Federated Learning Applied to Arrhythmia Detection on Electrocardiograms

Rodrigo S. Couto* and Lucas C. Favaro *
* Universidade Federal do Rio de Janeiro - PEE/COPPE/GTA - Rio de Janeiro, RJ, Brazil
Email: rodrigo@gta.ufrj.br, favaro@gta.ufrj.br

***Abstract*—Federated learning is an approach to training machine learning models without the dependency on centralized data, ensuring privacy and security for data owners. This approach can be helpful in applications based on medical data, such as electrocardiograms used to identify arrhythmias. This work analyses the feasibility of federated learning in a distributed model of arrhythmia detection based on Convolutional Neural Networks (CNNs). We employ a model based on implementations present in the literature, using its centralized and distributed versions trained on the MIT-BIH dataset. Our results show that the distributed model achieves a recall of 83.14% and a precision of 58.47% in the ideal scenario. Considering a non-independent and non-identically distributed (non-IID) scenario, the model achieves a recall of 85.1% and a precision of 57.34%. For comparison, the centralized model achieves a recall of 82.01% and a precision of 53.03%. Consequently, these results indicate that federated learning is a feasible option for developing a distributed model for arrhythmia detection.**

I. Introduction

Machine learning models, especially deep learning ones, generally require a large amount of data to perform well [1]. However, the data can be sensitive or private, such as medical or personal data. Hence, data owners resist providing their data for training and improving machine learning models [2].

Traditionally, machine learning employs centralized techniques in which client data is sent to a central server. The server uses the data to train a model. These centralized models have access to all client data, which can be a problem in the case of private and sensitive data. Distributed learning techniques can help solve this problem. In this case, client data is not sent directly to the central server. In general, each client performs computations locally, and only the result of these computations is sent to the server. In this context, Ramage *et al.* proposed Federated Learning [3]. This approach keeps data distributed across client devices and uses a shared model that aggregates information on a centralized server. In federated learning, each client receives the current version of the global model parameters from the server and improves them locally using its own data for training. Then, the clients send the local model's parameters to the server, which aggregates them, generates a shared global model and send its parameters again to the clients. This procedure is performed for several rounds to increase model performance while keeping data privacy.

The detection of cardiac arrhythmias is an area that can benefit from federated learning [4], [5]. An arrhythmia is a change in heart rate that can indicate conditions such as myocardial infarction, ventricular tachycardia, and atrial fibrillation. Usually observed by highly specialized professionals, these changes can be identified through neural networks [6], [7]. As medical data is private, federated learning can ensure the security and privacy of the data provided by clients while providing a global model that can be used to detect arrhythmias efficiently.

With the adoption of federated learning, data may not be independent and identically distributed (non-IID) among the clients. For example, considering each client is a hospital in a given country, the proportion of patients with arrhythmia may be different among their populations. The scenario in which the data is non-IID presents a challenge for using federated learning. In this case, there may be a performance loss if the parameter aggregation algorithm does not consider the difference in data distribution between clients [5], [8].

This work seeks to analyze the feasibility of federated learning applied to an arrhythmia detection model in electrocardiograms. Therefore, the main advantages and challenges of using such an approach are analyzed, exploring different scenarios of data distribution among clients, emphasizing good practices for working with such data in a federated scenario.

The model for arrhythmia detection utilized in this work was developed based on convolutional neural networks (CNNs) models presented in the literature [6], [7], [9] based on the recommendations of the Association for the Advancement of Medical Instrumentation (AAMI) for the development of computational models for arrhythmia classification [10].

The MIT-BIH Arrhythmia Database (MIT-BIH Arrhythmia Database) [11] was used for training and evaluating the model, and a new division of the dataset was proposed, in order to avoid data from the same patient being used for training and testing at the same time.

In the ideal case, where the data is independent and identically distributed (IID), the model based on federated learning showed recall of 83.14% and a precision of 58.47%. In the non-IID scenario, the recall was 85.1% and the precision was 53.03%. The centralized model, used for comparison, showed recall of 82.01% and a precision of 57.34%. It should be noted that the model proposed in this paper considers ideal network conditions, in which all clients are always available to the server and no communication failures occur.

