This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/MeditCom49071.2021.9647636

Asynchronous Federated Learning-based ECG Analysis for Arrhythmia Detection

Sadman Sakib*[1], Mostafa M. Fouda†[2], Zubair Md Fadlullah*‡[3], Khalid Abualsaud§[4], Elias Yaacoub§[5], and Mohsen Guizani§[6].
*Department of Computer Science, Lakehead University, Thunder Bay, Ontario, Canada.
†Department of Electrical and Computer Engineering, Idaho State University, Pocatello, ID, USA.
‡Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada.
§Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar.
Emails: [1]ssak2921@lakeheadu.ca, [2]mfouda@ieee.org, [3]zubair.fadlullah@lakeheadu.ca, [4]k.abualsaud@qu.edu.qa, [5]eliasy@ieee.org, [6]mguizani@ieee.org.

***Abstract*—With the rapid elevation of technologies such as the Internet of Things (IoT) and Artificial Intelligence (AI), the traditional cloud analytics-based approach is not suitable for a long time and secure health monitoring and lacks online learning capability. The privacy issues of the acquired health data of the subjects have also arisen much concern in the cloud analytics approach. To establish a proof-of-concept, we have considered a critical use-case of cardiac activity monitoring by detecting arrhythmia from analyzing Electrocardiogram (ECG). We have investigated two Federated Learning (FL) architectures for arrhythmia classification utilizing the private ECG data acquired within each smart logic-in-sensor, deployed at the Ultra-Edge Nodes (UENs). The envisioned paradigm allows privacy-preservation as well as the ability to accomplish online knowledge sharing by performing localized and distributed learning in a lightweight manner. Our proposed federated learning architecture for ECG analysis is further customized by asynchronously updating the shallow and deep model parameters of a custom Convolutional Neural Network (CNN)-based lightweight AI model to minimize valuable communication bandwidth consumption. The performance and generalization abilities of the proposed system are assessed by considering multiple heartbeats classes, employing four different publicly available datasets. The experimental results demonstrate that the proposed asynchronous federated learning (Async-FL) approach can achieve encouraging classification efficiency while also ensuring privacy, adaptability to different subjects, and minimizing the network bandwidth consumption.**

***Index Terms*—ECG data, federated learning, arrhythmia, IoT.**

I. Introduction

The Internet of Things (IoT) is apprehended to be an indispensable enabler of the next generation smart society. With the growing demand for remote health monitoring, the need for a paradigm shift from Artificial Intelligence (AI)-aided centralized remote cloud computing toward edge analytics for biomedical devices is rapidly growing [1]. Conventionally, the IoT devices and wearables have been utilized to collect and trace various well-being indicators such as Electrocardiogram (ECG), Electroencephalogram (EEG), and so forth. For implementing remote health monitoring, this traditional cloud-based analytics paradigm of regular IoT monitors contributes to prolonged delay, massive bandwidth consumption, and privacy concerns associated with the user's health data. In recent research, these challenges of traditional analytics were addressed and overcame with the emergence of a lightweight AI model

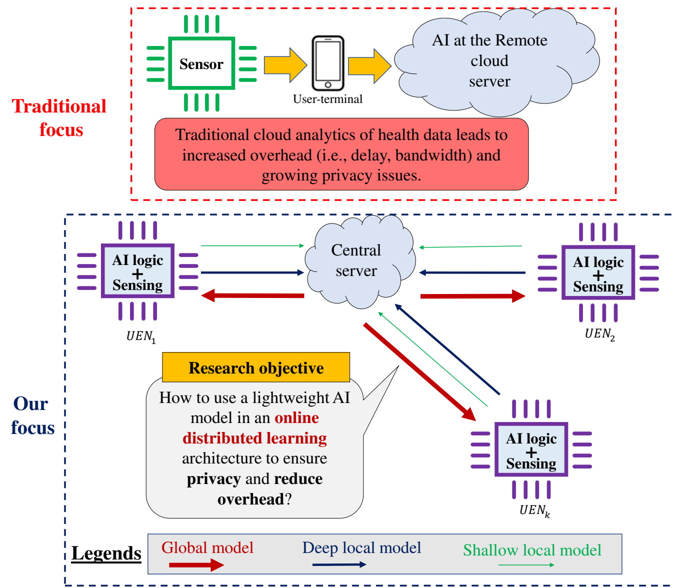


Fig. 1: Our main focus is to develop an asynchronously federated learning-based ECG analytic methodology at the distributed ultra-edge nodes (UENs) to classify irregular heartbeats while preserving patient-data privacy.

