A new vision of a simple 1D Convolutional Neural Networks (1D-CNN) with Leaky-ReLU function for ECG abnormalities classification

Lakhdari, Kheira and Saeed, Nagham ORCID logoORCID: https://orcid.org/0000-0002-5124-7973 (2022) A new vision of a simple 1D Convolutional Neural Networks (1D-CNN) with Leaky-ReLU function for ECG abnormalities classification. Intelligence-Based Medicine, 6. p. 100080. ISSN 2666-5212

http://dx.doi.org/10.1016/j.ibmed.2022.100080

Intelligence-Based Medicine 6 (2022) 100080

Contents lists available at ScienceDirect

Intelligence-Based Medicine

journal homepage: www.sciencedirect.com/journal/intelligence-based-medicine

A new vision of a simple 1D Convolutional Neural Networks (1D-CNN) with Leaky-ReLU function for ECG abnormalities classification

Kheira Lakhdari [a,*], Nagham Saeed [b]

[a] *Abou Bekr Belkaid University of Tlemcen, Tlemcen, Algeria*
[b] *School of Computing and Engineering, University of West London, London, UK*

A R T I C L E I N F O

Keywords:
AI
Cardiac abnormalities
Classification
CNN
Deep learning
ECG
Healthcare
Leaky-ReLU
MI
ReLU

A B S T R A C T

Artificial Intelligence (AI) is increasingly impacting the healthcare field, due to its computational power that reduces time, cost and efforts for both healthcare professionals and patients. Diagnosing cardiac abnormalities using AI represents a very attractive subject for both medical and technical professionals. Cardiac abnormalities are characterized by the ECG signal, which is known by its variable morphology and intense affection by noises and artifacts. In this context, the presented study aims to propose a simple yet efficient version of Convolutional Neural Networks (CNN) to classify those abnormalities. This version increases the ability to detect several heart rate arrhythmias and severe cardiac abnormalities based only on the original 1D format of the ECG signal, which reserve the main feature of this signal and can be very suitable for ready-to-use and real-time applications. The main used training datasets are the MIT-BIH arrhythmias and the PTB databases. The proposed architectures are mainly inspired by the most recent CNN models and introduce several modifications on functions and layers, such as the use of the Leaky-ReLU instead of the ReLU activation function. The results of the proposed model are varying from an accuracy of 97%–99% in classifying Normal (n), Supraventricular (s), Ventricular (v), Fusion of ventricular and normal (f), and noisy (q) beats, in addition to the Myocardial Infarction (MI) case. A continuous performance was achieved while testing the model on real data, and after its migration to real mobile devices.

1. Introduction

The numbers of cardiovascular diseases are increasing significantly [1,2]. The presence of this kind of diseases can increase the likelihood of other types of both chronic and communicable diseases [3,4]. Patients suffering from cardiac arrhythmias are more likely to be assisted permanently to avoid any dangerous complications [3-5]. On the other hand, intense advancements in technology are evolving every day [6-10], trying to handle critical situations that need immediate intervention, such as Myocardial Infarction (MI) [11-13]. This later is symbolized by a heart attack that can lead to a sudden death [14]. In its turn, AI is offering one of the most promising paths to create efficient solutions that may help chronic patients handle their cardiac abnormalities on a daily basis [15-17]. The Electrocardiograph (ECG) signal is a very important criterion for the diagnosis and the detection of cardiac abnormalities [14,18,19,20,21]. For that, classifying ECG data symbolizes a revolution in specialized medical services [18-24]. The idea of classifying the ECG is not very recent. For a long time, many studies have been devoted to introduce new methods, combine existing methods, and propose new scientific approaches [11,12,13,15,16,17,18,19,20,21,22,23].

The research in this domain is very broad; ranging from older mathematical methods, such as Fourier transforms, passing by the use of wavelets, and ending with the use of Machine Learning (ML) techniques and Deep Learning (DL) algorithms [15-36]. In 2015, Montano et al. [28] introduced a Neural Network (NN) composed of three layers. The layers were constructed of 12, 13, and 5 neurons, respectively. The dataset was trained using the “Levenberg Marquardt (MLA)"algorithm, while an accuracy of 98,67% was obtained. Another study was proposed in 2016 by Cruz et al. [29]. This study introduced two principal techniques, “Adaptive Neuro-Fuzzy Inference System (ANFIS)" and “Support Vector Machine (SVM)" coupled with “Mother Wavelet Daubechies (MWD)" method. The second technique was more efficient than the first one, with 95% accuracy compared to 85%, respectively. Zihlmann et al. [30] used two architectures of DL based on “Convolutional Neural Networks (CNN)" and “Convolutional Recurrent Neural Network (CRNN)", respectively. The CRNN was barely more accurate than the CNN, with 82.3% for the first compared to 81.2%. In 2018, Li et al. [31]

---

* Corresponding author.
E-mail address: kblakhdari@gmail.com (K. Lakhdari).

https://doi.org/10.1016/j.ibmed.2022.100080
Received 19 January 2022; Received in revised form 29 September 2022; Accepted 26 October 2022

and Zhai et al. [32] developed a new approach to handle the ECG signal, based on its transformation to a 2D format. This approach was followed to make the ECG data more adaptive to the morphology of CNN, regarding the good performances of this later on 2D images were used rather than 1D signals. Although, this method can neglect an important number of ECG features while transforming it to images. Following the same path, Li et al. used the “One-Hot Encoding” to obtain the 2D data, while Zhai et al. coupled the signals to a 2D matrix. The first approach attained an accuracy of 97% compared to 98.60% for the second one. In the same year, Savalia et al. [33] used the “Multi-Layer Perceptron (MLP)" in addition to the CNN, which gave an accuracy of 88.7%. In 2019, Ji et al. [34] used the “Short-Time Fourier Transform (STFT)" to represent the ECG signal on its frequency domain. This method was very efficient with an accuracy of 99%. Following the same path, Huang et al. [35] confirmed the high performances of the STFT transformation coupled with CNN models compared to 1D-CNN models without preprocessing.

In this same context, the main contributions of the proposed research can be resumed as follows:

- The introduction of a simple 1D-CNN model that suits directly the original 1D ECG format, which can avoid losing indicative features of this later.
- The deployment of two databases with the same model to include a larger number of cardiac abnormalities ranging from heart rate arrhythmias to MI.
- The use of several preprocessing steps to regularize the amount of data in each targeted class, to avoid the bias and the over-fitting effects.
- The choice of the most suitable structure, layers, parameters and functions, that can produce a highly performing model.
- The use of the most efficient tools and libraries to produce a simple, light and portable model, that can be easily exported to mobile environments.

This paper introduces firstly the global methodology of the proposed study. The next section is dedicated to the implementation process, where the paper describes the characteristics of the main used datasets, the steps followed to prepare this data and the architecture of the proposed model. It shows in the results section the significant achievements of this model, compared to the most relevant solution in the literature. Finally, this article resumes the presented work and introduces some of the related perspectives.

2. Methodology

In AI, the quality of data is crucial [36,37]. The used database must be as clear as possible to give the machine all the essential information for the generation of a good classification or prediction model. For this, filtering, preprocessing and also augmentation steps can be put in place before triggering the ML procedure. In addition, this data must include the greatest possible number of cases characterizing a class or a category, since the amount of data in addition to its quality can remarkably increase the performance of the generated model. The amount of available data must remain the same for every treated class or category. This later criterion is very important to avoid any biased result [37,38]. For that and basing on the massive literature reviews and reproduction of the available solutions, the implementation of presented work follows the procedure described in Fig. 1. It starts with several steps of visualization, preprocessing, filtering, segmentation and data augmentation. It continues by feeding the training dataset to the proposed model, validating this later and testing it to choose the most adequate functions and parameters. After obtaining the optimal model, the procedure ends up with exporting the model to the mobile environment. All the mentioned steps were executed after collecting suggestions, requirements and medical explanations from several conducted interviews with a cardiologist. This workflow will be detailed during the next section.

