UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Acquisition of electrocardiogram signals and cardiac arrhythmia detection using neural networks

Dissertação de Mestrado

Igor Lopes Souza

São Cristóvão – Sergipe

2024

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Igor Lopes Souza

Acquisition of electrocardiogram signals and cardiac arrhythmia detection using neural networks

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Engenharia da Computação.

Orientador(a): Prof. Daniel Oliveira Dantas

São Cristóvão – Sergipe
2024

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL
UNIVERSIDADE FEDERAL DE SERGIPE

Souza, Igor Lopes
S729a *Acquisition of electrocardiogram signals and cardiac arrhythmia detection using neural networks* / Igor Lopes Souza ; orientador Daniel Oliveira Dantas. - São Cristóvão, 2023.
50 f.; il.

Dissertação (mestrado em Ciência da Computação) – Universidade Federal de Sergipe, 2023.

1. Eletrocardiografia. 2. Coração – Doenças - Diagnóstico. 3. Redes neurais. I. Dantas, Daniel Oliveira orient. II. Título.

CDU 004.8

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do Curso de Mestrado em Ciência da Computação-UFS.
Candidato: IGOR LOPES SOUZA

Em 19 dias do mês de dezembro do ano de dois mil e vinte três, com início às 09h, realizou-se na Sala de Seminários do PROCC da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **IGOR LOPES SOUZA**, que desenvolveu o trabalho intitulado: "***Acquisition of electrocardiogram signals and cardiac arrhythmia detection using neural networks***", sob a orientação do Prof. Dr. Daniel de Oliveira Dantas. A Sessão foi presidida pelo Prof. Dr. Daniel Oliveira Dantas (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. Elyson Adan Nunes Carvalho (UFS) e, em seguida, o Prof. Dr. Leonardo Nogueira Matos (Procc/UFS). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO. Atendidas as exigências da Instrução Normativa 05/2019/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução no 04/2021/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária "*Prof. José Aloísio de Campos*", 19 de dezembro de 2023.

Prof. Dr. Daniel Oliveira Dantas
(PROCC/UFS)
Presidente

Prof. Dr. Leonardo Nogueira Matos
(PROCC/UFS)
Examinador Interno

Prof. Dr. Elyson Adan Nunes Carvalho
(UFS)
Examinador Externo

gov.br
Documento assinado digitalmente
IGOR LOPES SOUZA
Data: 08/01/2024 10:47:52-0300
Verifique em https://validar.iti.gov.br

Igor Lopes de Souza
Candidato

Abstract

Electrocardiography is a frequently used examination technique for heart disease diagnosis. Represented by the test called electrocardiogram (ECG), electrocardiography is essential in the clinical evaluation of athletes, risk patients who need surgery, and also those who have heart disease. Through electrocardiography, doctors can identify whether the cardiac muscle dysfunctions presented by the patient are of inflammatory or degenerative origin and early diagnose serious diseases that primarily affect the blood vessels and the brain. Thus, the objective of this project is to develop a prototype capable of capturing, analyzing, and classifying a patient's electrocardiogram signals for the detection and prevention of cardiac arrhythmia in clinical patients. Our ECG signal classification model obtained an accuracy of 98.12% and an F1-score of 99.72% in the classification of ventricular ectopic beats (V). Our ECG acquisition board circuit tested gain output is 28.8V/V and the frequency cut is 40Hz.

Keywords: electrocardiography, ECG, acquisition, classification.

Contents

1
Introduction

Cardiovascular diseases (CVD) are listed as the leading cause of death by the World Health Organization (MCALOON et al., 2016). Arrhythmia or heart rhythm disorder is considered one of the most common disorders of the heart. Arrhythmia is a problem with the rate or rhythm of the heartbeat. During an arrhythmia, the heart may beat too fast, slow, or irregularly. Atrial fibrillation (AF) is the most prevalent case of arrhythmia. AF causes irregular heartbeats. In AF, the electrical activity of the atria (upper chambers of the heart) is irregular, inconsistent, and not synchronized with the ventricles (HAGIWARA et al., 2018).

AF is diagnosed by interpreting the ECG. Automatic diagnosis is useful in home settings, where an ECG interpretation specialist is not available to diagnose AF (MANT et al., 2007). Classification of ECG signals is necessary for the automatic diagnosis of arrhythmia. To improve AF detection, machine learning methods were used by Lown (LOWN et al., 2020; POLLOCK et al., 2020; SHOEMAKER et al., 2020). Convolutional neural networks (CNN) have the capability of hierarchical feature learning, which allows the neural network to distinguish and generalize ECG signal patterns with greater accuracy than a human expert (CHEN et al., 2022; KIRANYAZ et al., 2021). CNNs have been used to diagnose arrhythmias, coronary artery disease, and classify strokes (ZHIQIANG; JUN, 2017).