deployed at the ultra-edge IoT nodes [2]. However, the ultra-edge health monitoring system lacks online learning capability, which cannot be achieved without re-training. Hence, in this article, we discuss this compelling urgency of drafting a distributed online learning technique (i.e., Federated Learning) using the Ultra-Edge Node (UENs) as the participating users, as depicted in Fig. 1, to bring localized intelligence with data privacy to biomedical edge devices as well as ensuring the online learning capability. We can summarize the main contributions of the paper as follows:

- Firstly, as depicted in Fig. 1, we enunciate the inherent shortcoming of the traditional cloud analytics-based health data analysis and reveal the pressing need for deploying lightweight AI solutions in a collaborative, distributed, and online paradigm to move the AI analytics from the cloud to the ultra-edge nodes to facilitate uninterrupted remote health monitoring with enhanced privacy.
- Among a diverse set of use-cases, we have chosen the use-

case of cardiac arrhythmia monitoring by analyzing ECG data. Arrhythmia is a significant contributor to cardiovascular diseases (CVD). Despite various technological advancements in the healthcare system, cardiovascular disease such as arrhythmia serves a noteworthy global public health predicament. It is still the most prominent life-threatening disease worldwide, with 15–20% of global mortalities [3]. According to American Heart Association (AHA), CVD costs will rise to approximately $750 billion by 2035. However, due to the privacy-sensitive of the ECG and the complexities of collecting ECG, obtaining a significant quantity of ECG to train a centralized machine learning (ML)-based model of arrhythmia detection is quite complicated and even not feasible under some circumstances such as remote monitoring of subjects. To ensure remote and secure cardiac monitoring in a lightweight manner with online learning ability, we employ two variations of the federated learning paradigm: asynchronous federated learning (referred to as Async-FL) and synchronous federated learning (referred to as Sync-FL). We propose the Async-FL as the most suitable distributed learning architecture for ECG analysis. It ensures less overhead and can obtain identical classification efficiency compared to the Sync-FL.

- As depicted in the second part of Fig. 1, our focus is to develop an agile data acquisition system for ECG for arrhythmia detection with mobile and deployable ultra-edge nodes (UENs) acting as edge computing nodes. A UEN can be considered a sensing node with a localized AI model to collect and gain knowledge from the subject's ECG data. As the AI model, we have used a (1-D) convolutional neural network (CNN)-based deep learning model (DL-LAC model) previously designed in one of our earlier papers [2]. We had presented that the proposed model can be used to classify heartbeats employing raw single-lead ECG. The AI model does not require any pre-processing (i.e., noise-filtering) of the ECG signal, making the system lightweight and easy to integrate with the ultra-edge node. To design a distributed learning setup, the model will be initially trained and deployed to the UENs; upon deploying, the model will be updated asynchronously using the Async-FL architecture. After updating its local AI model, the UEN shares the model parameters with neighboring UENs and the cloud by efficiently scheduling the shallow and deep parameters to reduce the communication overhead. By employing this unique concept of asynchronously updating AI model parameters (i.e., model weights) of the UENs, the privacy of the acquired medical data and network efficiency are jointly preserved.
- Rigorous experimental analysis of our proposed Async-FL approach is presented using four publicly open and clinically graded ECG datasets. The encouraging results of the proposed method illustrate its potential to implement the envisioned concept in remote cardiac activity monitoring as a practical solution. To the best of our knowledge, no existing research focuses on the distributed and online machine learning architecture for remote arrhythmia monitoring employing the ultra-edge IoT nodes and Async-FL.