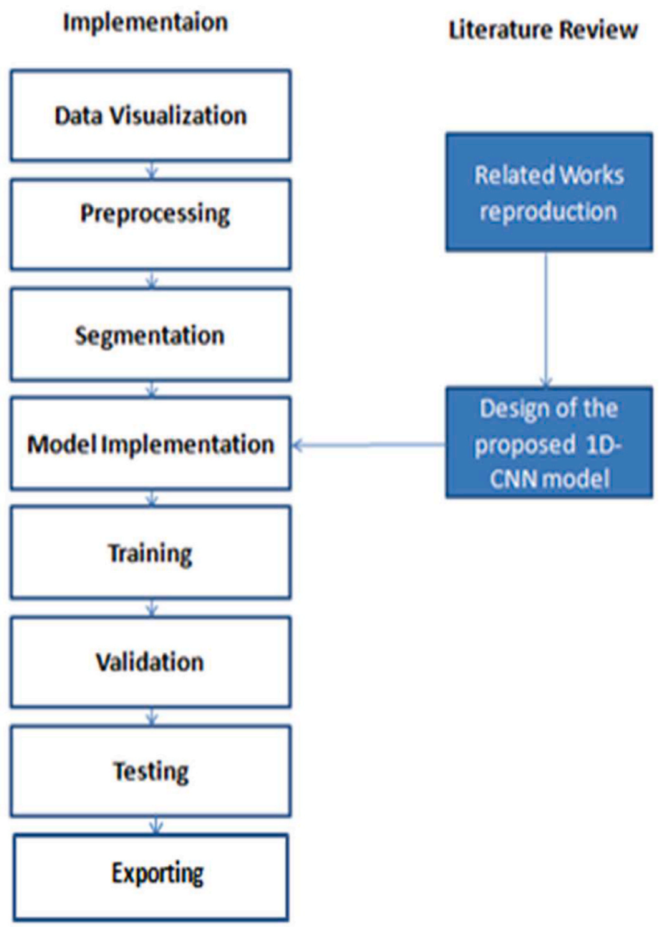


Fig. 1. Global methodology of the presented study.

3. Implementation

To cover a good number of cardiac abnormalities, the Implementation was processed on two main databases “MIT-BIH Arrhythmia Database” and the “PTB database”, as mentioned in Fig. 2. The preprocessing section is composed of some steps of annotation, manipulation and visualization. It is followed by the detection of the most indicative features of the ECG. Data augmentation techniques are affected just before splitting the data into the train and the validation datasets. The model's architecture is then implemented to proceed on the classification, the visualization of the results and the improvement of the final model. The testing of the improved model is finally affected on new data. Once the optimal model is achieved, a last step of exporting is executed, to add it to a simple mobile application to showcase its mobility feature. Fig. 2 resumes all the above steps, which are detailed, one by one in the following sections.

3.1. Experimental data

The ECG data used in this study belongs to the famous “MIT-BIH Arrhythmia Database”. This database contains recordings of the two-channel ambulatory ECG signals from 47 people [39]. The 47 subjects include a mixed population of inpatients and outpatients, with 60% and 40% of the recordings, respectively. The signals were digitized in 360 samples per second for each channel. The total number of annotations (explanatory symbols for signals, such as the N annotation for the normal signal) is 110,000 annotations. This database contains

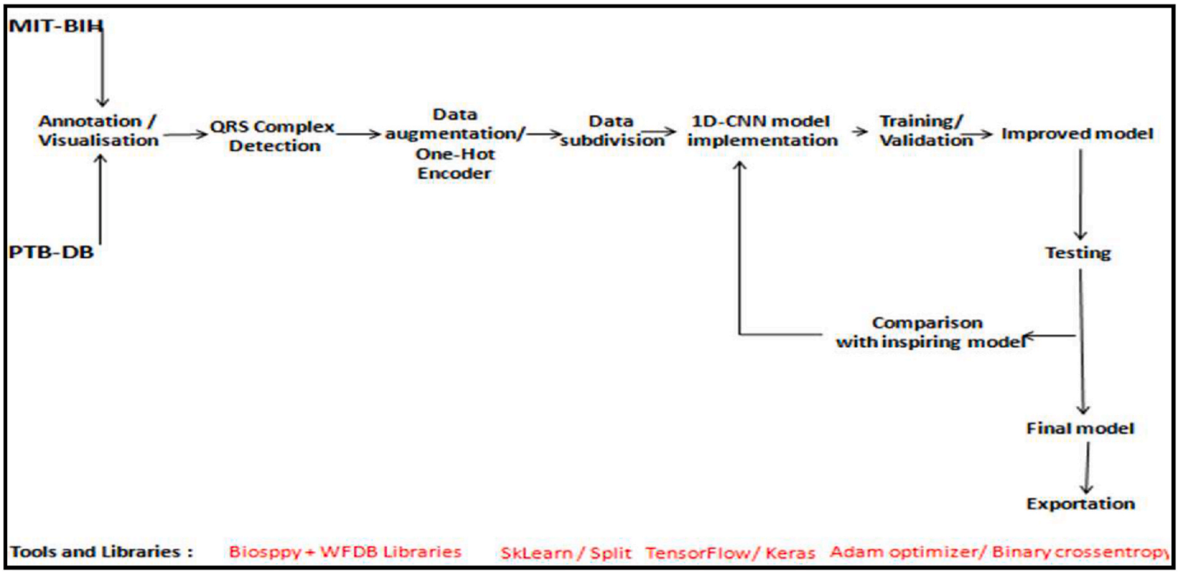


Fig. 2. A detailed architecture of the implementation process.

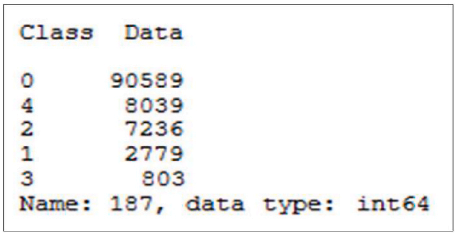
recordings of 1440 min in total. The MIT-BIH database remains the most famous ECG database, given its frequent scientific use. It represents the classes of normal ECG (which do not represent any abnormalities), premature contractions (arterial and ventricular), bundle branch block (right and left) and signal at an accelerated rate.

Another database integrated into this work is called “PTB-DB”. This database contains archives of ECG signals from 290 of mixed population, aged 17 to 87, with a rate of 1–5 recordings per person [40]. The recordings include the conventional 12 leads ECG along with the 3 Frank lead ECG. The studied classes can be summarized into two main classes representing the signals of healthy people and those of people with a variety of coronary heart disease such as Myocardial Infarction (the partial death of cells of the heart muscle, commonly known as heart attack [14]).

3.2. MIT-BIH data preprocessing

The very first step is to load the data needed to train the classification model. This loading is done directly from the selected databases and using the data manipulation tools offered by the Waveform Database (WFDB) library [41]. Since the first database is the one named MIT-BIH, loading the data highlights the five classes of the ECG signal it contains as described in Fig. 3. The classes are symbolized by the numbers from 0 to 4 and ordered according to their amount of data, decreasingly.

To fully understand and analyze this database, we have established some graphical presentations using programming tools and libraries such as Pandas and Numpy, to illustrate the subdivision of data according to these classes as shown in Fig. 4. The normal (n) signals in red take the majority with a percentage of 82.8%. The second class in green is called (q), it groups together random and noisy signals. The class known as (v) is that which represents rapid and abnormal ventricular contractions at the ventricles (visible on the “QRS” complexes of the ECG [42]), while (s) is the class which symbolizes anomalies on supra-ventricular contractions (between ventricles and atria). This later represents cardiac arrhythmias such as Tachycardia (elevated heart rate [43]). The last class (f) represents signals forming a fusion of normal and ventricular signals.


```
Class  Data

0      90589
4       8039
2       7236
1       2779
3        803
Name: 187, data type: int64
```

Fig. 3. MIT-BIH data format and size.

