage is also responsible for sharing the gradients of the best local model with the server. The third layer is where the models are aggregated, and a global model is developed to distribute the best model among the clients.

B. The SqueezeNet V1.0 model

The SqueezeNet V1.0 is a compact CNN model designed for efficient computation and resource usage while maintaining competitive performance in CV tasks like image classification [12]. Its features make SqueezeNet an excellent choice for implementing embedded devices like those used in the proposed application. Table I is a brief comparison between models developed with computational efficiency in mind and their respective sizes. In this sense, the highlight of choosing SqueezeNet is its lean architecture. It achieves this by utilizing squeeze layers with 1x1 convolutions to drastically reduce parameters and employing fire modules for compact yet deep architecture. Despite its lighter parameter count, SqueezeNet maintains competitive performance and is widely available for various CV applications. According to the paper proposing this model, [12], the architecture has approximately 1 million parameters, with a model file of 4.8 MB and a processing capacity of 823 million FLOPS.

C. Hardware Components of the FL Application

The solution utilizes two Raspberry Pi 4B (Pi4B) boards for local model training, featuring specifications such as Broadcom BCM2711 processor, varying RAM capacities, wireless and Bluetooth capabilities, Gigabit Ethernet, USB ports, micro-HDMI ports, MIPI display and camera ports, audio and video outputs, hardware decoding capabilities, OpenGL

TABLE I
COMPARISON BETWEEN THE COMPUTATIONAL EFFICIENT DESIGN MODELS [13], [14].

Architecture	Size	Parameters	Top-5 Accuracy
MobileNet V1	16MB	4.3M	89.5 %
MobileNet V2	14MB	3.5M	90.1 %
MobileNet V3	20MB	5.47M	75.6 %
SqueezeNet V1.0	<0.5MB	1.24M	80.4 %
SqueezeNet V1.1	<0.5MB	1.23M	80.6 %

and Vulkan support, micro-SD card slot, and power options via USB-C or GPIO header. A laptop server, specifically an IdeaPad Gaming 3i 6th Generation, with powerful Intel Core processors and NVIDIA GeForce GTX graphics card, is used for weight aggregation. However, another Raspberry Pi board can be used for this purpose.

D. Software Components of the FL Application

The SqueezeNet model's training parameters must be adjusted systematically since the experiment aims to compare centralized and distributed approaches. In this way, all the parameters were defined identically for both the federated and centralized processes. The main parameters defined were the following: scaling: normalization, Optimizer: Adam, Learning rate: 0.001, and Batch size: 32. These are standard values used during the training of CV models that do not change during the training phase.

The FastAI framework was used to process the image and train the SqueezeNet model [15]. FastAI is a high-level DL library built on top of PyTorch that aims to make DL more accessible to practitioners and researchers by providing easy-to-use application programming interfaces (APIs) and a range of pre-built models and utilities. On the other hand, the Flower framework was used for FL training [16]. The framework is an open-source platform designed to facilitate FL tasks, providing a simple API for FL and allowing developers to define tasks, models, and data sources easily. It supports various ML frameworks like TensorFlow and PyTorch and provides features for dynamic model averaging, secure aggregation, and asynchronous training. Google Remote Procedure Call (gRPC) is the API system used for communication between the two boards and the aggregator server [17]. In gRPC, a component (client/Raspberry Pi) calls or invokes specific functions in another software component (server/computer).

GRpC enables communication to take place in the application through the exchange of information and coordination between devices. Within the application, there are specific functions and registers for each stage of model training. For example, the strartFlwrGRPC function is used to initiate communication between the clients and the server. At the same time, there is a register responsible for receiving, sharing, and storing the parameters and messages exchanged by the devices. GRPC is, therefore, used throughout the communication process between devices, both for Clients and Servers and vice versa.

E. Federated Learning-based Training

ML models, including CNNs, learn by iterative adjusting their weights to minimize a loss function. This loss function quantifies the model's performance on a training dataset. To achieve optimal performance, the model must move towards a point in weight space where the loss function is minimized. Gradients play a crucial role in this optimization process. They are the partial derivatives of the loss function concerning the model's weights. These gradients indicate the direction and magnitude of change required for the weights to reduce the loss function.

