International Information and
Engineering Technology Association
Advancing the World of Information and Engineering

Processing Biomedical Signals by Neural Networks Using Hardware-Constrained System

Maytham N. Meqdad*[ID], Reyam Thair Ahmed[ID], Mustafa AL-Handhal[ID]

Intelligent Medical Systems Department, College of Sciences, Al-Mustaqbal University, Babylon 51001, Iraq

Corresponding Author Email: maytham.meqdad@uomus.edu.iq

ABSTRACT

On-line detection of arrhythmia in 12-lead electrocardiogram signals by deep learning models is essential for clinical care. If an 8-byte floating point data type is used to define each sample in a 12-lead ECG signal, the volume of a Rosent-18 class is 800.4 MB (100.06 M * 8 B). This model is challenging to apply to devices with minimal hardware. Consequently, these models are inadequate for practical purposes, and their utilization is restricted when it comes to low-capacity devices within emerging fields like the Internet of Medical Things. This article introduces a technique that aims to categorize irregularities in 12-lead electrocardiogram signals on edge devices. The method utilizes a lightweight learning approach for the classification of arrhythmias. The evident originality of this work is the use of different evaluations to deploy the suggested model on a device with hardware limitations. After employing the Tensor Flow Lite platform, a compact model has been derived from it. This model has been deployed on an Android device as an edge device, carrying forward from the previous context. According to the assessment, the suggested classification model, designed to categorize 11 different irregularities within the electrocardiogram (ECG) dataset comprising 10,646 patients, achieves an accuracy level comparable to 83.45%. Ultimately, the performed comparisons reveal that the proposed model exhibits competitive performance when compared to alternative approaches that rely on standard deep learning models.

1. INTRODUCTION

An electrode is applied to the skin during an electrocardiogram (ECG) to assess the electrical activity produced by the heart over time. Each person's ECG signal is unique, hence there are various uses for this signal. Among these uses are barometer detection, arrhythmia detection systems, and monitoring an individual's vital signs. Deep learning has emerged as a top method for classifying irregularities in ECG signals in recent years [1, 2]. Deep learning techniques may automatically produce high-level features that are similar to human knowledge by adding additional data and layers. This eliminates the requirement for expert knowledge or manual feature engineering. The quantity and diversity of input leads directly affect the deep learning classifier's ability to identify abnormalities in the ECG data [3, 4]. Reducing mortality and expanding treatment options are two benefits of quick and accurate identification of many kinds of illnesses. Deep learning classifiers have been used in an effort to develop practical solutions for the early identification of heart failure from the ECG signal since the advent of computer-aided diagnosis tools [5, 6]. The issue of manually extracting features has been resolved by deep learning classifiers, which carry out this task automatically in deep layers [7].

Deep learning classifiers, particularly Convolutional neural network (CNN), have demonstrated significant potential in the years to come in resolving computer vision issues [8-10].

By storing these features in the feature map, CNN's layered architecture enables the computer to recognize characteristics in the ECG signal, such as form, color, and arrhythmia [11]. The fully linked layer receives the features used to identify arrhythmia in the ECG signal and reports the type of potential arrhythmia [12].

Most deep learning models typically possess intricate structures and a substantial number of parameters, demanding substantial computational resources within the system [13, 14]. Nonetheless, due to the restricted resources available, edge devices are incapable of offering a platform to execute these models. The article introduces a technique called lightweight learning, suggesting a method capable of automatically categorizing 11 different types of irregular heart rhythms in electrocardiogram signals [15]. This classification is achieved by utilizing edge devices. The existence of this system results in faster detection and the ability to provide early alerts when an incident occurs. The suggested approach involves creating a model utilizing one-dimensional convolutional neural networks. Subsequently, through the utilization of the TensorFlow Lite framework, the model is converted into a more lightweight version suitable for running on a smartphone. The suggested approach involves creating a model utilizing one-dimensional convolutional neural networks. Subsequently, through the utilization of the TensorFlow Lite framework, the model is converted into a more lightweight version suitable for running on a smartphone.