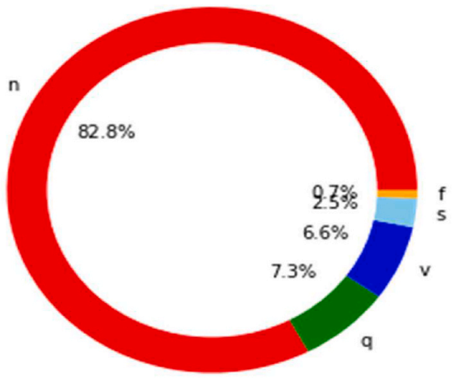


Fig. 4. The repartition of the five classes presented in the MIT-BIH database before preprocessing.

Illustrations of these classes are plotted for visualization in Fig. 5. The heterogeneity of this database requires data augmentation and preprocessing to avoid any biased decisions. The data augmentation step will regularize the amount of data in each class to be equal. This step is only possible after detecting the QRS complex in the ECG signals, segment the signals according to this complex and augmenting them where necessary. The QRS detection is detailed in the next section.

3.2.1. QRS complex detection

The QRS complex represents one of the most important characteristics of the ECG, which is the peak or the highest value of this signal. This complex is essential for the detection of a large number of cardiac abnormalities [44]. The chosen method for this detection was introduced in a recent scientific study by Jun et al. [45]. The choice of this method resides mainly on the good performances of its approach, in addition to the ease of its integration thanks to its segmentation function directly included in the Biosppy library [44,46]. The sampling frequency used is 1000 Hz. The results of this method are illustrated in Fig. 6.

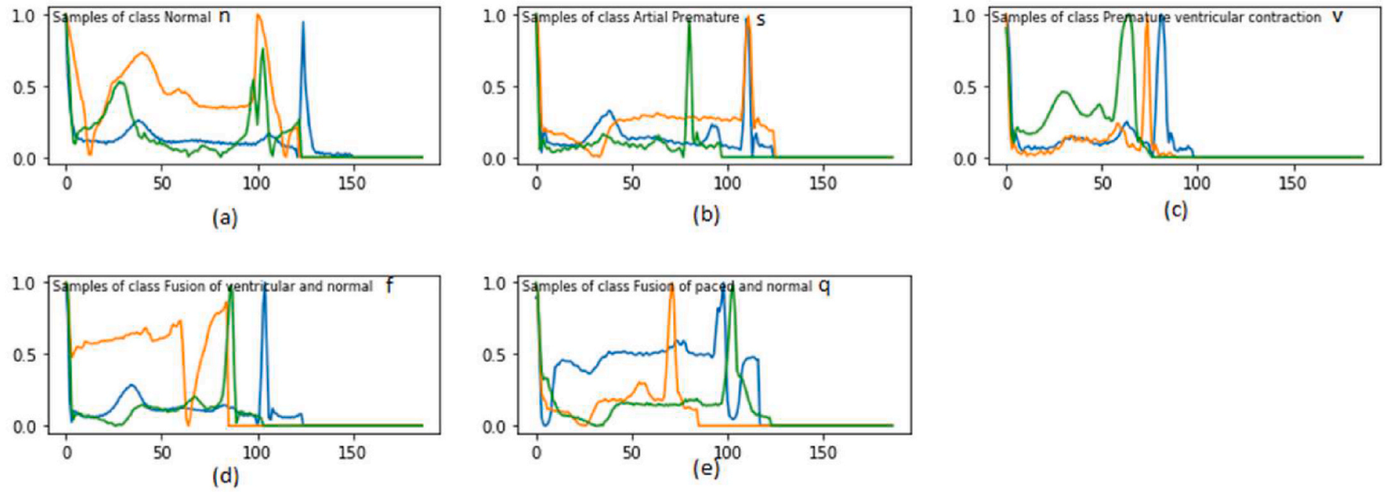


Fig. 5. Visualization of the five classes of the MIT-BIH database before preprocessing. (a) Samples of the Normal class “n". (b) Samples of the Supraventricular ectopic class “s".(c) Samples of the Ventricular ectopic class “v". (d) Samples of the Fusion of ventricular and normal class “f". (e) Samples of paced, noisy and unknown beats “q".

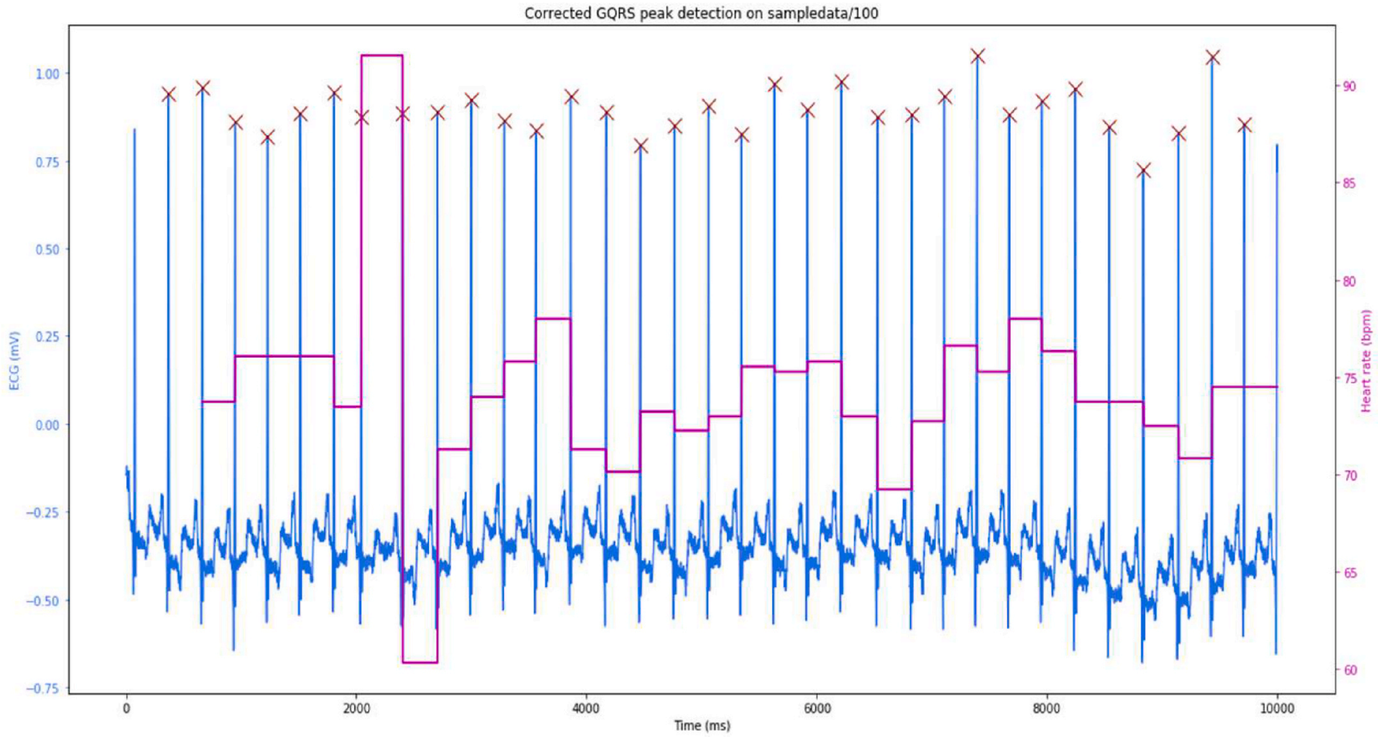


Fig. 6. QRS complex detection using the Biosppy library.

Another simpler approach consists in directly using the function called “gqrs_detect” proposed by the WFDB library. The results of this detection method are very favorable as shown in Fig. 7 and Fig. 8. This step is essential to segment the signals according to the detected QRS complex and prepare them for the next step, which is data augmentation.

