Contrastive learning for atrial fibrillation detection in challenging scenarios

Karen Fonseca* , Sergio Osorio *, Jeyson Castillo and Carlos Fajardo †
Department of Electrical Engineering
Universidad Industrial de Santander
Bucaramanga, Santander, Colombia
†cafajar@uis.edu.co

***Abstract*—In healthcare applications, retraining models for new users often require collecting many labeled data, which is challenging and expensive in these types of applications, such as atrial fibrillation detection. Unsupervised and self-supervised techniques have emerged as promising methods to deal with the scarcity of labeled data. Contrastive learning is a recent technique that aims to improve model accuracy by a pre-trained process with unlabelled data. In this work, we propose the implementation of contrastive learning to improve the performance of a CNN that classifies atrial fibrillation in scenarios with few labeled data, small models, and noisy data. The strategy was evaluated in the most extensive public ECG dataset. We present results regarding the F1-score for a different amount of unlabeled-labeled data and different model sizes. The results suggest that our strategy outperforms the baseline strategy up to 30% of the 10-fold mean F1-score compared to an improvement of 5.8% AUC in the state of the art.**

***Index Terms*—Atrial Fibrillation, Contrastive Learning, Convolutional Neural Networks, Data Augmentation, ECG Signals, Self-Supervised Learning.**

I. Introduction

A cardiac arrhythmia is a disorder that arises when the heart's electrical impulses do not work correctly, which in the worst cases can lead to a stroke, heart failure, or sudden death [1]. Atrial fibrillation (AFib) is the most common sustained arrhythmia in clinical practice and is responsible for high mortality, morbidity, and increased healthcare costs. This disease is characterized by irregular and disordered atrial beats, resulting in a quick and irregular heart rhythm [2].

Heartbeats can be detected by ECG signals, which can be processed to detect and classify many heart diseases [3]. A case in point is proposed to diagnose COVID-19 using ECG data with deep learning. Due to COVID-19, many cardiovascular changes caused by this disease can be classified as cardiac arrhythmias. It is relevant to consider that COVID-19 cannot be considered the complete cause of these cardiovascular complications but may reveal underlying conditions or worsen them [4].

Deep learning techniques applied to the ECG signals have allowed the development of models capable of detecting and classifying cardiac arrhythmias [5]. However, these techniques require a considerable amount of labeled data. In state of the art, ECG datasets such as Chapman [6], Cardiology [7], Physionet 2017 [8], and Physionet 2020 [9] have been employed for cardiac arrhythmia detection. Recently, Icentia11k [10] shows up as the largest public Afib database, which has a substantial amount of noisy one-lead ECG signals.

On the other hand, self-supervised learning has proven to be a learning technique that involves finding good representations from unlabeled data, taking advantage of easy access to this type of data. A case in point is contrastive learning [11], in which the learning process consists of the similarity between two transformed inputs (A and B) from the same instance by the data augmentation process. The model has to predict if A and B come from the same data (positive pairs) or not (negative pairs), thus requiring only the data on its own without any label. The authors of contrastive learning remark on the importance of multiple data augmentation to obtain an efficient representation. Recently, CLOCS (Contrastive Learning of Cardiac Signals Across Space, Time, and Patients) [12] proposes the implementation of a family of contrastive learning methods applied to ECG signals that work across space, time, and patients making the representation similar to each other. Their approach learns representation without the use of a projection head, as seen in [11].

In addition, Patient Contrastive Learning [13] proposes another type of contrastive learning applied to ECG signals. They present a pre-training approach known as Patient Contrastive Learning of Representations (PCLR). Positive pairs are defined from samples belonging to the same patient and collected at different time-lapses. In this work, we deal with the implementation of contrastive learning in challenging situations, which involve few labeled data, small models, and noisy signals. Facing this situation provides insight into finding feasible models to implement in wearable devices. Due to the explosive growth of interest, the number of IoT devices has increased dramatically in recent years [14]. For this reason, having small models with good performance is essential.

The rest of this work is ordered as follows: Section 2 describes the employed dataset. Section 3 explains the contrastive learning method and details the network structure to implement the method. Subsequently, section 4 describes the data augmentation applied to ECG signals and shows the procedure of the contrastive learning pre-training method. Finally, in sections 6 and 7, we conclude the results.

*These authors contributed equally to this work.