This paper is organized as follows. Section II gives an overview on federated learning. Section III reviews related work. Then, Section IV introduces the methodology, while Section V presents the obtained results. Finally, Section VI presents the conclusion and next steps.

II. Federated Learning

One of the most used algorithms for federated learning is Federated Averaging, which is an extension of a federated version of SGD (Stochastic Gradient Descent) called Federated SGD [2]. In Federated SGD, a fraction C of clients is selected in each round of communication. Each selected k client computes the gradient, $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$, from the loss function $J(\boldsymbol{\theta})$ [2]. Each client sends this computed gradient to the central server, which aggregates them through the equation:

$$w_{t+1} = w_t - \alpha \sum_{k=1}^{K} \frac{n_k}{n} \nabla J_k(w_t), \tag{1}$$

where w represents the model parameters, n is the size of the whole dataset, K is the number of selected clients, n_k is the size of the client-k's local dataset, and α is the learning rate.

An equivalent way of updating the parameters is achieved with the clients performing a SGD (or GD) step and the server aggregating the results via a weighted average [2]:

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k}, \tag{2}$$

where

$$w_{t+1}^{k} = w_t - \alpha \nabla J_k(w_t). \tag{3}$$

With the aggregation algorithm written this way, it is possible to perform more than one training step locally before sending the weights to the server [2]. This algorithm is called Federated Averaging (FedAvg) and has three main parameters: the fraction of clients selected at each round C; the number of local epochs E; and the size B of the minibatch, used for updating the client parameters.

III. Related Work

With the dissemination of the first public arrhythmia databases, such as MIT-BIH, several works started to be developed [11]. However, as observed by Luz *et al.*, many of these works use data from the same patient in the training and in the test sets at the same time [12]. This is inconsistent with real-world situations, since data from new patients would never have been seen by a detector. Many of the works done this way have accuracy above 95% [13]. Chazal *et al.* [14] proposes a split in the dataset to separate patients for training and test, resulting in a more realistic scenario.

Many works on arrhythmia classification or detection, use Convolutional Neural Networks (CNNs) [6], [7], [15]. Hannun *et al.* [7], for example, proposes a network of 34 layers, three of which are convolutional ones. The authors collected 64,121 electrocardiogram recordings from 29,163 patients. A committee of cardiologists performed annotations on each recording. The authors split the dataset so that no intersection of patients occurred between the training and test sets. And, for evaluation, another committee of cardiologists classified the beats, to serve as a comparison for the classifier results. The proposed model obtained an accuracy of 80.9% and outperformed the independent cardiologists (75.1%).

Due to the advantages of federated learning for medical data, some works incorporate it into the task of arrhythmia classification. In the proposal of Sakib *et al.* [4], the devices on edge nodes receive individual client data, perform local training and share the parameters with the central server and the other edge nodes. The authors split the MIT-BIH dataset in a way that no intersection occurs between the training and test data. The case of non-IID data among clients is not explored.

Gao *et al.* [9] perform an evaluation of federated learning applied to Internet of Things (IoT) devices. The evaluation uses a one-dimensional CNN and the arrhythmia detection problem as a case study. In the paper, IID and non-IID cases are evaluated. Meanwhile, the data is divided into the test and training sets in a randomized manner. This division does not necessarily guarantee that the patients in the training set are different from those in the test set.

Zhang *et al.* [5] proposes a strategy to optimize federated learning for non-IID data. In this strategy, a fraction of client data is shared with the server to form a global data distribution. The server performs training rounds and sends the updated weights to the clients. From their local data, the clients repeat the same process and send the updated weights to the server. This strategy, according to the authors, does not hurt the privacy of federated learning because the server is considered a trusted entity [5]. However, since a fraction of clients' data is shared with the central server, privacy is partially lost concerning the server, reducing one of the main advantages of using federated learning.

Ma *et al.* [8] propose a federated learning technique combined with feature alignment. In their proposal, the local training uses a feature alignment module, in which CNNs are used to extract the features from the global model and the local model. The final model for each client is generated in a way that minimizes the distance between the global and local features. Overall, the technique aims to decrease the impact of the differences between the local and global datasets.