The remainder of the paper is organized as follows. Section II surveys the relevant research work on combating arrhythmia with AI method. The formal problem is described in section III. Next, our considered asynchronously updating federated learning-based system model and proposed asynchronously updating federated learning algorithm is presented in section IV. The performance of our proposal is evaluated in section V. Finally, section VI concludes the paper.

II. Related Work

In cloud-based ECG monitoring systems, several techniques are utilized, including feature extraction and classification. Discrete Wavelet Transformation (DWT) and Artificial Neural Network (ANN) were adopted for feature extraction, and for binary heartbeat classification (normal or abnormal) are utilized in [4]. The authors in [5] trained a neural network (i.e., convolutional neural network) to classify 12 cardiac rhythm categories using the single-lead ECGs and achieved a higher classification efficiency than the traditional cardiologists. DWT and non-linear delay differential equations (DDE)-based optimization techniques [6], introduced in the literature for time-series ECG monitoring tasks, are unable to infer the system models in varying heart conditions adequately, as an exhaustive exploration is required to pick the most appropriate structure for the classification task in these approaches. For choosing an optimal DDE-based classification model, a Genetic Algorithm (GA) was applied in [7]. In another research [8], sparse decomposition was adopted for efficient feature extraction, and K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Radial Basis Function Neural Network were applied for classification. Different separate manual feature extraction techniques was used to determine features such as P-wave interval, QRS interval, and QT interval from 12-lead ECGs and applied the support vector machine model to detect myocardial infarction in [9]. However, most of the existing literature does not focus on designing lightweight AI models and utilizes a centralized ML algorithm to ignore the privacy leaks in collecting data. This means private ECG data must be collected and shared to train a data-driven ML model to detect arrhythmia events. Hence, these approaches are impractical in ultra-edge nodes due to their significantly high computational complexity.

The edge computing-focused ECG analytics was investigated in a few studies. Authors in [10] developed an ECG analysis algorithm with noise-filtering and manual feature extraction phases and implemented it on an IoT-based embedded platform. To press the need for collaborative online learning, the authors in [11] introduced a federated learning-based distributed algorithm that allows each medical hospital to participate in the AI model's training locally cooperatively. However, in the existing literature, the viability of the decentralized online federated learning technique for deployment in the ultra-edge nodes for remote ECG monitoring is still yet to be explored elaborately, and hence we focus on this research gap in this paper.

III. Problem Description

Cardiac arrests and arrhythmias are indirectly related to pandemics such as COVID-19 infection, especially among the

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/MeditCom49071.2021.9647636

more critical subjects [12]. Along with the rapidly changing variations of the COVID-19, their effect on cardiac activities needs to be monitored in a distributed manner to obtain the varying impact of different strains and a wide range of data distributions on the cardiac status. In rural areas where healthcare services are challenging to access for long-time cardiac monitoring, the UENs need to provide service coverage for subjects. The cardiac state (i.e., irregular heartbeats) of each subject can be obtained from each UEN. Multiple UENs can summarize a particular area so that the medical service providers can get an overall representation of the region. Thus, the initial challenge is to expedite distributed learning among the UENs to adapt and enhance itself with varying data distribution acquired from different UENs via the collaborative and online learning approach. Furthermore, there is a massive privacy requirement for the private data of the users as they are not willing to share the raw health-related data with a remote cloud server to avoid potential security threats.

Moreover, for each UEN, the private data should be utilized securely without transmitting raw data somewhere else and then removed upon local automatic decision making is completed. Thereby, the UENs need a distributed learning architecture to extract distinctive ECG features with enhanced privacy and learning efficiency. This task of decentralized or distributed collaborative learning can be a critical challenge because the UENs will not have the global status of all the other subjects. To resolve this challenge, we have investigated the different variations of federated learning techniques such as typical synchronous federated learning and asynchronous federated learning. The UENs who will act as the edge users in this collaborative learning scenario could also exchange knowledge gained from private data with the server to obtain global knowledge and ensure online learning capability. This approach will ensure data privacy, reduced latency, and adaptation ability to varying data distribution among diverse subjects. After receiving the local models from the UENs, the server will update the global model to obtain an optimized and efficient arrhythmia detection model. In other words, the purpose of loss function minimization can be observed as follows:

$$\min \sum_{i=1}^{n} \frac{\mathcal{X}_i}{\mathcal{X}} f_i(w), \tag{1}$$

where $f_i(w)$, $\mathcal{X}_i$, and $\mathcal{X}$ denote the loss function of UEN_i, the private ECG sample data of UEN_i, and the ECG data used by the cloud for training, respectively. The weight vector (w) of each UEN is denoted by w, indicating the parameters of the local AI model. With the increasing prediction error, the value of the loss function rises. Thereby, during the communication rounds in the learning phase, the constraint here is to ensure consistent weight parameters of the local AI models upon receiving the updated model from the server for learning convergence. The UENs and the server can utilize the global model for local decision-making without sharing raw ECG samples of each subject.

In summary, the research challenge is to devise a system so that each UEN is competent in training a distributed global model to update its local model by adopting its own private raw ECG data. Each UEN can also broadcast its updated local model to the neighboring UENs and the cloud server for asynchronous updates of the global model. This collaborative learning process will continue as long as the loss function is not minimized and the global model accuracy reaches a pre-defined performance threshold. Hence, in this paper, the primary research challenge is to develop such a decentralized learning architecture that will ensure effective decision-making (i.e., arrhythmia detection) with enhanced data privacy and minimize network overhead.

IV. System Design and Proposed Asynchronous Federated Learning-Based Algorithm

In this section, we describe the algorithm of the proposed method that contains Algorithms 1 and 2. The global and local AI models, constructed at the cloud and UENs, are facilitated by a customized lightweight CNN model. The deep learning model (referred to as Deep Learning-Based Lightweight Arrhythmia Classification, DL-LAC) was conceptualized in our earlier work in [2] for a centralized arrhythmia prediction at the ultra-edge sensors. The adopted lightweight AI model which will be deployed at the UENs for ECG analysis accepts a single-lead raw ECG heartbeat as input, represented as $X = [x_1, x_2, x_3, \dots x_n]$, and outputs predicted class labels $Y = [y_1, y_2, y_3, \dots y_k]$. Here, $y_k \in \{0, 1\}$, which indicates whether that particular heartbeat belongs to k_{th} class or not.

Algorithm 1 depicts the algorithm that takes place at the remote cloud server to update the global model. As input, the algorithm takes the set of local model parameters or weights of all UENs, denoted as W, and it delivers the updated global model (M_g). Steps 1 and 2 of this algorithm are the initialization phases. In steps 3 to 6, the transmitted local model parameters (W) of all UENs are aggregated to update the global model, M_g. The learning performance of the model is accessed in step 5, and during every iteration, it is compared with the previously defined loss threshold (ξ). Therefore, the overall learning process at the remote cloud server takes place as long as the current loss value is higher than ξ.

Algorithm 1: Local models' aggregation at the server

Input : W (Collection of local model parameters/weights of all UENs)
Output: M_g (global model)
1 $\xi \leftarrow$ define the minimum loss threshold
2 $M_g \leftarrow$ load the existing global model from storage
3 **while** $(curr_{loss} > \xi)$ **do**
4     $W \leftarrow$ load local model parameters of all UENs
      update M_g by aggregating all the local parameters (W)
5     $curr_{loss} \leftarrow$ compute current loss of M_g
6 **end**
7 return M_g

The algorithm that runs on the UEN's side is manifested in Algorithm 2. We have considered the algorithm that runs on one UEN (referred to as UEN j) in the Algorithm 2. The algorithm initiates with the inputs; these inputs' details are demonstrated in the algorithm's input fragment. First, in step 1, the local model of UEN j, M_j is initialized with the initially trained AI model. In steps 2 and 3, the initialization of different necessary parameters takes place. Step 4 loads the new private data, X_j, and its size by comparing it with the ECG data size threshold δ in step 5. The iteration starting at step 6 denotes that the overall training process at each UEN occurs from iterations $t = 1$ to $t = T$. The parameter T can be considered as the number of communication rounds during which the whole process at each UEN is executed. After Δ time rounds, when this condition is satisfied in step 7, time t is stored in $timestep_j$ list. Also, the local weight (W_j) with deep parameters and relevant access and time-step information ($block_j, timestep_j$) are communicated to the cloud in steps 9 and 10. Here, α denotes the deep parameter exchange ratio, indicating the deep parameter ratio contributing to the deep exchange. The parameter $timestep_j$ holds the iteration information when the deep parameter exchange with the cloud takes place. In steps 11 to 14, the local weight (W_j) shallow parameters of the $(1 - \alpha)$ ratio from the local model (M_j) are transferred to the server. Step 15 updates the local model of UEN j by using the aggregated model obtained from the server. In step 16, all the global model states and information regarding time-step t is stored in the $block_j$ for future reference. Finally, in step 18, after training for T rounds, the utilized data X_j are permanently removed from the cache to enhance the security of the user's data.