3.2.2. Data augmentation

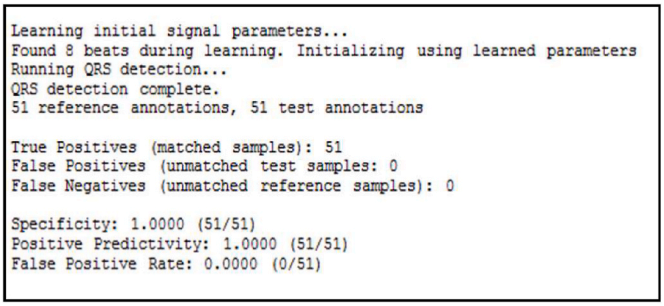
This step consists of using an encoding method called “One-hot Encoder”. This method is based on the principle of transforming all the values of the database into vectors of size “188″ bits. These vectors consist of “0″ and a single “1". The location of the “1″ differentiates each vector from the others [31].

3.2.3. Data subdivision

A subdivision of the whole data into two groups is then implemented. The first group represents the training dataset. This later is used to train the machine with a percentage of more than 96% of the initial data. About 4% of this training dataset is used for cross-validation. The second group is used to test the designed model with a percentage of about 4% of the initial data.

3.3. PTB-DB data preprocessing

The second database contains a similar format to the first one with less data as shown in Fig. 9. It is composed of two main classes which are signals for normal persons and signals presenting MI arrhythmia as



```
Learning initial signal parameters...
Found 8 beats during learning. Initializing using learned parameters
Running QRS detection...
QRS detection complete.
51 reference annotations, 51 test annotations

True Positives (matched samples): 51
False Positives (unmatched test samples: 0
False Negatives (unmatched reference samples): 0

Specificity: 1.0000 (51/51)
Positive Predictivity: 1.0000 (51/51)
False Positive Rate: 0.0000 (0/51)
```

Fig. 7. Notebook output after the QRS detection in numbers using the WFDB library.

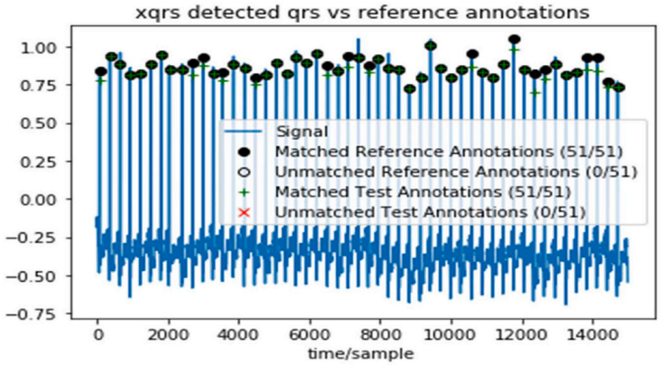
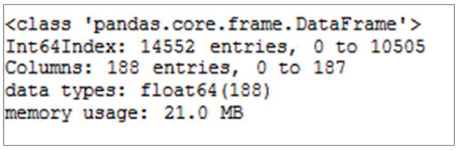


Fig. 8. Visualization of the QRS detection using the WFDB library.