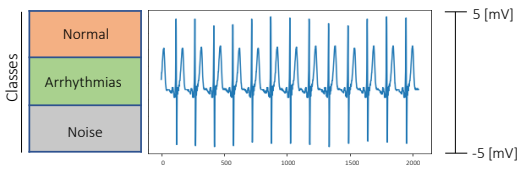


Fig. 1. Data processing. The classes we use for classification are normal, arrhythmia and noise. Also, we limit the range of the signal.

II. Dataset

We use the largest public ECG dataset of continuous raw signals for representation learning, the Icentia11k [10]. This dataset contains one-lead ECG signals recorded from 11 thousand patients using the CardioSTAT^{TM} [15] device. For each patient, 3 to 14 days are recorded at frame-level (approximately 8 seconds) with a sample rate of 250Hz and a size of 2049 samples. This raw data can capture features of the beat and the rhythm. The dataset is classified according to beat type into Normal, Premature Atrial Contractions, and Premature Ventricular Contractions. Also, the dataset is classified by rhythm type in: NSR (normal sinus rhythm), AFib (atrial fibrillation), Aflutter (atrial flutter), and Noise.

In this work, each labeled data was tagged according to [16] as NSR (Normal Sinus Rhythm) (1), Arrhythmia Afib and Aflutter (2), or Noise (3). It contains 57%, 4%, and 39%, respectively. Also, we filtered the raw data (Fig. 1) limiting the signals between 5 [mV] and -5 [mV] (range of cardiac signals) [17].

We use two different signal lengths depending on the process. For the training process, we employ K-signals of 2048 samples called single labeled data. On the other hand, for the pre-training process, we use M-double unlabeled data, referring to signals of 4096 samples. This length difference allows us to apply time transformations. As shown in Fig. 2, we ensure that both single and double data come from separate sets of the training dataset to avoid using the same patients on each stage.

III. Methods

A. Constrastive Learning

Constrative learning (CL) is a self-supervised method that aims to learn useful representations of instances that share some context. This method is based on three main aspects: a learning network structure, a set of transformations according to the type of data, and a contrastive loss function.

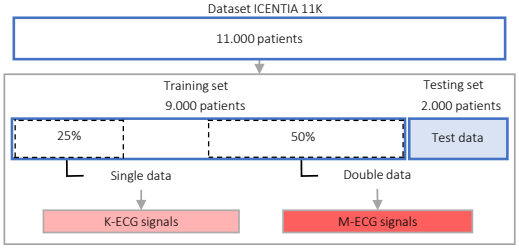


Fig. 2. Dataset distribution. After raw data processing, we divide the dataset into training and testing. From the training set, we extract the single and double data.

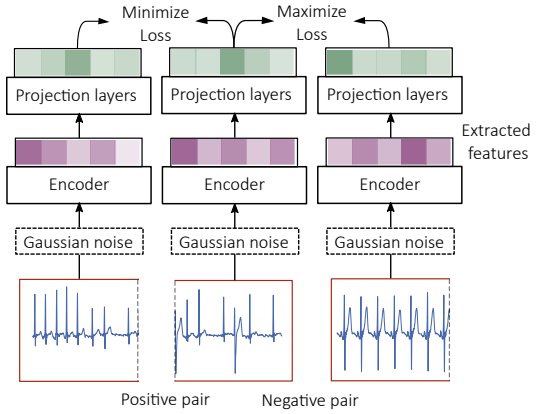


Fig. 3. Contrastive learning applied to ECG signals. Representations are obtained by passing the single ECG signal through the encoder and the projection layers. Finally, representations for positive pairs are encouraged to be similar to one another and dissimilar to representations of all other signals.

CL model comprises two parts: an encoder and projection layers. In the pre-training stage, the encoder is the convolutional network that learns to extract representations from the input data. Then, projection layers process these representations, and the contrastive loss in 1 is applied as shown in Fig. 3. An important fact about using non-linear projections layers is explored in SimCLR [11]. They conjecture that adding these layers helps the representations learn valuable information about the data.

The set of transformations generates multiple augmented views of the same instance, called positive pairs. On the other hand, views of different instances are defined as negative pairs. Finally, the contrastive loss function in equation 1 learns the outputs of the projection layers to be similar for positive pairs and dissimilar for negative pairs.

Fig. 3 illustrates a specific case in which the data come from ECG recordings. In the medical field, physiological recordings from the same patient collected on small time scales and similar scenarios are likely to share context. Therefore, positive pairs for ECG recordings could be obtained by applying time division and adding noise to the same signal.