The second section of this article will delve into an examination of the theories and historical context surrounding the previous research efforts related to the topic. The third section delves into the approach taken to carry out the task. Initially, there is an examination of the specifics regarding the dataset utilized and the essential preprocessing procedures. Subsequently, attention is directed towards the creative process of this system and its implementation on a mobile phone. The fourth section provides a detailed account of the outcomes and assessment of this work. Finally, the fifth section of this article evaluates the possibilities and potential applications that can be derived from it in future endeavours.

2. RELATED WORK

Every article that was reviewed was utilized in the CNC Challenge starting in 2021. With the Chapman ECG dataset, this has proven to be a true problem. All 10464 patients whose data was obtained for this dataset were actual. A model incorporating 31 one-dimensional convolutional layers has been introduced by Li et al [3], utilizing deep learning techniques and built upon the Residual Neural Network (ResNet) architecture. The foundation of this model relies on the Massachusetts Institute of Technology-Beth Israel (MIT-BIH) dataset, comprising data from two leads of electrocardiogram signals. Its purpose is to identify and diagnose different forms of cardiac arrhythmias. According to the study [4], this suggested approach has achieved an average accuracy of 99.06% in a single lead and 99.38% in two leads.

Presently, the field of automated detection of cardiac arrhythmias faces a challenge due to limited available samples for specific irregularities, creating a hurdle for the systems involved. Mehari and strodthoff [5] have enhanced the efficiency of the model by employing supervised learning in the same dataset size, as opposed to traditional learning methods. Using this technique, a model has been developed that relies on a 12-lead electrocardiogram signal obtained from the ChapmanECG, Ribeiro, and CinC2020 datasets. The model employs a self-encoding method with 8 convolutional layers and achieves an accuracy of 90%.

One more challenge in diagnostic systems is how they will perform when faced with unseen data. In this context, Meqdad et al. [4] have carried out a study employing a structural algorithm that incorporates a combined loss function within convolutional neural networks. This capability allows models to work together and share their information when faced with limited data. The method of evolutionary algorithm programming is a complex technique that represents them as evolutionary trees of genetic programming algorithms, aiming to address this problem in an encoded manner. The ChapmanECG dataset was used to train this model, which can identify 7 different cardiac arrhythmias with a precision of 98%.

Due to the functional correlations among the 12-lead electrocardiogram signals, it is possible to combine the data to detect cardiac arrhythmias. Meqdad et al. [6] utilized the time-frequency transformation method in a separate study to combine functional data obtained from electrodes and extract electrocardiogram frequency data in 12 leads. In the second phase, the process of genetic programming is utilized to encode deep learning. Eventually, these two stages are merged in order to enable the identification of abnormalities in the heart. The training of this model was conducted utilizing the ChapmanECG dataset, achieving an accuracy rate of 60/97%.

3. MATERIALS AND METHODS

This section will provide a detailed explanation of how the proposed system will be implemented. Figure 1 displays the stages that have been accomplished. Initially, we will delve into the dataset utilized and the necessary preprocessing procedures that need to be addressed. This passage examines the convolution model and its characteristics, followed by enhancing the trained model to a version that is lightweight and can be executed on edge devices. Ultimately, the implementation of the model on an Android smartphone will take place [16]. It should be mentioned that weights in the deployed model are taken from the primary ECG data in order to maintain security and privacy in the suggested method. As a result, actual evaluations do not employ the main data itself. Instead, weights from their extracted features will be used to determine the deployed model's weights. Thus, privacy and security are taken into consideration here.
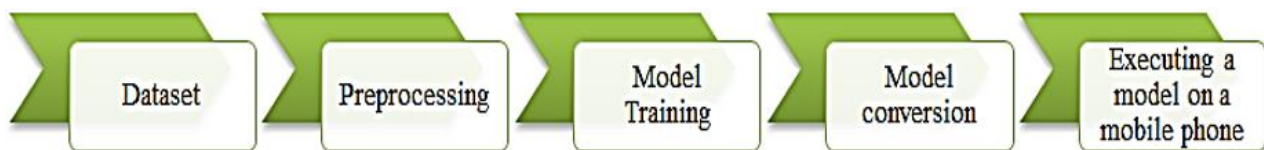