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14552 entries, 0 to 10505
Columns: 188 entries, 0 to 187
data types: float64(188)
memory usage: 21.0 MB
```

Fig. 9. PTB-DB data format and size.

visualized in Fig. 10. The next steps of preprocessing and data augmentation remain the same for this database as the first one.

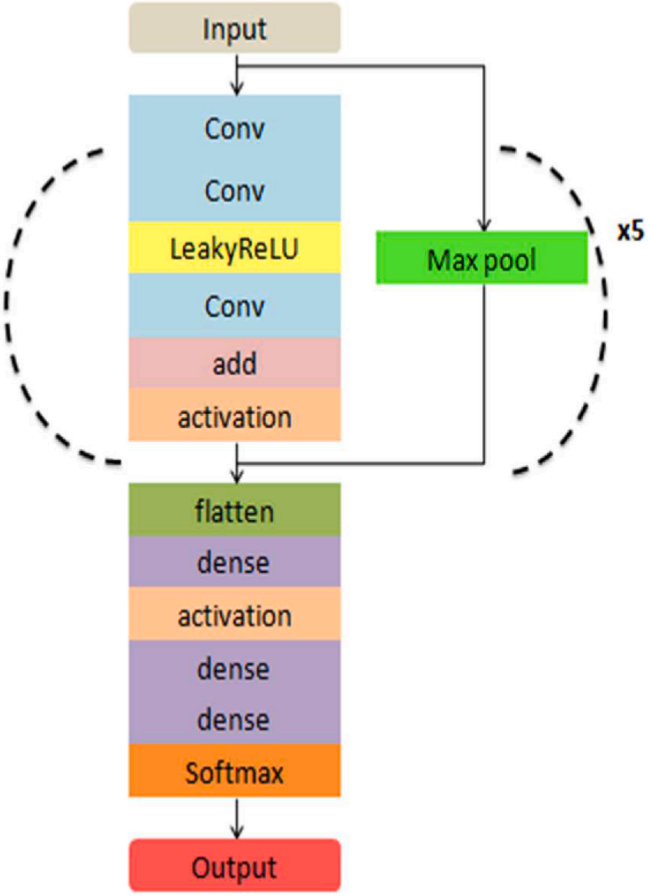


Fig. 11. Architecture of the proposed 1D-CNN model.

3.4. Model implementation on MIT-BIH dataset

The declaration of the main architecture of this model is made by designating the number as well as the type of layers used in Fig. 11. The 1D-CNN model is inspired by the study of Kachuee et al. [47], with several changes on functions and layers. It is composed of an initial convolution layer, then a block of another convolution layer and a “Leaky-ReLU” activation function. This particular function assigns a variant for negative values such that neurons never time out [48-50], as mentioned in (1). Unlike the “ReLU” function used in Kachuee et al. model, which forces the negative values to 0 [47,48], as described in (2).

$$f(x) = [1 * (x < 0) * \alpha x] + [1 * (x \geq 0) * x] \tag{1}$$

where α is a small constant.

$$f(x) = max\ (0, x) \tag{2}$$

This block is also composed of another convolution layer, addition layer, an activation layer and a last layer of “Max pooling”. This main block is repeated 5 times. Towards the end, a layer of “Flatten”, another of “Dense”, an activation layer in addition to two other layers of “Dense” and a last of “Softmax” are added. Table 1 describes this architecture in a more detailed way, mentioning the shape of the input data in each level and the number of parameters.

3.5. Model implementation on PTB-DB dataset

The classification model deployed for this database is the same 1D-

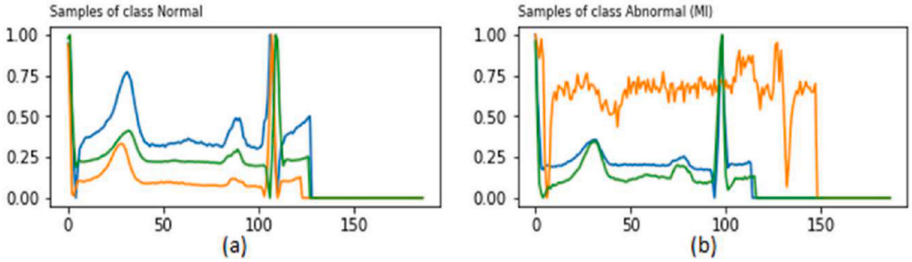


Fig. 10. Visualization of the two classes of the PTB-DB database before preprocessing. (a) Samples of the normal class. (b) Samples of the MI class.

Table 1
A detailed architecture of the proposed 1D-CNN model.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 187, 1)	0	
conv1d_1 (Conv1D)	(None, 183, 32)	192	input_1[0][0]
conv1d_2 (Conv1D)	(None, 183, 32)	5152	conv1d_1[0][0]
leaky_re_lu_1 (LeakyReLU)	(None, 183, 32)	0	conv1d_2[0][0]
conv1d_3 (Conv1D)	(None, 183, 32)	5152	leaky_re_lu_1[0][0]
add_1 (Add)	(None, 183, 32)	0	conv1d_3[0][0]
conv1d_1[0][0]			
activation_1 (Activation)	(None, 183, 32) 0	add_1[0][0]	
max_pooling1d_1 (MaxPooling1D)	(None, 90, 32)	0	activation_1[0][0]
conv1d_4 (Conv1D)	(None, 90, 32)	5152	max_pooling1d_1[0][0]
leaky_re_lu_2 (LeakyReLU)	(None, 90, 32)	0	conv1d_4[0][0]
conv1d_5 (Conv1D)	(None, 90, 32)	5152	leaky_re_lu_2[0][0]
add_2 (Add)	(None, 90, 32)	0	conv1d_5[0][0]
max_pooling1d_1[0][0]			
activation_2 (Activation)	(None, 90, 32)	0	add_2[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 43, 32)	0	activation_2[0][0]
conv1d_6 (Conv1D)	(None, 43, 32)	5152	max_pooling1d_2[0][0]
leaky_re_lu_3 (LeakyReLU)	(None, 43, 32)	0	conv1d_6[0][0]
conv1d_7 (Conv1D)	(None, 43, 32)	5152	leaky_re_lu_3[0][0]
add_3 (Add)	(None, 43, 32)	0	conv1d_7[0][0]
max_pooling1d_2[0][0]			
activation_3 (Activation)	(None, 43, 32)	0	add_3[0][0]
max_pooling1d_3 (MaxPooling1D)	(None, 20, 32)	0	activation_3[0][0]
conv1d_8 (Conv1D)	(None, 20, 32)	5152	max_pooling1d_3[0][0]
leaky_re_lu_4 (LeakyReLU)	(None, 20, 32)	0	conv1d_8[0][0]
conv1d_9 (Conv1D)	(None, 20, 32)	5152	leaky_re_lu_4[0][0]
add_4 (Add)	(None, 20, 32)	0	conv1d_9[0][0]
max_pooling1d_3[0][0]			
activation_4 (Activation)	(None, 20, 32)	0	add_4[0][0]
max_pooling1d_4 (MaxPooling1D)	(None, 8, 32)	0	activation_4[0][0]
conv1d_10 (Conv1D)	(None, 8, 32)	5152	max_pooling1d_4[0][0]
leaky_re_lu_5 (LeakyReLU)	(None, 8, 32)	0	conv1d_10[0][0]
conv1d_11 (Conv1D)	(None, 8, 32)	5152	leaky_re_lu_5[0][0]
add_5 (Add)	(None, 8, 32)	0	conv1d_11[0][0]
max_pooling1d_4[0][0]			
activation_5 (Activation)	(None, 8, 32)	0	add_5[0][0]
max_pooling1d_5 (MaxPooling1D)	(None, 2, 32)	0	activation_5[0][0]
flatten_1 (Flatten)	(None, 64)	0	max_pooling1d_5[0][0]
dense_1 (Dense)	(None, 32)	2080	flatten_1[0][0]
activation_6 (Activation)	(None, 32)	0	dense_1[0][0]
dense_2 (Dense)	(None, 32)	1056	activation_6[0][0]
dense_3 (Dense)	(None, 5)	165	dense_2[0][0]
softmax_1 (Softmax)	(None, 5)	0	dense_3[0][0]

Table 2
Results of the *kachuee* et al. model on MIT-BIH dataset.

Class	precision	recall	f1-score	support
0	0.82	0.99	0.90	800
1	1.00	0.84	0.91	800
2	0.90	0.96	0.93	800
3	0.99	0.87	0.93	800
4	0.99	0.99	0.93	800
accuracy			0.93	4000
macro avg	0.94	0 .93	0.93	4000
weighted avg	0.94	0.93	0.93	4000

ranking-based average precision: 0.964
Ranking loss: 0.020
Coverage_error: 1.082

Table 3
Results of the proposed 1D-CNN model on MIT-BIH test dataset.

Class	precision	recall	f1-score	support
0	0.84	1.00	0.91	800
1	1.00	0.86	0.92	800
2	0.93	0.96	0.94	800
3	0.99	0.92	0.95	800
4	0.99	0.99	0.99	800
accuracy			0.94	4000
macro avg	0.95	0.94	0.94	4000
weighted avg	0.95	0.94	0.94	4000

ranking-based average precision: 0.971
Ranking loss: 0.017
Coverage_error: 1.068

Table 4
Results of the proposed 1D-CNN model on PTB-DB test dataset.

Class	precision	recall	f1-score	support
0.0	0.99	0.97	0.98	634
1.0	0.99	1.00	0.99	1549
accuracy			0.99	2183
macro avg	0.99	0.98	0.99	2183
weighted avg	0.99	0.99	0.99	2183

CNN model used previously, with only one small change in the activation function which is replaced by the famous "ReLU" function for this case, regarding the small amount of data in the PTB-DB dataset, which is not affected by the dying effect of the "ReLU" function. The learning of this model is carried out over 30 epochs.

3.6. Evaluation metrics

The achieved performance of the proposed models was measured using the "Accuracy" and the "Loss" indicators, represented in (3) and (4), respectively. While True Positive (TP) refers to the abnormalities correctly detected, True Negative (TN) represents the normal cases correctly detected, False Positive (FP) refers to the normal cases wrongly detected as abnormalities, and False Negative (FN) represents the abnormalities wrongly detected as normal cases. The Loss function was based on the Binary Cross-entropy function as detailed in (4).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Loss(X_i, Y_i) = \sum_{j=1}^{c} y_{ij} * log(p_{ij}) \tag{4}$$