Algorithm 2: Learning at each UEN

Input : j (the current UEN), M_j (local model of UEN j), Δ (time-round), T (total number of communication rounds), α (deep parameter ratio, $0 < \alpha < 1$)

1 initialize M_j with the primarily trained AI model
2 $\delta \leftarrow$ initialize ECG data size threshold
3 initialize $timestep_j$ and $block_j$
4 $X_j \leftarrow$ obtain new data
5 $X_j \leftarrow$ validate the size of X_j by comparing with δ
6 **for** *(t=1 to T)* **do**
7   **if** *(t mod Δ)=0* **then**
8   t is assigned to $timestep_j$
9         $W_j \leftarrow$ extract local weights of α from M_j
10   deliver W_j, $block_j$, and $timestep_j$ to the server
11   **else**
12         $W_j \leftarrow$ extract local weights of $(1 - \alpha)$ from M_j
13   deliver W_j to the server
14   **end**
15     $M_j \leftarrow$ obtain updated model from the server
16     $block_j \leftarrow$ store global model state and data access information of time t
17 **end**
18 delete X_j from storage

V. Performance Evaluation

In this section, we assess the performance of our proposed algorithm based on extensive experimental analysis. Firstly, we discuss the dataset utilized in the experiments, and then we report the performance of the method through diverse simulation setups and performance metrics.

A. Data Preparation

We evaluated the generalization potential of the adopted FL-based distributed cardiac monitoring architecture employing four publicly available arrhythmia detection datasets. We utilized the MIT-BIH Supraventricular Arrhythmia database, referred to as DS1 [13] for the learning/training purpose. Then, the adopted techniques were extensively tested using three other public datasets from the Physionet repositories, such as the MIT-BIH Arrhythmia database, INCART 12-lead Arrhythmia database, and Sudden Cardiac Death Holter database, referred to as DS2 [14], DS3 [15], and DS4 [16], respectively. The recommendation of the Association for the Advancement of Medical Instrumentation (AAMI) is utilized for the arrhythmia classification task. We have considered four classes of heartbeats, namely N, S, V, and F, in developing this multi-class classification task, which represents normal, supraventricular ectopic, ventricular ectopic, and fusion beats, respectively [17].

B. Simulation Setup

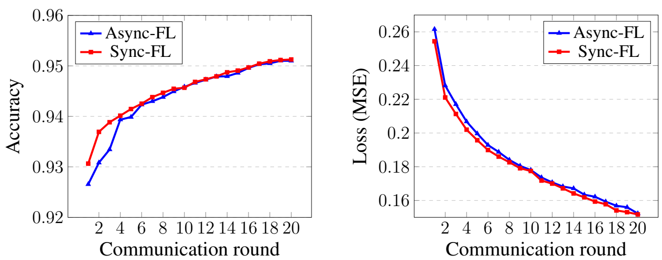
We have evaluated the performance of the FL architectures on four different classification performance metrics such as accuracy, weighted precision, weighted recall, and area under the receiver operating characteristic curve (AUC score). To identify the architectures' time and memory requirements, we have also determined the required time in seconds and memory requirement percentage for varying numbers of UENs. In the experimental analysis, the number of UENs contains different values such that $UEN \in \{2, 4, 6, 8, 10\}$.