FL allows training a single model across devices without sharing private/sensitive data. The two devices train a local copy of the model on their own information and calculate gradients, which indicate how to improve the model. Instead of the raw data, these gradients are shared with a central server, which averages them to update the global model. This updated model is then distributed back to the two devices for the next training round. Therefore, the proposed application works as follows:

1) The application is started on the server, randomly capturing the weights initialized on one of the clients to configure the global model;
2) The training is initialized on each of the clients individually, thus generating a local model at the end;
3) Once all the clients have sent their weights, the Global model is generated by the server using the arithmetic mean of all of them;
4) The Global model is shared with each client;
5) Another round of training is started, and the whole process is repeated until N rounds have been completed.

F. The dataset

Effective training of a CV model requires a comprehensive and representative set of images. However, it is not easy to meet these requirements in the clinical area due to problems related to the sensitivity of this data type [18]. Other characteristics that make it difficult to obtain this kind of data are related to the high cost of creating these databases. Since it is necessary to take care of the privacy of the information and the labeling of the images [19].

The model development phase uses a dataset obtained from Kaggle [20], a data science community that offers free access to dozens of image databases. The dataset contains three classes, organized as follows: I. No disease, II. Bacterial pneumonia, and III. Viral pneumonia. The dataset selected contains 4672 labeled images, which by standard were separated into 80% for training and 20% for validation purposes. Therefore, 3738 random images are used for model training and 934 for validation. However, in the FL approach, it is still necessary to separate the dataset for the devices to simulate scenarios that are closer to the real ones, where different image bases will be used for each of the Clients. Therefore, the image base was randomly separated into two smaller ones (2336 samples each), separating different images for each client.

The number of observations per class is distributed as follows: 1227 images belong to class I, 2238 images belong to

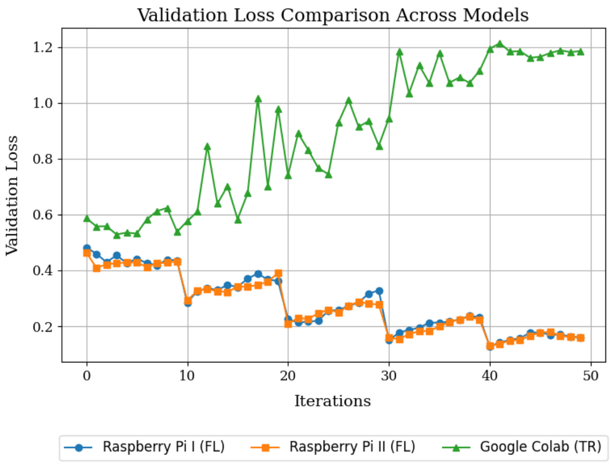

Fig. 1. Comparing the validation Loss presented by the SqueezeNet model in traditional (centralized) and FL approach.

Fig. 2. Comparing the training loss presented by the SqueezeNet model in traditional and federated approach.

class II, and 1207 images belong to class III. In this sense, the dataset used presents an imbalance between the classes and an unlabeled test set. However, class imbalance is common in medical scenarios, particularly when it is anticipated that only a small proportion of patients will ultimately receive a diagnosis of the disease in question. The dataset presented is used in both the centralized and federated approaches in an attempt to be as fair as possible when comparing the results presented.

IV. Results and Discussions

The results obtained in our experiments show several benefits of using the AIoMT approach. Figure 1 shows the validation loss of both architectures during 50 training epochs. It can be seen that using FL in this context avoided overfitting the model. The blue and orange curves show the validation loss of each Raspberry Pi device. The weights of both local models are sent to the central aggregator every 10 epochs. Therefore, after every 10 epochs, as it is noticeable, the loss value suffers a visible drop. It shows that the aggregated model is better than the local ones, achieving a higher generalization capacity than the individual models. However, the results also show that the local models exhibit an overfitting tendency between model aggregations, which is refrained by aggregating the local models' gradients. Figure 2 shows the training loss. As is visible, after around 15 epochs, the loss of the centralized model becomes lower than that of the Raspberry Pi devices, which, as supported by the validation loss, shows the centralized model is overfitting.