B. Network architecture

We use as encoder the proposed model in [18] that is used to detect AFib (Fig. 4). This model allows us to change the model size by modifying the size of the initial convolutional filter. The encoder does not include the classification layer.

In addition, we build a projection layer [11] that has an input size given by the length of the encoder's output. This tiny neural network consists of 2 dense layers with a corresponding number of units defined as width. On the other hand, the linear probe layer is created to perform the classification task. These final layers are composed of a dense layer with 128

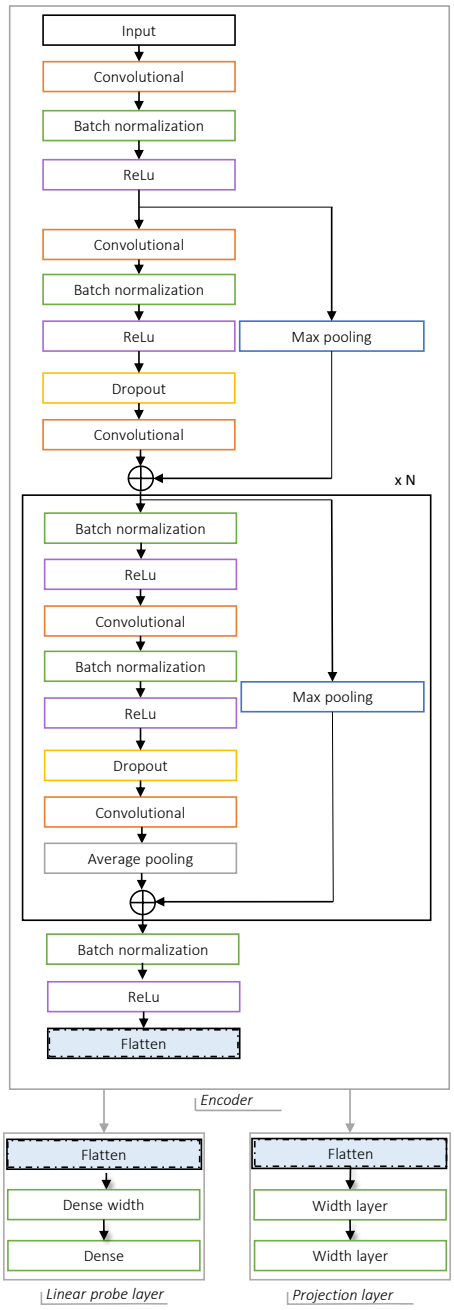


Fig. 4. Model architecture. From [18], we use an N of 13 and several values for the initial filter size to change the model parameters.

neurons (dense width) and a dense layer with three neurons (classification layer) as shown in Fig. 4

C. Data augmentation.

Data Augmentation (DA) is one approach for limited data in deep learning applications. This method adequately implemented is a critical factor that enables us to get good representations. Overall, CL has been developed in the imaging domain. Since progressively, more results have been obtained in the DA characteristics of the images. The results are intended to be repeated on data of different dimensionality. State of the art has explored ways to create similar performance by applying DA on temporal signals like ECGs.

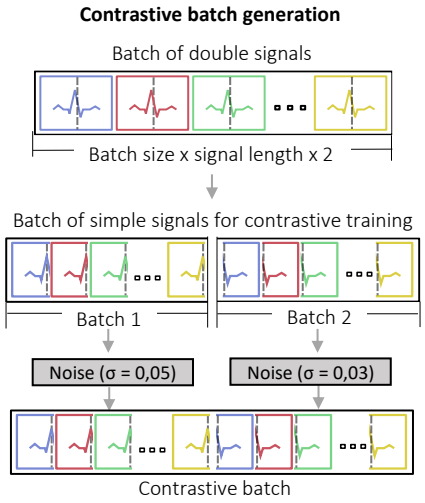


Fig. 5. Batch generation for contrastive. Two batches are created from each double data batch by dividing each signal into two equal segments. The first batch contains the first half samples, and the other has the remaining of each signal. The two mini-batches have added different Gaussian noises. Then, the batches are then concatenated.

In this work, based on [12], [13] and [19], the DA is performed applying time-division (split in two) and Gaussian's noise (different standard deviations for each transform) as shown in Fig. 5. Considering the dimensionality (one-lead) of the icentia11k signals, and therefore cannot generate DA in multiple leads. For that reason, we need to use data with double training data for pre-training to allow time transformations. We create the contrastive learning batch as illustrated in Fig. 6.