Figure 1. Flowchart of the proposed method

3.1 Dataset

The diagnostic model in this article has been trained using the ChapmanECG dataset. The dataset encompasses cardiac data obtained from patients at Shaoxing Hospital through the utilization of 12 leads in electrocardiography. Renowned experts have labeled all 11 heart rhythm types included in this dataset. This dataset contains data for a total of 10,646 individuals, comprising 5,956 samples from males and 4,690 samples from females. Out of the samples, 17% exhibit a regular sinus rhythm, while the remaining 83% have some form of cardiac irregularity. Patients within the age bracket range between 51 and 80 years old. The recording of electrocardiogram data has occurred at intervals of 10 seconds. The information is saved in CSV files, where there is a distinct identifier assigned to each patient, and the data is organized into 5000 rows and 12 columns [17].

3.2 Preprocessing

In certain cases, within the dataset, specific sample labels do not contain any stored data and are regarded as empty.

There are multiple approaches available for addressing this issue, which encompass filling the gaps with zeros, utilizing the average of the remaining data within that particular segment, or completely eliminating that specific sample. Samples that have missing values for certain leads have been eliminated in this task [18-21]. Next, the data undergoes normalization using the Min-Max normalization technique. This approach involves standardizing the data to fit within the uniform range of [0, 1].

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (1)$$

3.3 Model training

During this phase, we will train a convolutional model that is appropriate for the edge device with a reduced number of parameters when compared to alternative architectures. As mentioned earlier, convolutional models tend to have large sizes and complex structures, making them unsuitable for execution on processors of edge devices. Consequently, it becomes crucial to opt for a more compact, straightforward,

and efficient architecture with fewer parameters. This architecture should still be able to deliver fast performance despite its simplicity. The method proposed in this article makes use of the MobileNet architecture, renowned as one of the most well-known and lightweight models designed for running on-edge devices [22-24]. Many studies have been conducted recently to produce lightweight deep learning classifiers, and methods like compressed convolution filters, low-rank decomposition, pruning and sharing of parameters, and classifier acceleration have all been proposed. In addition, other classifications like have also been applied. The aforementioned techniques are effective at reducing the size of the classification; nonetheless, their main goal is to optimize the internal parameters. However, the CNN model's structure and architecture have been modified in the MobileNet model to incorporate new filters, leading to the creation of a more ideal architecture. Mobile architecture relies on incorporating 13 convolutional layers, specifically utilizing a separable depth-wise convolutional layer. This choice results in a decrease in parameter count and contributes to the lightweight nature of this architectural design.
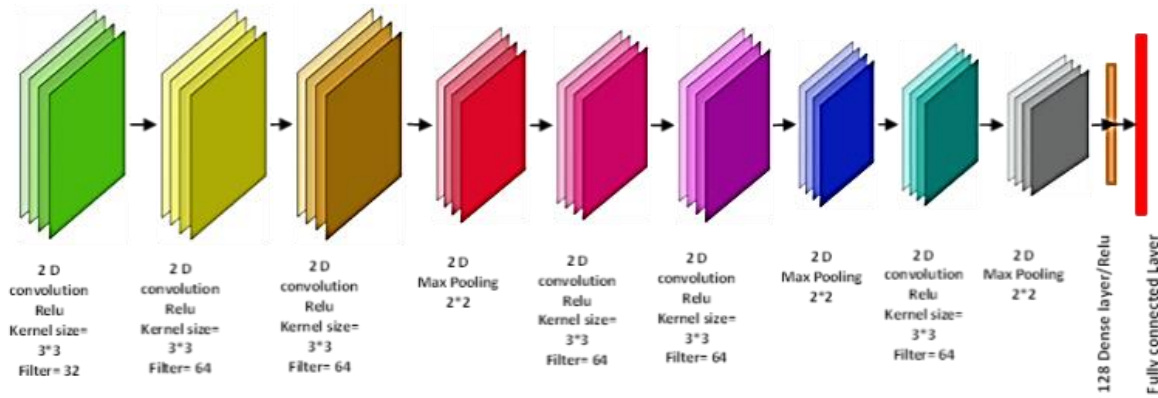