Where Y_i is one-hot encoded target vector ($y_{i1}, y_{i2}, ...y_{ic}$) and $P_{ij} = f(X_i)$ is the probability that the i_{th} element is in class j

The performance indicators were not limited to the accuracy and the loss rates. Three other indicators were added, referring to the "Precision", the "Recall" and the "F1-score", represented in (5), (6) and (7), respectively.

$$Precision = \frac{TP}{(TP + FP)} \tag{5}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{6}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \tag{7}$$

3.7. Model exportation

After improving the model and validating it, the last step of the presented study consists of its migration to a simple mobile application. The proposed 1D-CNN model was primarily implemented using "Keras " [51] and TensorFlow [52] libraries, in order to facilitate its exportation. This later requires firstly the conversion of the model to a TensorFlow Lite (.tflite) format and then its integration to the mobile application environment. For our case, the mobile application was developed in Android Studio [53] environment, to be used on Android devices. Although, the integration of the tflite version of the 1D-CNN model is also possible for a wide range of devices and environments.

4. Results

The proposed 1D-CNN model trained on MIT-BIH database produces very good results in terms of accuracy and loss rate as seen in Fig. 12 and Fig. 13, during its training over a number of 75 epochs and its validation by the rest of the non-labeled data (4% of the training dataset).

The overall accuracy and loss performance results produced by the proposed 1D-CNN model on the PTB-DB dataset are by far very advantageous, approaching to 99% and 0%, respectively as shown in Fig. 14 and Fig. 15. Against an accuracy of only 95.9% of the Kachuee et al. model.

The results shown in Tables 2, 3 and 4 have proven the high performance of the presented approach during the testing phase. The error and loss rates are also reduced in this approach. The confusion matrix in Fig. 16 illustrates in more details the performances according to each class of the MIT-BIH database.

The proposed 1D-CNN model on MIT-BIH test dataset achieved an average of precision of 97%, of recall and F1-score of 94%, while it achieved an average of 99% for all the three indicators on the PTB-DB test dataset, against an average of just 96% of precision and 93% of recall and F1-score for the Kachuee et al. model.

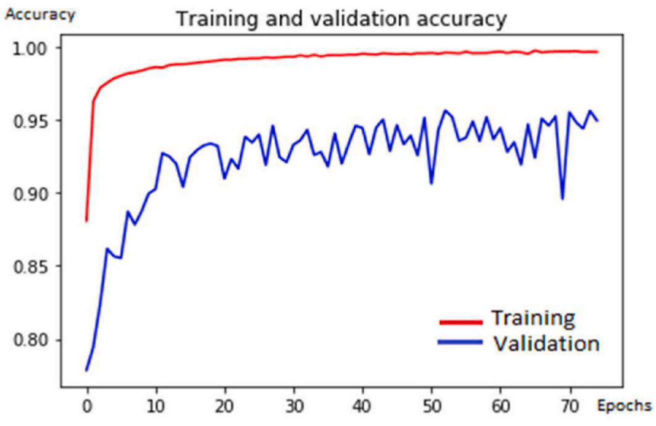


Fig. 12. Accuracy of the 1D-CNN model on MIT-BIH dataset.

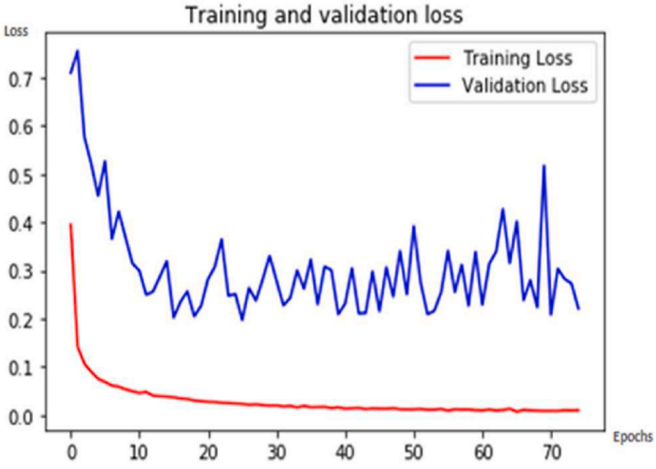


Fig. 13. Loss of the 1D-CNN model on MIT-BIH dataset.

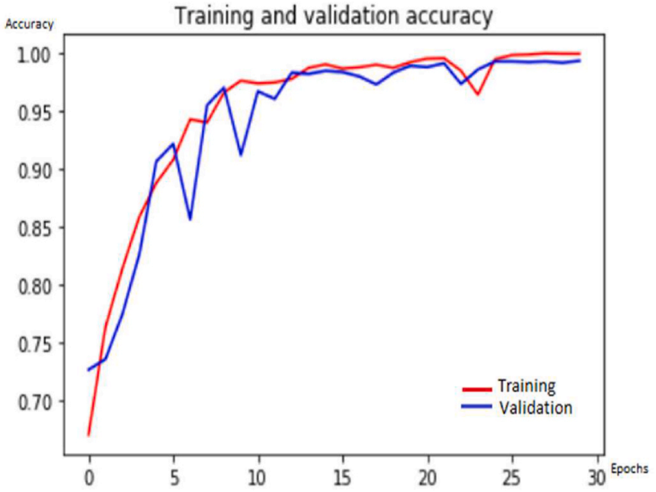


Fig. 14. Accuracy of the 1D-CNN model on PTB-DB dataset.

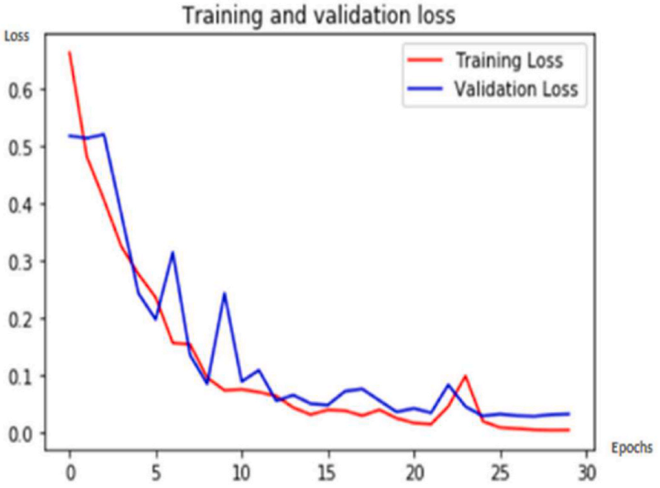
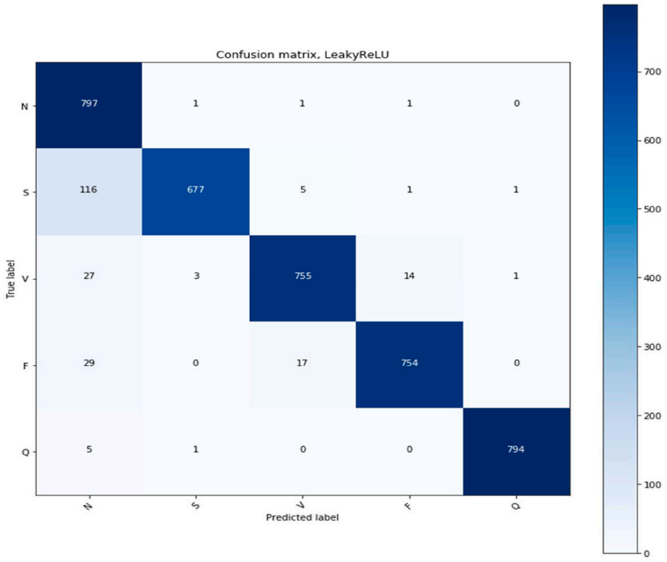


Fig. 15. Loss of the 1D-CNN model on PTB-DB dataset.

These results present a higher performance compared to the most recent studies in this field, namely the study presented by Niu et al. [54].

The performance of the 1D-CNN remains the same after its migration to the mobile environment. It proved its capacity to classify different



Confusion matrix, LeakyReLU

True label \ Predicted label	N	S	V	F	Q
N	797	1	1	1	0
S	116	677	5	1	1
V	27	3	755	14	1
F	29	0	17	754	0
Q	5	1	0	0	794

Fig. 16. Confusion matrix of the proposed 1D-CNN model on MIT-BIH dataset.

cases as shown in Fig. 17 and Fig. 18, based on new testing data. This later was introduced to the mobile model in a format of stocked files on the device, only for testing purposes. In the future, the model can be successfully integrated in a more complex application to classify real-time ECG data coming from medical sensors. The size of the exported 1D-CNN model equals to only 468 KB.

5. Discussion

The main particularity of the proposed 1D-CNN model resides on its simplicity, which make it directly suitable for the ECG data format instead of changing this format to make it more adaptive to the CNN nature. This aims to reserve the original ECG format, which automatically reserves its features and most indicative parts. Unlike the most advanced research following the 2D and the 3D trends which neglect the negative effect of these transformations on the ECG valuable data.

This particularity lies also on the use of the activation function named Leaky-ReLU, unlike the basic model and most scientific works that use the ReLU function. The main point of difference between these two approaches is that neurons do not die in our case. It is because of the Leaky-ReLU effect which assigns a variant to negative values to reserve the neurons and in consequence the valuable data. Unlike the ReLU function which gives a 0 to these values, kills the neurons, and may neglect some of the valuable data. This later can significantly affect the learning process for large amounts of data. Although, it is not very harmful for small datasets.

The preprocessing carried out on the original data, included the segmentation and the data augmentation to normalize the amount of data in each class, in addition to the Dense layers also improved these results, thanks to their effect of regularization on the output in each level to avoid the over-fitting effect and any biased decision.

The 1D-CNN model has also proven to be particularly effective on test dataset (new and non-labeled data), and after exportation to the mobile environment.

However, the proposed 1D-CNN model needs to be tested on much more ECG data, collected continuously and in real-time to evaluate its time consumption and ability to carry out real ECG data, which can

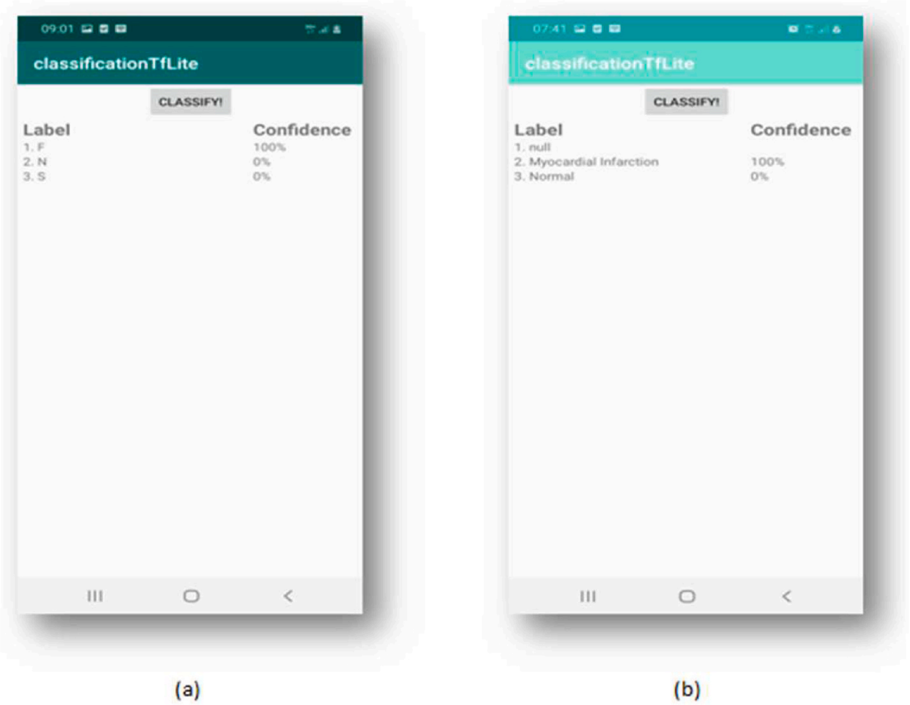
Label
1.
2.
3.

Label 1. F 2. N 3. S
Confidence 100% 0% 0%

Label 1. V 2. N 3. S
Confidence 43% 28% 18%

Label 1. Q 2. V 3. S
Confidence 53% 35% 8%

Fig. 17. Examples of the ECG classification results on a simple mobile application. (a) The application before classification. (b) The detection of Ventricular ectopic class. (c) The detection of a Fusion of ventricular and normal class. (d) The detection of paced, noisy and unknown beats.



(a) classificationTfLite

Label	Confidence
1. F	100%
2. N	0%
3. S	0%

(b) classificationTfLite

Label	Confidence
1. null	