D. Pre-training stage.

We conduct our experiments in two main stages illustrated in Fig. 6. For the pre-training process [1], we include two dense layers of 128 neurons. These layers are used for the projection. We use a scheduled learning rate starting at 0.01 and early stopping callbacks for the training. First, we perform the encoder pre-training employing contrastive learning. Then, we apply a classification task to evaluate the initialization introduced by the pre-training. We conduct this procedure as follows:

- We take M-signals from the unlabeled double data (Fig. 1) to build the contrastive batches of a selected batch size as shown in Fig. 5.
- We take the contrastive batches to pre-train the encoder with the projection layers. From [11], we use the normalized temperature-scaled cross-entropy loss with $\tau = 0.1$ as the temperature parameter. The loss function works by maximizing the similarity between the positive pair's projections. In 1, the loss function of a i, j positive pair

[1]Codes available in *https://github.com/karena2/CL-for-1-D-ECG.git*

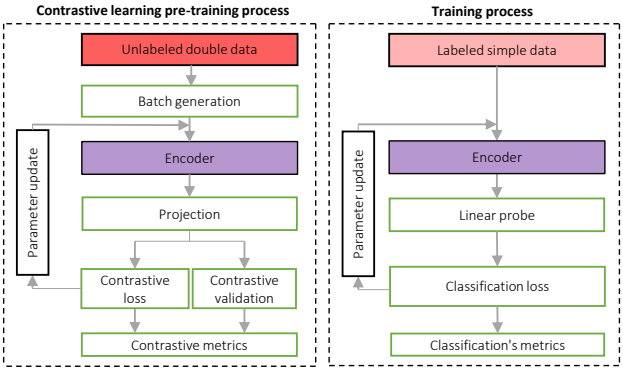


Fig. 6. Procedure for contrastive learning implementation. M-signals are drawn from the double dataset to generate the contrastive batches used to pre-train the encoder. Then, taking K-signals from the single dataset, the encoder and the linear probe layer are trained to perform the classification task.

of signals is shown, where z indicates the outputs of the projection layers, and N refers to the contrastive batch size [13]. The final loss is calculated for the N-positive pairs for each contrastive batch.

$$\ell_{i,j} = -\log \frac{\exp\left[sim\left(z_i, z_j\right)/\tau\right]}{\sum_{s=1}^{2N, s\neq i} \exp\left[sim\left(z_i, z_s\right)/\tau\right]} \quad (1)$$

Where sim corresponds to the cosine similarity.

- The contrastive metrics are the contrastive loss and accuracy.

E. Training stage.

The training process uses the Sparse Categorical Cross-entropy loss function and the Adam optimizer with default parameters. We use a scheduled learning rate starting at 0.001 and early stopping callbacks for the training.

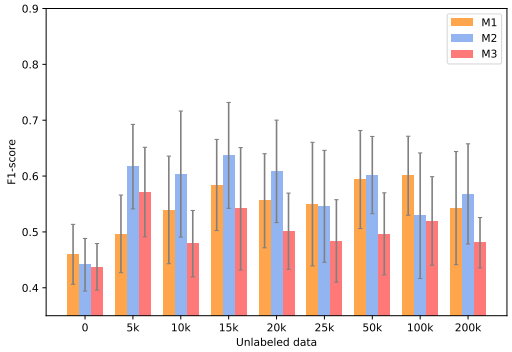
We conduct this procedure as follows:

- Once the pre-training stage is completed, we take the pre-trained encoder and append the linear probe layer to it. Finally, we train the encoder using K-signals from the single data, as shown in Fig. 6.
- The classification metrics are the loss and F1-score.

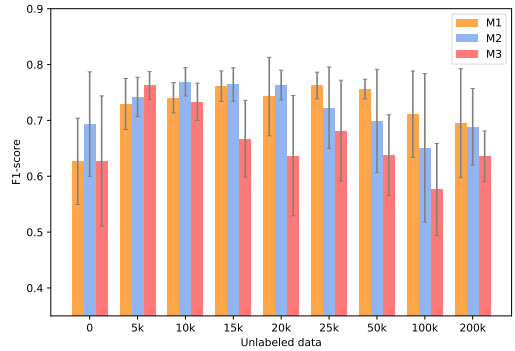
IV. Results

We train several models and calculate the F1-score, which is the harmonic mean of the precision and the recall. The F1-score is ideal for unbalanced datasets. Also, we use Stratified data 10-Folds to calculate the mean and the standard deviation of the results. We run the simulations on Tesla V100 SXM2 GPUs. For both processes, pre-training and training, we set a batch size of 256 ECG signals.