Figure 2. A representation of the layer arrangement in the architecture used in the proposed method

```
107  #%% 6.Creating LSTM model¶
108  # importing libraries
109  from keras.layers import Dense , Dropout , LSTM
110  from keras.models import Sequential
111  from sklearn.preprocessing import StandardScaler
112
113  scaler = StandardScaler()
114  model_lstm = Sequential()
115
116  model_lstm.add(LSTM(500,return_sequences=True , activation = "relu",
117                      input_shape = (X_train.shape[1], w ))) #64 lstm neuron block
118  model_lstm.add(Dropout(0.1))
119
120
121  model_lstm.add(LSTM(350, return_sequences= False))#, activation = "relu" )
122  model_lstm.add(Dropout(0.1))
123
124
125  model_lstm.add(Dense(1))
126  #
127  model_lstm.compile(loss = "mean_squared_error", optimizer = "adamax", metrics = ["accuracy"])
128  history2 = model_lstm.fit( X_train, y_train , epochs = 20 , batch_size = 200 )
129
```

Figure 3. The pseudocode for transform the trained model into a compact and runnable model

The purpose of incorporating the separable depth-wise convolution layer in convolutional models is to decrease the computational burden. This layer comprises two elements, namely depth-wise convolution and pointwise convolution, which serve as substitutes for conventional convolution. Depth-wise convolution performs a similar function to the

filtering stage in standard convolution but with a distinction in that, this layer employs only a single m×m kernel, while standard convolution allows for the utilization of multiple kernels. Convolution involving a dot can be seen as analogous to the merging phase within regular convolution. During the integration phase of regular convolution, the

layers are merged together. Conversely, in pointwise convolution, a 1×1 convolution is employed on the depth-wise outputs of the convolution, leading to the merging process. This approach results in a notable reduction in computational workload when compared to the standard mode. Overall, the architecture consists of standard convolutions, which make up 61.74% of the parameters, while 6.1% are separable depth-wise convolutions, and the remaining 33.24% are comprised of fully connected layers. Figure 2 displays the depicted arrangement of layers within the MobileNet architecture. This approach employs mutual cross-validation in combination with k-fold. This approach aims to generate a model that is both appropriate and more effective. During the ongoing model training process, a combination of 16 categories and the Adam optimizer will be employed, utilizing a five-fold approach [25].

3.4 Model conversion

Typically, running machine learning models necessitates substantial computational power and ample memory on the device. However, this specific demand presents difficulties when it comes to implementing these models on edge devices like mobile phones. To address this issue, TensorFlow has made available a tool known as TensorFlow Lite to its users. TensorFlow Lite enables the utilization of machine learning models on edge devices, such as smartphones. The primary objective of this tool is to perform model execution with minimal delay and an appropriate execution timeframe. TensorFlow Lite offers several characteristics, such as robust security, the ability to work offline, compact dimensions, adaptability, and energy efficiency. You can utilize this format in languages such as Java, Python, C++, C, and Swift [26]. To achieve this, the initial step involves providing the pre-trained model as input to the TensorFlow Lite converter. Subsequently, the model will undergo a conversion process resulting in the creation of a file with TensorFlow Lite extension. The code fragment that corresponds to Figure 3 can be seen.

3.5 Executing the model on a mobile device

The aim of this method is to implement an intelligent detection system on an Android smartphone. Android is widely recognized as a popular mobile operating system, boasting the largest user base among individuals. The objective of this article is to deploy the suggested system on Android smartphones as a device operating at the edge. The Android programming environment, known as Android Studio, has the ability to handle TensorFlow Lite files. Hence, it is essential to include the required dependencies in the Android project initially. Once the file is added to the project, executing the trained model becomes straightforward by utilizing the TensorFlow Lite interpreter in Android.