Sharing the gradients in each aggregation round shows a significant performance improvement compared to the model residuals for both devices. Some points to note are that the traditional, i.e., centralized, method obtained a lower loss in the training set, but the loss presented in the validation set shows a big difference between the methods. Once again, the FL contribution to improving the error rate of the final model is visible. The gap between the two models' final error rates is also outstanding. In this respect, the FL has much to

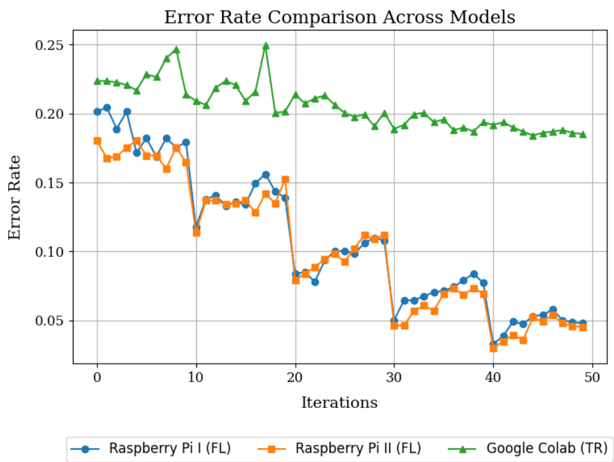

Fig. 3. Comparing the Validation Error Rate presented by SqueezeNet model in traditional and FL approach.

offer when training the distributed model. Figure 3 shows this phenomenon.

In this way, it is possible to see that the model trained using the FL had a higher generalization capacity. As the main objective in training a model is to maximize its ability to generalize, the final model of the FL method is better. The Table II shows the best values obtained.

The accuracy shown by the model using FL is much higher than that obtained by the centralized approach. Once again, the advantage of aggregation is twofold: it improves the generalization capacity of aggregated models and avoids

TABLE II

Comparison between the results presented by the models in the centralized and federated approaches.

Training place	**Train loss**	**Validation loss**	**Error rate (%)**	**Accuracy (%)**
Raspberry I	0.0579	**0.1598**	4.7832	95.2168
Raspberry II	0.0632	0.1604	**4.5116**	**95.4884**
Centralized	**0.0051**	1.1854	18.4971	81.5029

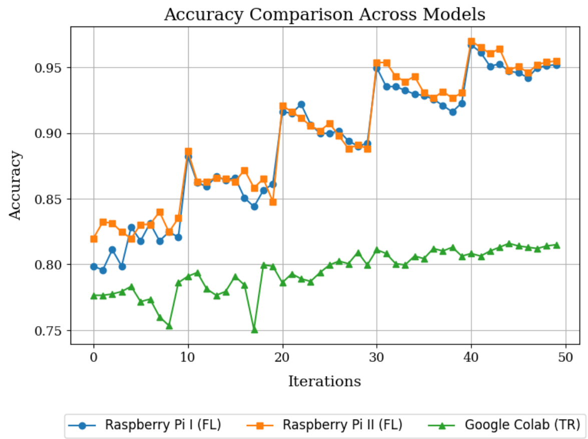

Fig. 4. Comparing the validation accuracy presented by the SqueezeNet model in traditional and federated approach.

overfitting local models. The accuracy of the models is shown in Figure 4.

V. Conclusions

This study proposed a solution with applications in low-income and remote areas with constrained devices. It uses embedded devices supported by FL to train compact and efficient local models. The study compares centralized and FL-based training approaches using the SqueezeNet V1.0 neural network architecture to classify pneumonia. Our results show significant improvements in metrics when comparing FL training to a centralized approach. The results also indicate that a better model generalization capacity is achieved using FL, surpassing the centrally trained model's results. Therefore, the study offers a simple and cost-effective solution for pneumonia detection in remote and low-income environments.