Raza *et al.* [16], use transfer learning and Explainable Artificial Intelligence to build a model that ensures data privacy and security in the IoT context. The model consists of two modules: the federated learning module and the Explainable AI module. CNNs are used in the federated module to classify electrocardiogram data. This module uses transfer learning as a way to optimize the learning process. Then, the Explainable AI module is used to facilitate the interpretation of the classifier results and expedite physician decision-making. The authors use Raspberry Pi devices for testing with the MIT-BIH database and highlight the impact of data unbalance on the classification problem [16].

Our work differs from the others by analyzing the feasibility of federated learning for arrhythmia detection with no intersection between training and test sets and considering the non-IID case. Also, we employ a public database. Hence, a new division of MIT-BIH is proposed to avoid the intersection of patients in the test and training sets. Table I summarizes the difference between this work and the others in the literature.

TABLE I
RELATED WORK INVOLVING FEDERATED LEARNING

Paper	**Public Database**	**Training and Test sets with no intersection**	**Non-IID Experiment**
Sakib *et al.* [4]	Yes	Yes	No
Gao *et al.* [9]	Yes	No	Yes
Zhang *et al.* [5]	Not informed	No	Yes
Ma *et al.* [8]	No	Not informed	Yes
Raza *et al.* [16]	Yes	No	No
This work.	Yes	Yes	Yes

IV. METHODOLOGY

This work uses the MIT-BIH Arrhythmia Database [11], developed with electrocardiogram data from 47 patients, collected between 1975 and 1979. The dataset has 48 recordings of approximately 30 minutes, sampled at 360 Hz. From these recordings, 23 were generated by randomly choosing patient samples collected at Boston's Beth Israel Hospital, to provide waveforms that routinely appear in clinical examinations. These 23 recordings are provided in files numbered 100 to 124. The other 25 recordings were generated by choosing patients from the hospital whose samples contain clinically rare but important arrhythmia phenomena and are provided in files numbered 200 through 234. The beats in each recording were found by detecting the R-peaks of the QRS complex [17]. The QRS complex is a combination of 3 deflections present in an ordinary ECG that allows you to graphically identify abnormalities in the ECG. The R-peak is the point with the maximum amplitude of the QRS complex. Each peak was classified by two cardiologists.

AAMI developed a standard [10] for experiments involving the classification of arrhythmias by computer models. This standard aims to provide a "common ground" for different works in the field and also aims to minimize errors and biases. In this standard, it is recommended to use the MIT-BIH database, excluding 4 recordings that have problems in the representation of the beats. In addition, AAMI recommends that training and testing sets be created in such a way that samples from the same patient are not used simultaneously for training and testing [10].

In this work, beat signals are divided into two classes. Class "0" represents the beats considered normal and class "1" represents arrhythmias. The dataset is extremely unbalanced since about 90% of the beats belong to the "0" class. For each recording, the beats were represented using a 2-second window centered on each R-peak. The beat signals were passed through a low-pass filter to remove unwanted noise above 60 Hz. Each signal was normalized using min-max normalization.

Chazal *et al.* [14] propose to split the dataset into the DS1 and DS2 sets. This division uses data from one patient only for training or testing (i.e. never both simultaneously). Also, it takes into account the distribution of the classes, trying to maintain a balance between sets DS1 and DS2.

The split proposed by Chazal *et. al* [14] has 54% of the data in DS1, used for training, and 46% of the data in DS2, used for testing. To increase the amount of data available for training, we propose splitting the dataset into the DS3 and DS4 sets. Table II presents the sets DS3 and DS4 and their associated files, according to the numbering of MIT-BIH. This proposed split seeks to replace DS1 and DS2 so that 75% of the dataset is used for training and the rest for testing. As with the splif from Chazal *et al.*, the recordings are arranged into sets DS3 and DS4 to keep the distribution of classes close to the distribution of the original dataset.

TABLE II
RECORDINGS PRESENT IN DS3 AND DS4 SETS.