The system has the ability to receive a CSV file containing electrocardiogram data from the user. Subsequently, the file will undergo conversion to a byte buffer format and then be used as input for the TensorFlow Lite interpreter. Once the model predicts the type of cardiac arrhythmia, the system presents the diagnostic results to the user.

4. RESULTS AND DISCUSSION

This section presents the findings of evaluating the proposed method through multiple experiments. The simulation was conducted using a system equipped with an 11th Gen Intel Core i7 11800H processor, 32GB RAM, and NVIDIA GeForce RTX 3060 Laptop graphics processing unit (GPU). The article explores various architectures within convolutional neural networks, each with its unique features and efficiencies, to train the model. Table 1 displays the details of three trained models using the ChapmanECG dataset. The results obtained from these training sessions suggest that, for cardiac arrhythmia detection, the one-dimensional convolutional network outperforms the two-dimensional convolution approach. Table 2, which contrasts the number of parameters and computational complexity of the MobileNet model with other models. This table makes it evident that the use of MobileNet in this piece is justified by the model's low weight.

Table 1. Evaluation results of deep learning models for classification of samples in the ChapmanECG dataset

Model Name	Results	One-Dimensional Convolution	Two-Dimensional Convolution
Mobile Net	Accuracy	82.21%	83.45%
	Loss function value	324	51.22
	Time	93 min	14 min
ResNet	Accuracy	96.25%	92.25%
	Loss function value	12.67	18.09
	Time	123 min	24 min
LeNet	Accuracy	95.94%	91.24%
	Loss function value	12.98	14.57
	Time	3 min	4 min

Table 2. Comparing the models in terms of size and computational complexity

Models	No. of Parameters	Computation (GFLOPs)	GPU Memory
MobileNet	~64.12 M	~ 7.81	2.36 G
ResNet	~192. 84 M	~ 55.92	30.61 G
LeNet	~102. 84 M	~ 36.90	21.64 G

As previously mentioned, the number of parameters and the complexity of a convolutional neural network architecture play a crucial role in its performance on edge devices. In this study, a convolutional neural network model was trained using the ChapmanECG dataset, which contains electrocardiogram data for 10,646 individuals. To tackle the challenge of running neural network models on edge devices, it is beneficial to utilize architectures with reduced complexity and a smaller number of parameters. The model has undergone training using the k-fold cross-validation technique, with a value of 5 assigned to the parameter 'k'. The loss function graph is depicted in Figure 4, illustrating the training and testing results of the first to fifth folds.
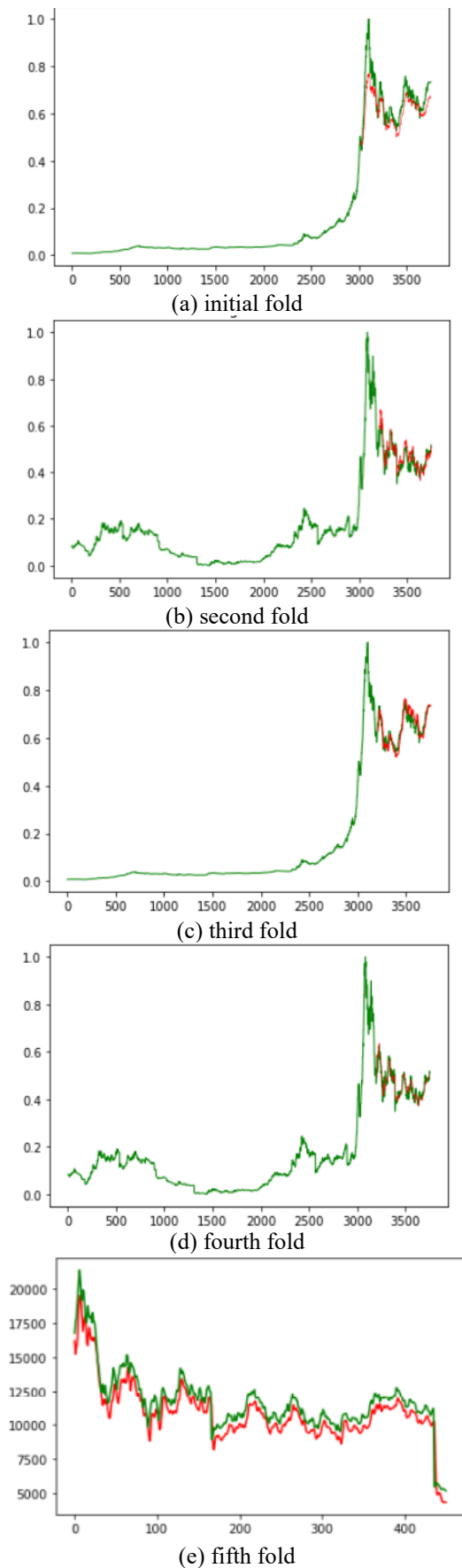
(a) initial fold