Two variations of the proposed federated learning architecture based on a custom lightweight CNN model are simulated that are indicated as Async-FL and Sync-FL, respectively. Also, we define the number of communication rounds/iterations and time-round (Δ) to evaluate our proposal's performance. A communication round or iteration (T) refers to the number of times the entire process of the local training for all the UENs and exchanging parameters with the server takes place. In terms of the Async-FL, a time-round consists of multiple iterations/communication rounds, after which the cloud is updated with the deep model parameters. Note that the time-round concept is exclusive to the Async-FL only, treated as a hyperparameter of the system. In terms of the Sync-FL, deep model exchange occurs in every communication rounds. The number of iterations or communication round is set to 20 for both FL variations. For the Async-FL, Δ = 5. We adopted the CNN model from the proposed DL-LAC model presented in our previous work [2]. It consists of three convolution layers, followed by three fully connected layers and the output layer

This is the author's version of an article that has been published in IEEE.
The final version of record is available at https://doi.org/10.1109/MeditCom49071.2021.9647636

TABLE I: Classification performance of adopted FL architectures over varying number of UENs using three different test datasets.

Method	Dataset	Metrices	Number of Ultra-Edge nodes				
			2	4	6	8	10
Sync-FL	DS2	Accuracy	**0.88874**	**0.88979**	0.88969	0.87833	0.87717
		Precision	0.87830	0.87949	0.87440	0.86868	0.87193
		F1-score	0.88061	0.88151	0.88122	0.87185	0.87198
	DS3	Accuracy	**0.89293**	**0.88947**	0.88612	0.89517	0.86304
		Precision	0.88463	0.86772	0.85985	0.88153	0.83830
		F1-score	0.86715	0.86181	0.86750	0.86573	0.84906
	DS4	Accuracy	**0.76714**	**0.77717**	**0.75305**	0.73222	0.74874
		Precision	0.94192	0.95222	0.94839	0.95070	0.95052
		F1-score	0.61370	0.78300	0.83498	0.82060	0.79105
Async-FL	DS2	Accuracy	0.86853	0.87911	**0.89147**	**0.87903**	**0.87837**
		Precision	0.87277	0.87818	0.88315	0.87529	0.87749
		F1-score	0.86915	0.87554	0.88723	0.87520	0.87505
	DS3	Accuracy	0.87094	0.88612	**0.89478**	**0.89524**	**0.89892**
		Precision	0.84107	0.85985	0.88966	0.88134	0.89130
		F1-score	0.85204	0.86750	0.86772	0.86862	0.87387
	DS4	Accuracy	0.74033	0.73773	0.73240	**0.76935**	**0.76287**
		Precision	0.95062	0.94869	0.94417	0.95335	0.94796
		F1-score	0.81196	0.82400	0.73588	0.84454	0.69541



(a) Learning accuracy.

(b) Value of loss function

Fig. 2: The performance comparison of two federated learning architectures during the learning/training phase over varying communication rounds (employing DS1).

provides us the final heartbeat label.