DS3						DS4			
100	103	101	106	108	109	105	114	116	124
112	115	118	111	113	117	200	203	209	213
121	123	119	122	201	205	214	222	231	234
207	208	215	220	223	230				
202	210	212	232	233	228				
219	221								

Since the dataset is extremely unbalanced between classes "0" and "1" the technique of random oversampling was applied to the class 1 samples of the training set (DS3) to circumvent the unbalancing problem.

The neural network proposed in this paper has two one-dimensional convolutional layers with kernel of size 21, plus leaky RELU activation layers, dropout layers, pooling layers of size two, a batch normalization layer and two fully connected layers. The dropout and the batch normalization layers are used as a form of regularization. L1 regularization and the weight decay parameter are also used. The loss function is cross entropy combined with a sigmoid layer. The optimizer is SGD with a 0.9 momentum [18]. The learning rate, chosen after empirical tests, is 0.01. The centralized model serves as the basis for comparisons of the results of the distributed cases. Local training was performed for 100 epochs and with batches of sizes 16, 128 and 512.

The experiments with distributed models are divided into two cases. In the ideal case (IID), all clients have the same data distribution. In the other case, the clients have different data distributions from each other (non-IID). For the IID case, the data from the DS3 set was distributed equally among the clients so that they all had the same distribution. For the non-IID case, the beats from the DS3 set were distributed among the clients by patients so that each client has a different distribution of data. In both scenarios, the tests are implemented using five independent clients that communicate

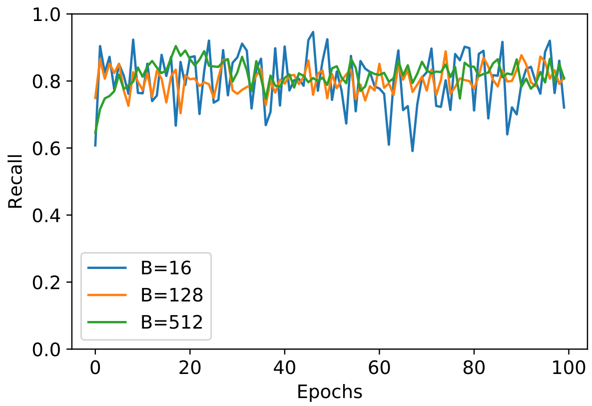

Fig. 1. Recall for the centralized model.

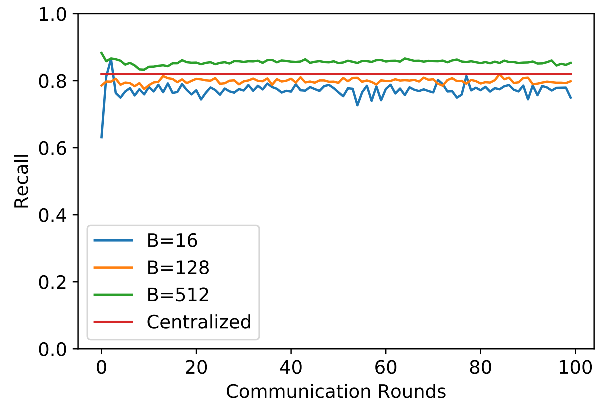

Fig. 2. Recall for the distributed IID case.

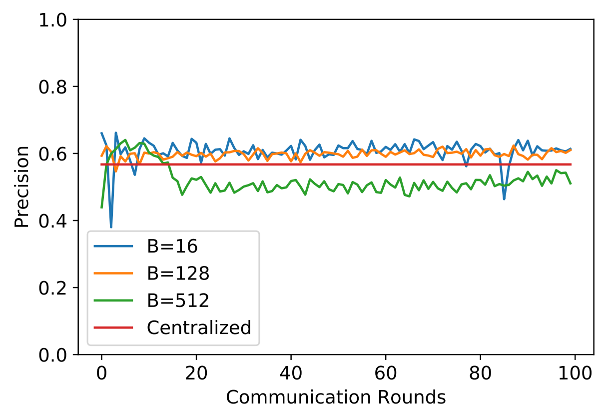

Fig. 3. Precision for the distributed IID case.

with a central server considering optimal network conditions.