Telemedicine is the remote delivery of healthcare services, including examinations and consultations, through telecommunications infrastructure (ROCKWELL; GILROY, 2020). Telemedicine allows healthcare providers to assess, diagnose and treat patients without the need for a face-to-face visit. The implementation of a device for home and academic use to acquire ECG signals combined with an automatic classifier is useful to increase the popularization of heart disease prevention devices. The low cost of the device and the ease of its implementation outside the hospital environment, through telemedicine by connecting the device to the Internet network, make the new circuits feasible for remote environments, far from the help of a health professional (AMINE; REDOUANE; NARIMA, 2022).

1.1 Objective

This dissertation focus on the classification of arrhythmia signals in public ECG datasets and the implementation of an ECG signal acquisition device. The improved classification of ECG signals can generate accurate responses in the detection of cardiac arrhythmias, facilitating the care of patients by health professionals. The creation of a cardiac arrhythmia detection device with home and academic use may help expand access to ECGs and reduce the number of deaths from heart disease (CHEN et al., 2022).

1.2 Specific objectives

This section lists the specific objectives needed to implement an ECG signal acquisition device and the classification of cardiac arrhythmia signals. The objectives required to implement ECG acquisition device are:

- Design and simulation of the ECG signal acquisition circuit.
- Implementation in the protoboard and testing of signal-to-noise ratio and gain of the ECG signal in the protoboard.
- Validation of frequencies cut and gain of the ECG signal on the protoboard.

The objectives required for the classification of cardiac arrhythmia signals are:

- Develop the code for preprocessing the ECG signal.
- Develop the architecture of the CNN classifier.
- Develop the classifier training method.
- Evaluate the trained classifier by comparing accuracy, sensitivity, specificity, precision, and F1-score tests with those of other works.

1.3 Work Organization

This document is organized as follows into seven chapters that will serve as a basis for understanding the problem, the rationale for the work, the platform, the experimental study used to achieve the results, and the conclusion. The following topics describe the content of each chapter:

- Chapter 1 presents the introduction;

- Chapter 2 presents the theoretical background on CNN, performance metrics, and machine learning.
- Chapter 3 presents works related to the project;
- Chapter 4 presents the ECG signal acquisition board. The acquisition circuit is formed by the ECG signal conditioning and amplification steps;
- Chapter 6 presents the methodology to be used in the classification of ECG signals;
- Chapter 5 presents the classifier results obtained and discusses them;
- Chapter 7 presents the device results obtained and discusses them;
- Chapter 8 exposes the final considerations and the contributions

2

Machine Learning

Machine learning (ML) is a branch of artificial intelligence where computers can learn behaviors, identify patterns, and make decisions with minimal interaction from humans (Mitchell, 1997). From a dataset, learning algorithms go through a training process, generating a model that can handle new situations and solve problems (Tan; Steinbach; Kumar, 2006). According to Marsland (MARSLAND, 2015), there are four categories of learning:

- **Supervised Learning**: It is done using a set of training examples containing the correct answers. Based on this training set, the algorithm generalizes the answers to correctly answer all possible inputs.
- **Unsupervised Learning**: Correct answers are not provided. The algorithm identifies similarities between entries so that entries that have something in common are categorized together.
- **Reinforcement learning**: It is the training of machine learning models to make decisions. The agent learns to achieve a correct response in an uncertain and potentially complex environment. In reinforcement learning, the artificial intelligence system faces a situation, in which the computer uses the strategy of trial and error to find a solution to the problem. To get the machine to do what the programmer wants, the classification model receives rewards or penalties for the actions it performs, with the goal being to maximize the total rewards.
- **Evolutionary Learning**: The biological evolution algorithm can be seen as a learning process. Organisms adapt to improve their survival rates and their chances of having offspring in their environment.

In classification problems, the classifier needs to be able to generalize the data contained in the training set and classify each of the inputs, with the purpose of later labeling objects not found in the training set. In regression problems, the algorithm seeks to estimate continuous values based on a series of other historical data (Guimarães; Meireles; Almeida, 2019). This learning process requires several steps: getting the data; preparing, cleaning, and manipulating the data; choosing and training the model; testing the data; evaluating the model, and improving the model (Tan; Steinbach; Kumar, 2006). Data is one of the most important factors for ML and its quality is a major concern. The quality of the knowledge extracted is largely determined by the quality of the data provided in the input (Guimarães; Meireles; Almeida, 2019). It is very common for the database to have some category of noise, and it ends up being necessary to do data cleaning to detect and remove anomalies, make decisions for missing data, or even remove irrelevant attributes.