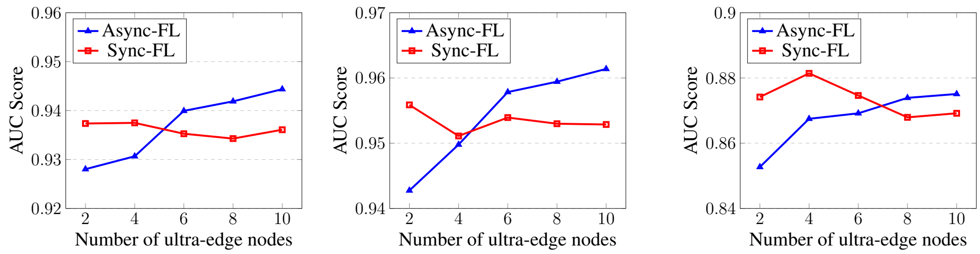
C. Results and Discussion

Firstly, employing the DS1, we train the synchronous and asynchronous federated learning architectures, referred to as Sync-FL and Async-FL. Fig. 2 compares the learning accuracy and loss of the Sync-FL and Async-FL architectures over a varying number of iterations/communication rounds. In Fig. 2 we have demonstrated the mean accuracy and loss curve for varying number of UENs. Note that the UENs start the first communication rounds with a centrally trained AI model, hence even in the first iteration/round, it obtains more than 90% accuracy. Notice from Fig. 2a that both models' learning accuracy reaches almost 95% over time. Both the variations of our proposal gain higher accuracies with the increasing number of communication rounds. During the initial rounds, the Sync-FL showed better performance than the Async-FL; however, both architectures obtain almost identical learning performance as the communication rounds progress. A similar trend is also illustrated in the learning loss, manifested in terms of the mean squared error (MSE), in Fig. 2b.

After the learning phase, we investigate the FL architectures generalization ability in the inference phase on three datasets (e.g., DS2, DS3, and DS4). Table I demonstrates the classification performance of adopted FL architectures over a varying number of UENs using the different test datasets. The results indicate that both the adopted FL architectures achieved encouraging classification accuracy in the test datasets, especially in DS2 and DS3. The Sync-FL shows superior performance than Async-FL when the number of UEN is relatively low. However, as the number of UEN increased, the classification performance showed better results in the case of the Async-FL for each of the adopted test datasets. A similar trend is also observed in the other two performance matrices (i.e., precision and f1-score).

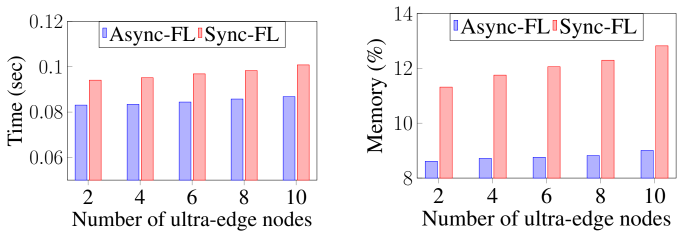
Next, in Fig. 3 we explore the AUC score to identify the area under the ROC curve by using DS2, DS3, and DS4 as the test dataset in the inference phase. Both FL architectures obtained robust AUC scores for all three adopted test datasets. A similar trend to the accuracy values demonstrated in Table I is also observed in the AUC score values as the Sync-FL obtains better scores when the number of UEN is low. Whereas, for the increasing number of UENs, the Async-FL's AUC score values are superior to the Sync-FL. The results prove that even though the Async-FL architecture transmits learned deep model parameters after time-round (Δ), as the number of users (i.e., UEN) grows higher, the classification performance becomes more robust. The encouraging arrhythmia detection performance ensures the proposed Async-FL's viability for deploying in an online collaborative learning paradigm.

In the next simulation setup in Fig. 4, we compare the required execution time and memory consumption for the inference phase for the proposed Sync-FL and Async-FL architectures. For a varying number of UENs, the average execution time (in seconds) needed in the inference phase by both FL paradigms is shown in Fig. 4a, and the memory consumption (in percentage) is displayed in Fig. 4b. Although both systems can generate results with a low time and memory requirements, due to regulating the deep model exchange ratio in the Async-FL technique, the execution time and memory consumption are lower than the counter-part Sync-FL approach. The low time and memory demand of the Async-FL architecture ensure



(a) AUC Score (DS2). (b) AUC Score (DS3) (c) AUC Score (DS4).

Fig. 3: AUC score values acquired in the inference phase of the Sync-FL and Async-FL methods using three test datasets.



(a) Time required (in seconds). (b) Memory consumption (%).

Fig. 4: Required execution time and memory consumption for varying number of UENs (inference phase).

its suitability to get used in a collaborative online learning approach with the resource-constrained ultra-edge IoT nodes.

VI. Conclusion

In this paper, we proposed an asynchronously updating FL architecture (Async-FL) for mobile and deployable UENs to build decentralized and collaborative arrhythmia detection without the need for direct ECG data exchange with the cloud. We employed raw single-lead ECG data to design the distributed FL architectures, preserve patient data privacy, and mitigate the network overhead. Extensive experimental results demonstrated the viability of the proposed Async-FL in terms of lightweight operation (e.g., low execution time and memory consumption) while attaining a considerable arrhythmia detection accuracy. Our proposal also leads to lower network overhead (i.e., bandwidth consumption, time, and memory requirements) for an increasing number of UENs. As the demand for remote patient monitoring is escalating with the emergence of pandemics such as the novel coronavirus, this particular ECG monitoring use-case using the Async-FL paradigm can lead the way to implement the envisioned future generation smart and remote health monitoring system at a mass scale.