The clients perform training for 10 local epochs and send the obtained parameters to the server. Each iteration of this process is called a communication round. In all cases, 100 communication rounds are performed between clients and the server. We employ the federated learning algorithm FedAvg, available through the Flower [19] library. All experiments use Python [20], with the libraries PyTorch [21], Scikit-learn [22], Pandas [23], Numpy [24], and Flower.

V. Results

Table III shows the recall and precision values obtained for each batch size for the centralized model. The results show that the performance improves as the size of the batch increases. This effect is observed in other works involving CNNs [25]. However, an increase in the size of the batch results in a longer training time. Thus, it is necessary to use the most suitable batch be used for the available computational infrastructure.

TABLE III
Results for the centralized model.

Batch	Recall	Precision
16	80.37%	43.37%
128	80.29%	58.03%
512	82.01%	57.34%

Figure 1 presents the recall obtained over the epochs, for the centralized model. The precision curves present similar behavior and have been omitted for conciseness. Each curve represents the results obtained for a different batch size “B”. The figure shows that a smaller batch results in more noise in the obtained recall. This behavior is expected in SGD, since, for a smaller batch, there is more noise in the computation and more steps are required to minimize the cost function. Even for the largest batches, the result obtained is noisy and the model has difficulty reaching convergence. It can be explained by the large unbalance between the classes of the problem.

Figure 2 presents the recall results obtained with ten local epochs per client for the distributed model in the IID case. We also plot the recall with a batch of size 512 from Table III for comparison. The figure shows that the recall values with the larger batches are better than that the ones with the centralized model. This behavior indicates that a side-effect of federated learning can be to improve the model’s performance. This indicates that federated learning may result in a regularizing effect due to the aggregation of the weights.

Figure 3 exposes the precision or the IID case. We also plot the centralized value for a batch size of 512, from Table III, as a reference. Comparing this figure with Figure 2, we can note a tradeoff between recall and precision. This tradeoff is more evident in the larger batches. Note that, different from the centralized model, a large batch size leads to a drop in precision as the number of rounds increases. One explanation for this effect is that a larger batch size may require adjustments to the learning rate to perform as expected. This effect may indicate that there is a given number of rounds that results in the best model performance. Hence, an interesting research direction is to propose a federated learning algorithm to explore this tradeoff.

For the scenario in which the data is non-IID, the recall, precision results are displayed in Figures 4 and 5, respectively.

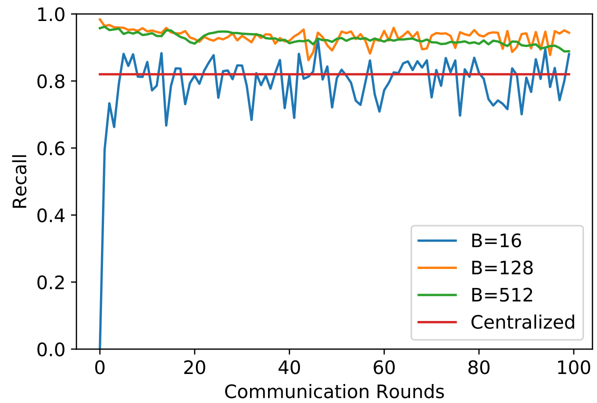

Fig. 4. Recall for the distributed non-IID case.

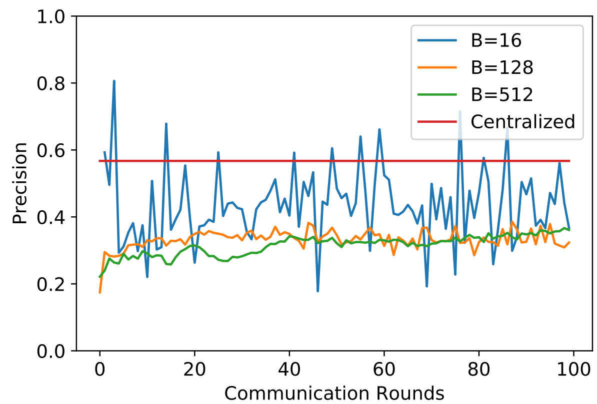

Fig. 5. Precision for the distributed non-IID case.