Data preprocessing prepares the data for training by removing input noise and improving the quality of model training. Data preprocessing is a process that depends on the ability of the person conducting it to identify the problems present in the data and to use appropriate methods to solve each of the problems (Neves, 2003). In addition to problems in data preprocessing, there are also problems in the training and testing steps. One of these problems is overfitting. This occurs when the model is trained too much, generating low error rates only on the training set and causing any other unknown input to have a greater error rate (Russell; Norvig, 2010). Just as there is overfitting, there is also underfitting which is when the model cannot find a relationship between variables, causing the model to be of no use at all (Russell; Norvig, 2010).

Generalizability refers to the ability of a model to adapt and react appropriately to novel data chosen from the same distribution as the initial model input. Overtraining the data will prevent a model from generalizing its data. In such cases, when new data is provided, it will make inaccurate predictions. Even if the model can make accurate predictions based on the training data set, it will become inefficient, or overfitted.

To avoid overfitting, underfitting and improve the generalization of the classifier, regularization techniques have been created. One of the simplest regularization techniques involves adding a regularization parameter b to the Loss error function.

$$\text{Loss} = f(\text{preds}, y) \tag{2.1}$$

$$\text{preds} = WX + b \tag{2.2}$$

where y is the vector of the desired output, preds is the model's predicted value, W is the parameters, and X is the input vector. The softmax function is a function that transforms a vector of Q real values into a vector Q of non-negative real values smaller than 1. The values of the function's input can be positive, negative, null, or greater than 1, but the softmax function transforms the values to be between 0 and 1, to be interpreted as probabilities (equation 2.3). The advantage of softmax is to mimic one hot coding.

$$\sigma_i(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.3}$$

where $\vec{z}$ is the input vector for the softmax function, consisting of (z_0, ..., z_K). The values z_i are the elements of the input vector for the function softmax. The e^{z_i} is the standard exponential function, applied to each input vector element. The function generates a positive value greater than zero, which will be very small if the input is negative, and very large if the input is positive. The $\sum_{j=1}^{K} e^{z_j}$ is the normalization term, which ensures that all output values of the function will sum up to 1 and each will be in the range 0 to 1, thus constituting a valid probability distribution. The K is the number of classes in the multiclass classifier.

One of the most widely used forms of regularization is the L_1 regularization, LASSO (Least Absolute Shrinkable and Selection Operator) regularization. LASSO regularization adds a factor to the sum of the absolute values of the model coefficients. LASSO regularization attempts to minimize the following function:

$$\mathrm{J}(\theta_1, \theta_2, \theta_3, \ldots, \theta_\mathrm{n}) = \mathrm{Loss} + \lambda \sum_{j=1} |\theta_j| \tag{2.4}$$

The values of θ are the weights that are being adjusted, λ is the regularization rate and is responsible for controlling the amount of regularization applied to the model. The λ is selected using cross-validation (BERRAR, 2019).

Cross-validation is a technique used to evaluate and test a machine learning model's performance. The technique involves reserving a specific sample of a dataset with which the model is not trained and then testing the model on this sample to evaluate it. Cross-validation methods can be classified into exhaustive and non-exhaustive methods. Exhaustive methods test all possible ways of partitioning the original data sample into a training set and a test set, while non-exhaustive methods test only part of the ways of partitioning the data.

The Loss function for the LASSO and ridge regularizations can be obtained with the following methods: mean square error function (Equation 2.5), residual sum of squares (Equation 2.6), and ordinary least squares method (Equation 2.7).

$$\mathrm{Loss} = \sum_{i=1}^{n} \frac{(y^i - \hat{y}^i)^2}{n} \tag{2.5}$$

$$\mathrm{Loss} = \sum_{i=1}^{n} (y^i - f(x_i))^2 \tag{2.6}$$

The values $\hat{y}^i$ and $f(x_i)$ represent the values of the i-th variable to be predicted, y^i represents the true value of the i-th observation and n is the total number of observations. In the ordinary least squares method, values of b_1 and b_0 are selected so that the sum of squares of the difference between the calculated and observed values of y are minimized, ($\hat{e_i}$) represents the error for the i-th observation, as shown in equation 2.7.

$$
\begin{aligned}
\text{Loss} &= \sum_{i}^{n} (y^i - \hat{y}^i)^2 \\
&= \sum_{i}^{n} (y^i - b_1 x_1 + b_0)^2 \\
&= \sum_{i}^{n} (\hat{e_i})^2
\end{aligned}
\tag{2.7}
$$

LASSO regularization (equation 2.4) can be used for feature selection because the coefficients of the least important features are reduced to zero. If λ is too large, the regularization term will be high, the cost function will be too high. Then the downward gradient will try to make all the values of λ reach zero to reduce the cost.