Acknowledgment

This work was made possible by the NPRP award [NPRP13S-0205-200270] from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors. Also, the work was supported by Mitacs.

References

[1] R. K. Pathinarupothi, P. Durga, and E. S. Rangan, "IoT-Based Smart Edge for Global Health: Remote Monitoring With Severity Detection and Alerts Transmission," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2449–2462, 2019.

[2] S. Sakib, M. M. Fouda, Z. M. Fadlullah, N. Nasser, and W. Alasmary, "A Proof-of-Concept of Ultra-Edge Smart IoT Sensor: A Continuous and Lightweight Arrhythmia Monitoring Approach," *IEEE Access*, vol. 9, pp. 26 093–26 106, 2021.

[3] S. Berrouiguet, M. L. Barrigón, J. L. Castroman, P. Courtet, A. Artés-Rodríguez, and E. Baca-García, "Combining mobile-health (mHealth) and artificial intelligence (AI) methods to avoid suicide attempts: the Smartcrises study protocol," *BMC psychiatry*, vol. 19, no. 1, 2019.

[4] K. Balaskas and K. Siozios, "ECG Analysis and Heartbeat Classification Based on Shallow Neural Networks," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, May 2019.

[5] A. Hannun, P. Rajpurkar, M. Haghpanahi, T. Geoffrey, C. Bourn, M. Turakhia, and A. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Medicine*, vol. 25, Jan. 2019.

[6] A. Daamouche, L. Hamami, N. Alajlan, and F. Melgani, "A wavelet optimization approach for ECG signal classification," *Biomedical Signal Processing and Control*, vol. 7, no. 4, pp. 342–349, Jul. 2012.

[7] C. Lainscsek, P. Rowat, L. Schettino, D. Lee, D. Song, C. Letellier, and H. Poizner, "Finger tapping movements of Parkinson's disease patients automatically rated using nonlinear delay differential equations," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 1, Mar. 2012, Art. no. 013119.

[8] R. Sandeep and K. C. Ray, "Sparse representation of ECG signals for automated recognition of cardiac arrhythmias," *Expert Systems with Applications*, vol. 105, pp. 49–64, Sep. 2018.

[9] A. K. Dohare, V. Kumar, and R. Kumar, "Detection of myocardial infarction in 12 lead ECG using support vector machine," *Applied Soft Computing*, vol. 64, pp. 138–147, Mar. 2018.

[10] D. Azariadi, V. Tsoutsouras, S. Xydis, and D. Soudris, "ECG signal analysis and arrhythmia detection on IoT wearable medical devices," in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, May 2016.

[11] 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)*, Chengdu, China, Dec. 2020.

[12] A. N. Kochi, A. P. Tagliari, G. B. Forleo, G. M. Fassini, and C. Tondo, "Cardiac and arrhythmic complications in patients with COVID-19," *Journal of Cardiovascular Electrophysiology*, vol. 31, no. 5, pp. 1003–1008, 2020.

[13] S. D. Greenwald, R. S. Patil, and R. G. Mark, "Improved detection and classification of arrhythmias in noise-corrupted electrocardiograms using contextual information," in *[1990] Proceedings Computers in Cardiology*, Chicago, IL, USA, Sep. 1990.

[14] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, May 2001.

[15] A. L. Goldberger and et al., "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–220, Jun. 2000.

[16] S. D. Greenwald, "The development and analysis of a ventricular fibrillation detector," Ph.D. dissertation, Massachusetts Institute of Technology, 1986.

[17] S. Sahoo, M. Dash, S. Behera, and S. Sabut, "Machine Learning Approach to Detect Cardiac Arrhythmias in ECG Signals: A Survey," *IRBM*, vol. 41, no. 4, Aug. 2020.