The effects observed in the ideal scenario occur with greater intensity in this scenario: noise is larger with smaller batch size and there is a tradeoff between recall and precision. Thus, despite the better results for recall, the precision drops dramatically when compared to the IID case. This behavior can be explained by the fact that the non-identical distribution between clients causes the parameters not to converge to the optimal values. This behavior occurs since the aggregation algorithm, FedAvg, does not take into account how far the distribution of a client is from the overall distribution of the data. Furthermore, the oversampling of the minority class has a negative effect compared to the centralized case and the IID federated scenario. Since some clients do not have enough arrhythmia samples compared to the global data, oversampling on these clients results in a local model that suffers from overfitting and that undermines the global model.

Figures 6 and 7 show the recall and precision results for the original and the reduced oversampling rate, with a batch size of 512. The original rate replicates the minority samples by a factor of 9, while the reduced rate replicates by a factor of 4. The behavior observed in this experiment indicates that adaptations to the local model may be needed to ensure the best result from the global model considering the balance between recall and precision. In a scenario in which the global statistics of the problem are known, such as the problem of the incidence of arrhythmias in a population, these small changes to the local model are feasible because one has *a priori* knowledge of the statistics of the global data, even without direct access to the data itself.

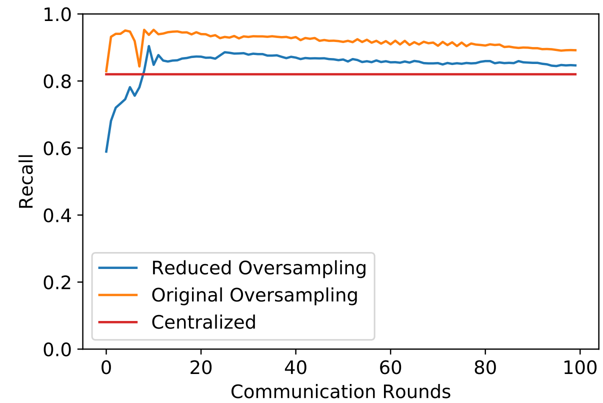

Fig. 6. Recall for the distributed non-IID case with a 512 batch size, using different oversampling rates.

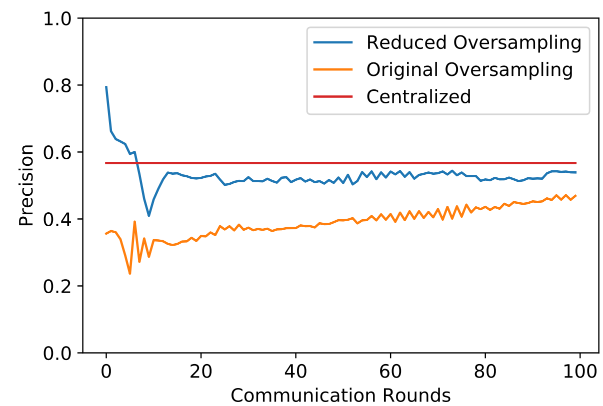

Fig. 7. Precision for the distributed non-IID case with a 512 batch size, using different oversampling rates.

VI. Conclusions

In this work, we have analyzed the feasibility of a distributed model for arrhythmia detection using federated learning. We have performed experiments with the MIT-BIH dataset to consider the ideal case where the data is IID and the more realistic case with non-IID data. The centralized model was used as a reference for comparisons.

With the results presented in this paper, it is possible to conclude that it is feasible to develop a computational model based on federated learning for the identification of arrhythmias in electrocardiograms. In the ideal scenario, the distributed model has presented compatible and even better results than the centralized model. This indicates that, in some cases, federated learning can have the side effect to increase detection's performance.

The non-IID scenario represents a challenge in the development of federated learning-based arrhythmia detection models. This challenge is because clients with data distributed in very different ways lead to a model that does not fit the global distribution of the data well. However, even in this scenario, small changes in the local models, such as reducing oversampling, generate results comparable to those obtained in the IID scenario. Furthermore, such problems can be solved using other aggregation algorithms, such as the federated version of adaptive optimizers [26].