Different values of λ can reduce different θ parameters to zero, for example, λ_2 can penalize θ_3, θ_4 and θ_5, while λ_3 can penalize only θ_3 and θ_4. The λ_4, on the other hand, may not penalize any θ.

The L_2 regularization, ridge regularization, is also a widely used form of regularization. Ridge regularization adds a factor of the sum of the square values of the model coefficients. Ridge regularization tries to minimize the 2.8 function.

$$
\mathrm{J}(\theta_1, \theta_2, \theta_3, \ldots, \theta_\mathrm{n}) = \text{Loss} + \lambda \sum_{j=1} \left|\theta_j^2\right| \tag{2.8}
$$

The difference between smoothing L_1 and L_2 is that the gradients of the functions with respect to the parameters in smoothing L_1 are independent of the input parameters, so the parameters of L_1 can be set completely to zero and can be ignored. In smoothing L_2 the gradients of the function are linearly dependent on the parameters, so the parameters cannot be zero. The regularization dropout will be explained in section 2.1.

Once the model is built, it is very useful to measure its performance, as such a measurement can provide an unbiased evaluation of your model. Among the measures for comparing model performance are accuracy, precision, sensitivity, specificity, and F1-score (Tan; Steinbach; Kumar, 2006).

The confusion matrix (Table 1) is a table that shows the performance of a classification algorithm. The binary classification problem will show four values, which are:

- **True positive**: Occurs when the model correctly predicts the positive class.
- **False positive**: Occurs when the model incorrectly predicts the positive class.
- **True negative**: Occurs when the model correctly predicts the negative class.

- **False negative**: Occurs when the model incorrectly predicts the negative class.

Table 1 – Confusion Matrix - source: (MARSLAND, 2015)

True positive	False positive
False negative	True negative

- **Accuracy (ACC)**: This is a widely used metric in the literature and its formula calculates the number of hits divided by the total sample size, represented by the number of hits + number of errors, as shown in the equation 2.9.

$$\mathrm{ACC} = \frac{(\mathrm{TP}+\mathrm{TN})}{(\mathrm{TP}+\mathrm{FN}+\mathrm{FP}+\mathrm{FN})} \tag{2.9}$$

- **Precision (PRE)**: Precision evaluates the correctly classified samples among those classified as positive. It is used to work only with the positive values as shown in the equation 2.10.

$$\mathrm{PRE} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FP})} \tag{2.10}$$

- **Sensitivity (SEN)** and **specificity (SPE)**: Sensitivity is used to evaluate model performance, as it allows us to see how many positive instances the model was able to correctly identify, i.e., sensitivity measures a model's ability to correctly identify positive instances (equation 2.11). Specificity measures the proportion of true negatives that are correctly identified by the model. High specificity means that the model correctly identifies most negative outcomes, while low specificity means that the model incorrectly labels negative outcomes as positive (equation 2.12).

$$\mathrm{SEN} = \frac{\mathrm{TP}}{(\mathrm{TP}+\mathrm{FN})} \tag{2.11}$$

$$\mathrm{SPE} = \frac{\mathrm{TN}}{(\mathrm{TN}+\mathrm{FP})} \tag{2.12}$$

- **F1-score (F1S)**: This is the harmonic mean of the precision with the sensitivity, resulting in a single number that indicates the overall quality of the model (equation 2.13).

$$\mathrm{F1S} = \left(2\frac{\mathrm{PRE}\times\mathrm{SEN}}{\mathrm{PRE}+\mathrm{SEN}}\right) \tag{2.13}$$

Any pair of these measures provide more information than just accuracy. By using various evaluation metrics, it is possible to measure the performance of the classifier and see the advantages and disadvantages of the model for improvement.

After going through all the steps of the machine learning flow, the model can be used to store the acquired knowledge and be used in another unrelated but similar task. This learning is called transfer learning. Transfer learning occurs when one domain uses as much information from another domain as possible, without affecting predictions due to the difference between the training and application sets (Shao; Zhu; Li, 2015).