(b) second fold

(c) third fold

(d) fourth fold

(e) fifth fold

Figure 4. The loss function's values during both the training and testing phases in the different fold

The analysis of the loss function charts during both the training and testing reveals that the imbalance in sample distribution across different classes within the dataset has led

to a substantial increase in the loss value during evaluation. However, during the training phase in each of the five folds, the loss value gradually decreased at a moderate rate. Conversely, during testing, there has been a consistent upward trend in the loss value. Specifically, in the second fold, the loss function experienced a notable increase with a steep slope. This ascending pattern persisted until the fourth fold, but in the fifth fold, there was a considerable decrease in the loss value during both the training and evaluation stages. The model training utilized Keras, which is a library built on TensorFlow. When trained and tested using a GPU and the one-dimensional MobileNet architecture, it took approximately 90 minutes to complete the process. The model achieved an accuracy of 83.45%, based on 10,646 samples and 11 labels. Figure 5 provides a comprehensive overview of the suggested Android system. Figure 5 shows how to enter the ECG file and the result of detecting the type of arrhythmia. The MobileNet model in this article is assessed using the folding method, as the article's text states. As a result, the model's performance against the hidden data has not decreased. By doing this, the model's generalization will be preserved.
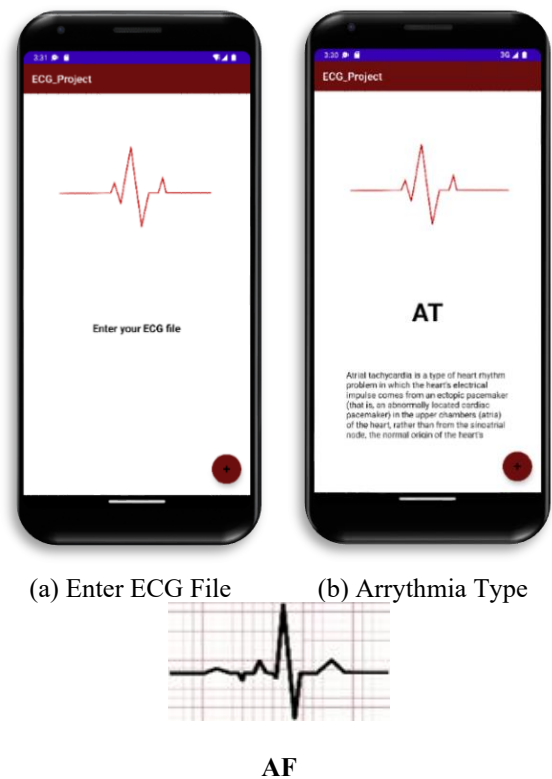

(a) Enter ECG File (b) Arrythmia Type

AF

Figure 5. An illustration of the stages of real-world evaluation of the proposed system on an Android smartphone