In a nutshell, one of the challenges of federated learning lies in using aggregation algorithms that enable better performance in non-IID data scenarios. Thus, performance analysis and the development of other algorithms that perform better in non-IID scenarios is an interesting direction for future work. In addition, the development of strategies that adjust the clients' local models independently is another possible direction to provide a better global model.

ACKNOWLEDGEMENT

This study was financed in part by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, CNPq, PR2/UFRJ, FAPERJ Grants E-26/203.211/2017, E-26/010.002174/2019, and E-26/201.300/2021, and FAPESP Grant 15/24494-8.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, 2015.

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: http://arxiv.org/abs/1602.05629

[3] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

[4] S. Sakib, M. M. Fouda, Z. Md Fadlullah, K. Abualsaud, E. Yaacoub, and M. Guizani, "Asynchronous federated learning-based ecg analysis for arrhythmia detection," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 277–282.

[5] M. Zhang, Y. Wang, and T. Luo, "Federated learning for arrhythmia detection of non-iid ecg," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1176–1180.

[6] D. Li, J. Zhang, Q. Zhang, and X. Wei, "Classification of ecg signals based on 1d convolution neural network," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–6.

[7] H. AY, R. P, H. M, T. GH, B. C, T. MP, and N. AY, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nat Med*, vol. 25, pp. 65–69, 2019.

[8] L. Ma, R. Tang, J. Luo, J. Qian, and J. Jin, "Personalized federated learning for ecg classification based on feature alignment," *Security and Communication Networks*, vol. 2021, 2021.

[9] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtep, H. Kim, and S. Nepal, "End-to-end evaluation of federated learning and split learning for internet of things," in *2020 International Symposium on Reliable Distributed Systems (SRDS)*, 2020, pp. 91–100.

[10] A. AAMI, "Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms," *American National Standards Institute, Inc. (ANSI), ANSI/AAMI/ISO*, vol. EC57, 1998-(R)2008, 2008.

[11] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[12] E. J. da S. Luz, T. M. Nunes, V. H. C. de Albuquerque, J. P. Papa, and D. Menotti, "Ecg arrhythmia classification based on optimum-path forest," *Expert Systems with Applications*, vol. 40, no. 9, pp. 3561–3573, 2013.

[13] C. Ye, M. Coimbra, and B. Kumar, "Arrhythmia detection and classification using morphological and dynamic features of ecg signals," *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, vol. 2010, pp. 1918–21, 08 2010.

[14] P. de Chazal, M. O'Dwyer, and R. Reilly, "Automatic classification of heartbeats using ecg morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.

[15] A. Kurniawan, Ananda, F. N. Pradanggapasti, R. F. Rachmadi, E. Setijadi, E. M. Yuniarno, M. Yusuf, and I. K. E. Purnama, "Arrhythmia classification on electrocardiogram signal using convolution neural network based on frequency spectrum," in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 2020, pp. 29–33.

[16] A. Raza, K. P. Tran, L. Koehl, and S. Li, "Designing ecg monitoring healthcare system with federated transfer learning and explainable ai," *Knowledge-Based Systems*, vol. 236, p. 107763, 2022.

[17] A. R. Perez-Riera, L. C. de Abreu, R. Barbosa-Barros, K. C. Nikus, and A. Baranchuk, "R-peak time: An electrocardiographic parameter with multiple clinical applications," *Annals of noninvasive electrocardiology : the official journal of the International Society for Holter and Noninvasive Electrocardiology, Inc*, vol. 21, no. 1, pp. 9–10, 2016.

[18] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1139–1147.

[19] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *CoRR*, vol. abs/2007.14390, 2020. [Online]. Available: https://arxiv.org/abs/2007.14390

[20] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[24] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020.

[25] P. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, pp. 20–24, 12 2017.

[26] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *CoRR*, vol. abs/2003.00295, 2020. [Online]. Available: https://arxiv.org/abs/2003.00295