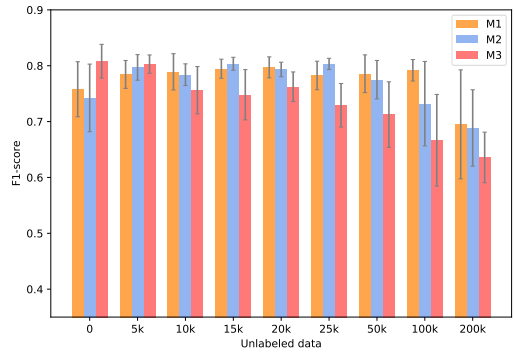
Fig. 7 shows results regarding the amount of labeled data, unlabeled data and model size. Labeled data ranges between 5k (Fig. 7(a)), 10k (Fig. 7(b)), and 15k (Fig. 7(c)) ECG signals. We also use three models with different amount of parameters: M1 (orange), M2 (blue), and M3 (red). These values correspond to 66k, 250k, and 1M of parameters. The unlabeled data ranges between 0, 20k, 25k, 50k, 100k, and 200k. The zero value indicates that the model has random initialization, i.e., there is not pre-training process.



(a) 5k labeled data.



(b) 10k labeled data.



(c) 15k labeled data.

Fig. 7. Graphs of F1-score results by changing unlabeled data.

In Fig. 7(a), note a significant improvement in the mean F1 score when using CL compared to random initialization with few labeled data. Furthermore, our results suggest that 15k of unlabeled ECG signals can achieve better results.

Meanwhile, Fig. 7(b) shows similar behavior The results obtained by applying CL decrease the standard deviation in big models with more labeled data. However, Fig. 7(c) shows that the use of a large amount of unlabeled data does not show any improvement for large models.

On the other hand, tests varying the depth of the projection layer in the pre-training stage did not show enhanced results,

TABLE I
COMPARISON TABLES ABOUT IMPROVEMENT WITH DIFFERENT AMOUNT OF LABELED DATA AND DATASETS

[a]

Method	Training dataset	Labeled data	Test data	AUC-Random Initialization	AUC-Contrastive initialization	AUC-mean improvement
CLOCS	Cardiology [7]	∼3k	∼1k	0.669 ± 0.007	0.708 ± 0.017	5.8%
	Physionet 2017 [8]	∼5k	∼6k	0.738 ± 0.014	0.770 ± 0.012	4.6%
	Physionet 2020 [9]	∼16k	∼16k	0.766 ± 0.005	0.801 ± 0.013	4.3%

[b]

Method	Training dataset	Labeled data	Test data	F1-Random Initialization	F1-Contrastive initialization	F1-mean improvement
Ours	Icentia 11k [10]	∼5k	∼465k	0.459 ± 0.053	0.601 ± 0.071	30.5%
		∼10k		0.627 ± 0.077	0.763 ± 0.024	21.6%
		∼15k		0.742 ± 0.060	0.803 ± 0.010	8.1%

giving similar values despite the changes. Likewise, freezing the encoder when performing the classification task showed unfavorable results; since the models tends to stabilize at a value of F1-score significantly lower than the average obtained by random initialization.

V. DISCUSSION

Deep learning applied to medical ECG datasets with few labeled training signals presents challenges. CL can use unlabeled data, which are easier to obtain, to compensate for the scarcity of labeled data. Our research focuses on the CL implementation at scenarios to use in portable devices with few labeled data, noisy data, and small models.

Table I shows a comparison between our method and CLOCS [12] methods. We use a percentage of improvement in the F1 score referenced on the random initialization. Note that our work presents an improvement compared to CLOCS in nearby conditions. However, we know that the standard deviations are high in most of our results. This high deviation could be that we use the larger test set with a significant amount of noisy signals.

The 1-D data has proved to be a challenge in the context of CL. This type of data reduces the number of possibilities in DA, which is important to improve the learning in the pre-training stage.

Our results suggest that increasing the amount of labeled data or the model size reduces the impact of our method. On the other hand, our results also suggest that the amount of unlabeled data used in pre-training determines the success of the implementation technique. To find the optimal amount of unlabeled data is necessary to get the best performance of contrastive learning.