5. CONCLUSIONS

The proposed approach involved creating a system specifically designed to classify 11 different types of irregular heart rhythms using electrocardiogram (ECG) signals from the ChapmanECG dataset. This dataset contained a total of 10,646 samples and was tailored for use on edge devices. The system utilized one-dimensional convolution and adopted the MobileNet architecture, which was trained and tested with an accuracy level of 83.45%. The findings clearly indicated that one-dimensional convolution

outperforms two-dimensional convolution when it comes to analyzing electrocardiogram signals. Following this, a lightweight model was derived using the TensorFlow Lite framework, and ultimately, the model was implemented on an Android device. This article demonstrated how, because of the way its internal filters are designed, the MobileNet model is used to analyze one-dimensional data like ECG. Furthermore, compared to previous deep learning models, this one has been demonstrated to have fewer parameters and lower computing complexity. The topic of diversity was the one that this paper did not completely explore. Because deep or other models are probabilistic, variety is a critical concern. Different behaviors are displayed by the model due to diversity. Consequently, the writers intend to look into this matter in their upcoming work.

REFERENCES

[1] Faust, O., Acharya, U.R. (2021). Automated classification of five arrhythmias and normal sinus rhythm based on RR interval signals. Expert Systems with Applications, 181: 115031. https://doi.org/10.1016/j.eswa.2021.115031

[2] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

[3] Li, Z., Zhou, D., Wan, L., Li, J., Mou, W. (2020). Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram. Journal of Electrocardiology, 58: 105-112. https://doi.org/10.1016/j.jelectrocard.2019.11.046

[4] Meqdad, M.N., Abdali-Mohammadi, F., Kadry, S. (2022). Meta structural learning algorithm with interpretable convolutional neural networks for arrhythmia detection of multisession ECG. IEEE Access, 10: 61410-61425. https://doi.org/10.1109/ACCESS.2022.3181727

[5] Mehari, T., Strodthoff, N. (2022). Self-supervised representation learning from 12-lead ECG data. Computers in Biology and Medicine, 141: 105114. https://doi.org/10.1016/j.compbiomed.2021.105114

[6] Meqdad, M.N., Abdali-Mohammadi, F., Kadry, S. (2022). A new 12-lead ECG signals fusion method using evolutionary CNN trees for arrhythmia detection. Mathematics, 10(11): 1911. https://doi.org/10.3390/math10111911

[7] Sepahvand, M., Abdali-Mohammadi, F. (2022). A novel method for reducing arrhythmia classification from 12-lead ECG signals to single-lead ECG with minimal loss of accuracy through teacher-student knowledge distillation. Information Sciences, 593: 64-77. https://doi.org/10.1016/j.ins.2022.01.030

[8] Suh, J., Kim, J., Lee, E., Kim, J., Hwang, D., Park, J., Rhee, W. (2021). Learning ECG representations for multi-label classification of cardiac abnormalities. In 2021 Computing in Cardiology (CinC), Brno, Czech Republic, pp. 1-4. https://doi.org/10.23919/CinC53138.2021.9662753

[9] Liu, Y., Li, Q., He, R., Wang, K., Liu, J., Yuan, Y., Zhang, H. (2022). Generalizable beat-by-beat arrhythmia detection by using weakly supervised deep learning. Frontiers in Physiology, 13: 850951. https://doi.org/10.3389/fphys.2022.850951

[10] Sodmann, P.F., Vollmer, M., Kaderali, L. (2021). Segment, perceive and classify-multitask learning of the electrocardiogram in a single neural network. In 2021 Computing in Cardiology (CinC), Brno, Czech Republic, pp. 1-4. https://doi.org/10.23919/CinC53138.2021.9662830

[11] Mastoi, Q.U.A., Wah, T.Y., Mohammed, M.A., Iqbal, U., Kadry, S., Majumdar, A., Thinnukool, O. (2022). Novel DERMA fusion technique for ECG heartbeat classification. Life, 12(6): 842. https://doi.org/10.3390/life12060842