2. Myocardial Infarction	100%
3. Normal	0%

Fig. 18. Examples of the ECG classification results on a simple mobile application. (a) The classification of a Fusion of normal and ventricular beats. (b) The detection of MI abnormality.

change continuously during the day. These types of tests can be conducted in future works using wearable ECG sensors.

6. Conclusion

Selecting CNN networks to classify ECG signals is a challenge in the truest sense of the word, since this type of neural networks was designed especially to classify images with two-dimensional values, unlike the ECG signal. For that, the current trend is to transform the ECG signal to 2D and even 3D images, to make it more suitable for CNNs. Despite this trend, the presented study proved that it is possible and even more effective to deploy a simple 1D-CNN model. This later can be easier to deploy and does not require the transformation of the ECG signal, which may affect its main features.

The quality of the chosen data, the steps of preprocessing, filtering and augmentation, in addition to the choice of the activation function, the number and the type of layers play a major role in improving the overall performance of the classification model. These methods have given rise to algorithms for high-quality classification, which are able to detect ECG abnormalities with a very high precision, recall and accuracy. This performance remained the same when tested on new and non-labeled data.

The presented work is open to be improved to handle more cardiac abnormalities because of its ability to automatically learn the features basing on the provided dataset. While the quality of this model can be enhanced by merging available datasets or creating a larger one.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ibmed.2022.100080.

References

[1] World Health Organization. Cardiovascular diseases (CVDs). Accessed on: June 23, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds; 2021.

[2] Mena LJ, Felix VG, Melgarejoand JD, Maestre GE. Mobile personal health monitoring for automated classification of electrocardiogram signals in elderly, vol. 2018. Computational and Mathematical Methods in Medicine,Hindawi; 2018. Article ID 9128054.

[3] Huygens MWJ, Vermeulen J, Swinkels ICS, Friele RD, Van Schayck OCP, De Witte LP. Expectations and needs of patients with a chronic disease toward self-management and eHealth for self-management purposes. BMC Health Serv Res Jul. 2016;19.

[4] Milani RV, Bober RM, Lavie CJ. The role of technology in chronic disease care. In: Progress in cardiovascular diseases. Elsevier; 2016. p. 579–83. https://www.sciencedirect.com/science/article/pii/S0033062016300019.

[5] Wilhide III CC, Peeples MM, Kouyaté RCA. Evidence-based m-health chronic disease mobile app intervention design: development of a framework. Online J. Pub. Health Inf. Feb. 2016;5(1). https://doi.org/10.2196/resprot.4838.

[6] Metcalf D, Milliard STJ, Gomez M, Schwartz M. Wearables and the internet of things for health : wearable, interconnected devices promise more efficient and comprehensive health care. IEEE Pulse Oct. 2016;7:35–359.

[7] Ching T, Himmelstein DS, Beaulieu-Jones BK, et al. Opportunities and obstacles for deep learning in biology and medicine. Royal Society; 2018. 01- 47, https://royalsocietypublishing.org/doi/10.1098/rsif.2017.0387.

[8] Bote-Curiel L, Muñoz-Romero S, Gerrero-Curieses A, Rojo-Álvarez JL. Deep learning and big data in healthcare: a double review for critical beginners. MDPI 6 jun. 2019;1:2–34. https://www.mdpi.com/2076-3417/9/11/2331.

[9] Pramanik PKD, Upadhyaya BK, Pal S, Pal T. Internet of things, smart sensors, and pervasive systems: enabling connected and pervasive healthcare. Chapter, Healthcare Data Analytics and Management, Elsevier Inc; Nov. 2018. https://doi.org/10.1016/B978-0-12-815368-0.00001-4.

[10] Eckert M, Volmerg JS, Friedrich CM. Augmented reality in medicine: systematic and bibliographic review. JMIR MhealthUhealth Apr. 2019;7(4). https://doi.org/10.2196/10967.

[11] Hammad M, Alkinani MH, Gupta BB, et al. Myocardial infarction detection based on deep neural network on imbalanced data. Multimed Syst 2021. https://doi.org/10.1007/s00530-020-00728-8.

[12] Jian JZ, Ger TR, Lai HH, Ku CM, Chen CA, Abu PAR, Chen SL. Detection of myocardial infarction using ECG and multi-scale feature concatenate. Sensors 1906, Mar;21(5):2021.

[13] Cho Y, Kwon Jm, Kim KH, et al. Artificial intelligence algorithm for detecting myocardial infarction using six-lead electrocardiography. Sci Rep 2020;10:20495. https://doi.org/10.1038/s41598-020-77599-6.

[14] Medgeeks, Champigny D. How to read an ECG - medgeeks [Online] Available, https://medgeeks.co/articles/how-to-read-an-ecg. [Accessed 9 August 2021].

[15] Ragnarsson F. A portable ECG system for real-time arrhythmia classification - smartphone implementation of a modular convolutional network applied on ECG-signals. Uupsala University; Jul. 2019. http://uu.diva-portal.org/smash/get/diva2:1345590/FULLTEXT01.pdf.

[16] Utomo TP, Nuryani N, Darmanto. QRS peak detection for heart rate monitoring on Android smartphone. J Phys: Conf. Ser. 2017 https://iopscience.iop.org/article/10.1088/1742-6596/909/1/012006/pdf.

[17] Luz EJS, Schwartz WR, Cámara-Chávez G, Menotti D. ECG-based heartbeat classification for arrhythmia detection: a survey. Comput Methods Progr Biomed Apr. 2016;127:144–64. https://doi.org/10.1016/j.cmpb.2015.12.008.

[18] Picariello F, Iadarola G, Balestrieri E, Tudosa I, De Vito L. A novel compressive sampling method for ecg wearable measurement systems. Measurement 2021;167.

[19] Cai J, Zhou G, Dong M, Hu X, Liu G, Ni W. Real-time arrhythmia classification algorithm using time-domain ECG feature based on FFNN and CNN. Mathematical Probl Eng 2021;2021. https://doi.org/10.1155/2021/6648432. Article ID 6648432, 17 pages.

[20] Ribeiro AH, Ribeiro MH, Paixão GMM, et al. Automatic diagnosis of the 12-lead ECG using a deep neural network. Nat Commun 2020;11(1760):2020. https://doi.org/10.1038/s41467-020-15432-4.

[21] Hao C, Wibowo S, Majmudar M, Singh Rajput K. Spectro-temporal feature based multi-channel convolutional neural network for ecg beat classification. In: 41st annual international conference of the IEEE engineering in medicine and biology society (EMBC). IEEE; 2019. p. 5642–5.