Future research directions could tackle the problems and assess performance improvements when increasing the number of embedded devices used to create the aggregated model. This would probably attain more accurate models, making diagnosis more accurate. Moreover, the proposed solution could be integrated into smartphones and computers with interactive and easy-to-use user interfaces, making it even easier to use in remote locations. Another direction would be comparing different model architectures (such as MobileNetV1, MobileNetV2, and even Squeezenet V1.1) and approaches to dealing with gradient aggregation and class unbalancing, such as data augmentation, over/under-sampling, and generative models. Also, the environment worked on during the development of the application does not point to possible adverse scenarios such as connection problems, network latency, or lower capacity networks. Therefore, future research should be carried out to verify the application's behavior in these scenarios. Finally, a study of the network's behavior during training should be carried out, analyzing in detail how the application impacts the network and how it behaves during execution.

Acknowledgments

This work was partially funded by CNPq (Grant Nos. 403612/2020-9, 311470/2021-1, and 403827/2021-3), by Minas Gerais Research Foundation (FAPEMIG) (Grant No. APQ-00810-21) and by the project XGM-AFCCT-2024-2-5-1 supported by xGMobile – EMBRAPII-Inatel Competence Center on 5G and 6G Networks, with financial resources from the PPI IoT/Manufatura 4.0 from MCTI grant number 052/2023, signed with EMBRAPII.

References

[1] M. W. Condry and X. I. Quan, "Digital health innovation, informatics opportunity, and challenges," *IEEE Engineering Management Review*, vol. 49, no. 2, pp. 81–88, 2021.

[2] A. Hosny, C. Parmar, J. Quackenbush, L. H. Schwartz, and H. J. Aerts, "Artificial intelligence in radiology," *Nature Reviews Cancer*, vol. 18, no. 8, pp. 500–510, 2018.

[3] B. H. da Silva, F. S. Franco, and A. M. Tannús, "Tecnologia iot na área da saúde," 2022.

[4] K. Kakhi, R. Alizadehsani, H. D. Kabir, A. Khosravi, S. Nahavandi, and U. R. Acharya, "The internet of medical things and artificial intelligence: trends, challenges, and opportunities," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 3, pp. 749–771, 2022.

[5] D. Ng, X. Lan, M. M.-S. Yao, W. P. Chan, and M. Feng, "Federated learning: a collaborative effort to achieve better medical imaging models for individual sites that have small labelled datasets," *Quantitative Imaging in Medicine and Surgery*, vol. 11, no. 2, p. 852, 2021.

[6] C. Shiranthika, P. Saeedi, and I. V. Bajić, "Decentralized learning in healthcare: A review of emerging techniques," *IEEE Access*, vol. 11, pp. 54188–54209, 2023.

[7] N. Ferraz Junior, *FedSensor: framework de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos.* PhD thesis, Universidade de São Paulo, 2022.

[8] C.-L. Kao, L.-C. Chien, M.-C. Wang, J.-S. Tang, P.-C. Huang, C.-C. Chuang, and C.-L. Shih, "The development of new remote technologies in disaster medicine education: A scoping review," *Frontiers in public health*, vol. 11, p. 1029558, 2023.

[9] T. B. N. Pham and M. H. Nguyen, "Classification of pneumonia on chest x-ray image using transfer learning," tech. rep., EasyChair, 2023.

[10] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE access*, vol. 6, pp. 64270–64277, 2018.

[11] B. Kakkar, P. Johri, Y. Kumar, H. Park, Y. Son, and J. Shafi, "An iomt-based federated and deep transfer learning approach to the detection of diverse chest diseases using chest x-rays," *Hum Centric Comput Inf Sci*, vol. 12, p. 24, 2022.

[12] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016.

[13] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and lt;0.5mb model size," 2016.

[15] J. Howard and S. Gugger, "Fastai: a layered api for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.

[16] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, *et al.*, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[17] M. A. d. Silva Neto, "grpc: a tecnologia de chamada de procedimento remoto do google para arquiteturas distribuídas,"

[18] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pp. 1–11, 2019.

[19] X. Zhao, W. Liang, and J. Zou, "Data budgeting for machine learning," 2022.

[20] K. Diamantaras, "Chest x-rays: Bacterial/viral pneumonia/normal," 2020.