[12] Subasi, A., Ercelebi, E. (2005). Classification of EEG signals using neural network and logistic regression. Computer Methods and Programs in Biomedicine, 78(2): 87-99. https://doi.org/10.1016/j.cmpb.2004.10.009

[13] Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. Journal of Neural Engineering, 4(2): R1. https://doi.org/10.1088/1741-2560/4/2/R01

[14] Sepahvand, M., Abdali-Mohammadi, F. (2017). Evolutionary metric-learning-based recognition algorithm for online isolated Persian/Arabic Characters, reconstructed using inertial pen signals. IEEE Transactions on Cybernetics, 47(9): 2872-2884. https://10.1109/TCYB.2016.2633318

[15] Mathunjwa, B.M., Lin, Y.T., Lin, C.H., Abbod, M.F., Shieh, J.S. (2021). ECG arrhythmia classification by using a recurrence plot and convolutional neural network. Biomedical Signal Processing and Control, 64: 102262. https://doi.org/10.1016/j.bspc.2020.102262

[16] Rahul, J., Sharma, L.D. (2022). Automatic cardiac arrhythmia classification based on hybrid 1-D CNN and Bi-LSTM model. Biocybernetics and Biomedical Engineering, 42(1): 312-324. https://doi.org/10.1016/j.bbe.2022.02.006

[17] Ramkumar, M., Lakshmi, A., Rajasekaran, M.P., Manjunathan, A.J.B.S.P. (2022). Multiscale Laplacian graph kernel features combined with tree deep convolutional neural network for the detection of ECG arrhythmia. Biomedical Signal Processing and Control, 76: 103639. https://doi.org/10.1016/j.bspc.2022.103639

[18] Kobat, M.A., Karaca, O., Barua, P.D., Dogan, S. (2021). Prismatoidpatnet54: An accurate ECG signal classification model using prismatoid pattern-based learning architecture. Symmetry, 13(10): 1914. https://doi.org/10.3390/sym13101914

[19] Mohonta, S.C., Motin, M.A., Kumar, D.K. (2022). Electrocardiogram based arrhythmia classification using wavelet transform with deep learning model. Sensing and Bio-Sensing Research, 37: 100502. https://doi.org/10.1016/j.sbsr.2022.100502

[20] Chen, H.Y., Chao, W.L. (2020). Fedbe: Making bayesian model ensemble applicable to federated learning. arXiv preprint arXiv:2009.01974. https://doi.org/10.48550/arXiv.2009.01974

[21] Shoham, N., Avidor, T., Keren, A., Israel, N., Benditkis, D., Mor-Yosef, L., Zeitak, I. (2019). Overcoming forgetting in federated learning on non-iid data. arXiv preprint arXiv:1910.07796. https://doi.org/10.48550/arXiv.1910.07796

[22] Zenke, F., Poole, B., Ganguli, S. (2017). Continual learning through synaptic intelligence. In International Conference on Machine Learning, the Netherlands, pp. 3987-3995.

[23] Sepahvand, M., Abdali-Mohammadi, F., Taherkordi, A. (2023). An adaptive teacher–student learning algorithm with decomposed knowledge distillation for on-edge intelligence. Engineering Applications of Artificial Intelligence, 117: 105560. https://doi.org/10.1016/j.engappai.2022.105560

[24] Sepahvand, M., Abdali-Mohammadi, F. (2023). Joint learning method with teacher–student knowledge distillation for on-device breast cancer image classification. Computers in Biology and Medicine, 155: 106476. https://doi.org/10.1109/ACCESS.2022.3181727

[25] Sepahvand, M., Abdali-Mohammadi, F. (2022). Overcoming limitation of dissociation between MD and MI classifications of breast cancer histopathological images through a novel decomposed feature-based knowledge distillation method. Computers in Biology and Medicine, 145: 105413. https://doi.org/10.1016/j.compbiomed.2022.105413

[26] Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., Rakovski, C. (2020). A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. Scientific Data, 7(1): 48. https://doi.org/10.1038/s41597-020-0386-x