[22] Oliveira AT, GO Nobrega E, et al. A novel arrhythmia classification method based on convolutional neural networks interpretation of electrocardiogram images. In: IEEE International conference on industrial technology. Piscataway, NJ; 2019.

[23] Izci E, Ozdemir MA, Degirmenci M, Akan A. Cardiac arrhythmia detection from 2d ecg images by using deep learning technique. In: 2019 medical technologies congress (TIPTEKNO),IEEE; 2019. p. 1–4.

[24] Hannun AY, Rajpurkar P, Haghpanahi M, et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. Nat Med Jan. 2019;25:65–9. https://doi.org/10.1038/s41591-018-0268-3.

[25] Houssein EH, Kilany M, Hassanien AE. ECG signals classification: a review. Int J Med Eng Inf Jan. 2017;5(4):376–96. https://doi.org/10.1504/IJIEI.2017.10008807.

[26] Ebrahimi Z, Loni M, Daneshtalab M, Gharehbaghi A. A review on deep learning methods for ECG arrhythmia classification. Expert Syst Appl X 2020;7(Sept):174–82. Elsevier, https://www.sciencedirect.com/science/article/pii/S2590188520300123.

[27] Tan WW, Foo CL, Chua TW. Type-2 fuzzy System for ECG arrhythmic classification. In: 2007 IEEE international fuzzy systems conference, London; 2007. p. 1–6. https://doi.org/10.1109/FUZZY.2007.4295478.

[28] Montaño VD, Linsangan N, Ramos ND. ECG signal classification for remote area patients using artificial neural networks in smartphone. Int J Computer Electrical Eng 2015;7:215–22. https://doi.org/10.17706/ijcee.2015.7.3.215-222.

[29] Cruz CIM, Marasigan JP, Perez APG, Pillejera JEV, Veron NP, dela Cruz AR. A comparative study between DWT-ANFIS and DWT-SVM in ECG classification. In: 2016 IEEE region 10 conference (TENCON), Singapore; 2016. p. 2052–5. https://doi.org/10.1109/TENCON.2016.7848386.

[30] Zihlmann M, Perekrestenko D, Tschannen M. Convolutional recurrent neural networks for electrocardiogram classification. 2017 Computing in Cardiology (CinC), Rennes 2017:1–4. https://doi.org/10.22489/CinC.2017.070-060.

[31] Li J, Si Y, Xu T, Jiang S. Deep convolutional neural network based ECG classification System using information fusion and one-hot encoding techniques. Math Probl Eng 2018;2018. https://doi.org/10.1155/2018/7354081. Article ID 7354081, 10 pages.

[32] Zhai X, Tin C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. IEEE Access 2018;6:27465–72. https://doi.org/10.1109/ACCESS.2018.2833841.

[33] Savalia S, Emamian V. Cardiac arrhythmia classification by multi layer Perceptron and convolution neural networks. Bioengineering May 2018;5(2). https://doi.org/10.3390/bioengineering5020035.

[34] Ji Y, Zhang S, Xiao W. Electrocardiogram classification based on faster regions with convolutional neural network. Sensors Jun. 2019;19(11). https://doi.org/10.3390/s19112558. https://www.mdpi.com/journal/sensors.

[35] Huang J, Chen B, Yao B, He W. ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network. IEEE Access 2019;7:92871–80. https://doi.org/10.1109/ACCESS.2019.2928017.

[36] Saeed N, Husamaldin L. Big data characteristics (V’s) in industry. Iraqi J Industrial Res 2021;8(Issue 1). https://doi.org/10.53523/ijoirVoI8I1ID52. Pages1-9.

[37] Husamaldin L, Saeed N. Big data analytics correlation taxonomy. formation 2020;11(Issue 1):17. https://doi.org/10.3390/info11010017.

[38] McDuff D, Cheng R, Kapoor A. Identifying bias in AI using simulation. Sept. 2018. arXiv:1810.00471.

[39] Moody GB, Mark RG. The impact of the MIT-BIH arrhythmia database. IEEE Eng Med Biol Mag May-June 2001;20(3):45–50. https://doi.org/10.1109/51.932724.

[40] Goldberger A, et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation [Online] 2000;101(23). e215–e220.

[41] archive.physionet.org. The WFDB software package - PhysioNet [Online]. Available, https://archive.physionet.org/physiotools/wfdb.shtml. [Accessed 18 August 2021].

[42] Sherathia D, Patel VP. Classification of ECG beats based on fuzzy inference System. Int J Sci Eng Technol Res(IJSETR) May 2017:835–40. 06, http://ijsetr.org/wp-content/uploads/2017/05/IJSETR-VOL-6-ISSUE-5-835-840.pdf. 05.

[43] Mousavi S, Afghah F. Inter- and intra- patient ECG heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach. In: Icassp 2019 - 2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), brighton, United Kingdom; 2019. p. 1308–12. https://doi.org/10.1109/ICASSP.2019.8683140.

[44] I Christov I. Real time electrocardiogram QRS detection using combined adaptive threshold. BioMedical Engineering OnLine; Aug. 2004. p. 1–9. http://www.biomedical-engineering-online.com/content/3/1/28.

[45] Jun TJ, Minh Nguyen H, Kang D, Kim D, Kim D, Kim YH. ECG arrhythmia classification using a 2-D convolutional neural network. arXiv:1804.06812v1, https://arxiv.org/abs/1804.06812; 18 April 2018.

[46] biosppy.read the docs.io. Welcome to BioSPPy — BioSPPy 0.6.1 documentation, 2015-2018 [Online]. Available, https://biosppy.readthedocs.io/en/stable/. [Accessed 19 August 2021].

[47] Kachuee M, Fazeli S, Sarrafzadeh M. ECG heartbeat classification: a deep transferable representation. In: 2018 IEEE international conference on healthcare informatics (ICHI), New York, NY; 2018. p. 443–4. https://doi.org/10.1109/ICHI.2018.00092.

[48] Mastromichalakis S. ALreLU: a different approach on Leaky ReLU activation function to improve Neural Networks Performance. 2020.

[49] Parisi L, Ma R, RaviChandran N, Lanzillotta M. Hyper-sinh: an accurate and reliable function from shallow to deep learning in TensorFlow and keras. https://arxiv.org/abs/2011.07661. [Accessed 15 November 2020].

[50] L. Parisi, D. Neagu, R. Ma, F. Campean, "QReLU and m-QReLU: two novel quantum activation functions to aid medical diagnostics", arXiv:2010.08031v1, 15 Oct. 2020.https://arxiv.orgabs/2010.08031.

[51] Keras. The Python deep learning API [Online]. Available, https://keras.io/. [Accessed 19 August 2021].

[52] www.tensorflow.org, with Tensorflow [Online]. Available, https://www.tensorflow.org/?hl=fr. [Accessed 19 August 2021].

[53] www.android.com,. developers, Building anything on Android [Online]. Available, https://developer.android.com/. [Accessed 19 August 2021].

[54] Niu L, Chen C, Liu H, Zhou S, M. A deep-learning approach to ECG classification based on adversarial domain adaptation. Healthcare 2020. https://doi.org/10.3390/healthcare8040437.