VI. CONCLUSION

This work covers different scenarios to perform deep neural network training for cardiac arrhythmia detection using the CL method. We found that CL improves the F1-mean in small models with few labeled data, which are better for deploying portable devices. However, bigger models and many labeled data do not show the best conditions for our method.

REFERENCES

[1] ClevelandClinic, "Arrhythmia: Types, symptoms, causes, diagnosis & treatments," Cleveland Clinic https://my.clevelandclinic.org/health/diseases/16749-arrhythmia, Dec 2018.

[2] N. C. G. I. García-bolao, "Fibrilación auricular y flutter auricular Keywords :," *Medicine*, vol. 11, no. 39, pp. 2338–2345, 2013. [Online]. Available: http://dx.doi.org/10.1016/S0304-5412(13)70627-9

[3] P. Shahbeigi-Roodposhti and S. Shahbazmohamadi, "Acquisition and Analysis of an ECG (electrocardiography) Signal," Biomedical Engineering Department, University of Connecticut, Storrs, Connecticut. JoVE, Cambridge, MA, 2021.

[4] M. A. Ozdemir, G. D. Ozdemir, and O. Guren, "Classification of covid-19 electrocardiograms by using hexaxial feature mapping and deep learning," *BMC medical informatics and decision making*, vol. 21, no. 1, pp. 1–20, 2021.

[5] S. Parvaneh, J. Rubin, S. Babaeizadeh, and M. Xu-Wilson, "Cardiac arrhythmia detection using deep learning: A review," *Journal of electrocardiology*, vol. 57, pp. S70–S74, 2019.

[6] J. Zheng, J. Zhang, S. Danioko, H. Yao, H. Guo, and C. Rakovski, "A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients," *Scientific data*, vol. 7, no. 1, pp. 1–8, 2020.

[7] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," Nature Medicine, Vol 25, 65-69, pp. 65–69, 2019.

[8] G. D. Clifford, C. Liu, B. Moody, L. H. Lehman, I. Silva, Q. Li, A. E. Johnson, and R. G. Mark, "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017," *Computing in Cardiology*, vol. 44, pp. 1–4, 2017.

[9] E. A. P. Alday, A. Gu, A. J. Shah, C. Robichaux, A.-K. I. Wong, C. Liu, F. Liu, A. B. Rad, A. Elola, S. Seyedi *et al.*, "Classification of 12-lead ecgs: the physionet/computing in cardiology challenge 2020," *Physiological measurement*, vol. 41, no. 12, p. 124003, 2020.

[10] S. Tan, G. Androz, A. Chamseddine, P. Fecteau, A. Courville, Y. Bengio, and J. P. Cohen, "Icentia11k: An unsupervised representation learning dataset for arrhythmia subtype discovery," *arXiv preprint arXiv:1910.09570*, 2019.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *37th International Conference on Machine Learning, ICML 2020*, vol. PartF168147-3, pp. 1575–1585, 2020.

[12] D. Kiyasseh, T. Zhu, and D. A. Clifton, "Clocs: Contrastive learning of cardiac signals across space, time, and patients," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5606–5615.

[13] N. Diamant, E. Reinertsen, S. Song, A. Aguirre, C. Stultz, and P. Batra, "Patient Contrastive Learning: a Performant, Expressive, and Practical Approach to ECG Modeling," *arXiv preprint arXiv:2104.04569*, pp. 1–17, 2021. [Online]. Available: http://arxiv.org/abs/2104.04569

[14] M. Merenda, C. Porcaro, and D. Iero, "Edge machine learning for ai-enabled iot devices: A review," *Sensors*, vol. 20, no. 9, p. 2533, 2020.

[15] P. Paquet, D. Levesque, and P. Fecteau, "Adhesive extender for medical electrode anduse thereof with wearable monitor," Jun. 20 2019, uS Patent App. 16/093,151.

[16] C. Fajardo, A. S. Parra, T. Castellanos, and K. Roy, "Towards atrial fibrillation detection in wearable devices using deep learning," *Available at SSRN: https://ssrn.com/abstract=3995164*, 2022.

[17] P. Kumar and V. K. Sharma, "Cardiac signals based methods for recognizing heart disease: A review," in *(ICICV)*, 2021, pp. 1375–1377.

[18] C. Fajardo and K. Roy, "A compressed deep learning model to detect cardiac arrhythmias," 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Submitted), 2021.

[19] C. Liu and J. Li, *Feature engineering and computational intelligence in ECG monitoring*. Springer, 2020.