Deep neural networks (DNNs) are another machine learning technique built using artificial neural networks. This model resembles how neurons work, consisting of input, output, and hidden layers. The main purpose of a deep neural network is to receive a set of inputs, perform progressively complex computations on them, and provide output to solve problems, such as classification.

2.1 Convolutional neural networks

Convolutional neural networks are an effective deep learning algorithm for solving classification problems. Its effectiveness comes through the use of the multi-layer convolution procedure (Simard; Steinkraus; Platt, 2003). The convolution of x and h, denoted by $x*h$, returns a third function, $z(t)$, with the sum of the multiplication of the function kernel by its offset k (Yamashita et al., 2018) (equation 2.14).

$$z(t) = x * h = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{2.14}$$

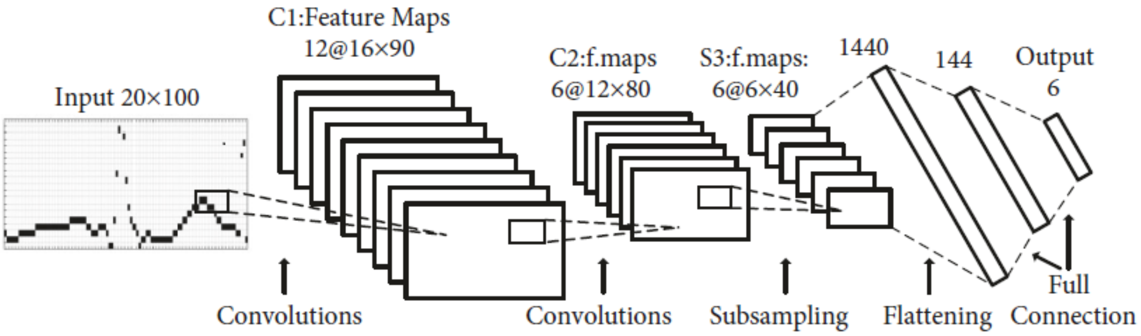


Figure 1 – Typical CNN model

CNNs have several layers and the only requirement is that the CNN must have at least one convolutional layer, as shown in Figure 1. Below are the most common layers.

- **Convolutional layer**: This layer aims to extract the labeled data and obtain common characteristics (features) of its input. In the convolutional layer, a convolution is performed between the input and a filter that is represented by a matrix with an equal number of rows and columns. By sliding the filter over the input, the sum of the product of the filter values by the input values is obtained to generate the output. The output is the feature map that gives us information about the input. The feature map is then sent to other layers to get new features of the input (Schmidhuber, 2015).

- **Subsampling layer**: The function of this layer is to progressively reduce the spatial size of the input to reduce the number of parameters and computations needed in the network. One of the most widely used techniques for performing pooling is max-pooling. Let n be the size in pixels of a square image. The max-pooling layer takes a pool size as a parameter, usually a 2×2 matrix. It then divides the image into 2×2 areas and selects for each area the pixel with the highest value. The selected pixels compose a new image with the same order of magnitude as the original image. The max-pooling layer produces an image $n/2 \times n/2$ with half the side, a quarter of the size of the original image. Max-pooling is done in part to help reduce overfitting. The other forms of pooling are averaging and summation. Average pooling calculates the average of the elements in an image section of a predefined size. The total sum of the elements in the predefined section is calculated using sum pooling.

- **Fully connected layer**: This layer is at the end of the network, also known as dense layer. It extracts the features obtained by the previous layers and uses them to obtain an output with n neurons, where n is the number of classes needed for the model to finish classification (Schmidhuber, 2015).

- **Dropout layer**: The dropout layer removes, at each training step, individual nodes from the network with probability $1 - p$ or keeps them with probability p, so its output is a network with more relevant neurons; the incoming and outgoing edges for a dropped node are also removed. In the fully connected layer, all nodes in the output layer are directly connected to nodes of the previous layer. Most parameters and neurons develop co-dependence on each other during training, which restricts the individual power of each neuron, leading to overfitting of the training data. The dropout layer prevents overfitting by turning off these neurons (Krizhevsky; Sutskever; Hinton, 2017). The dropout layer receives as a parameter the percentage of neurons that are turned off.

- **Activation layer**: This layer uses an activation function, i.e., an artificial neuron that calculates the weighted sum of its inputs and adds a bias. The activation layer is usually present after the convolutional layers or the dense layers.

- **Flatten layer**: This layer prepares the data to be sent to other layers of the neural network. It transforms the resulting matrix from the previous convolutions and pooling layers into an array that will serve as the input layer for the dense layer.

- **Softmax layer**: Many multilayer neural networks end in a penultimate layer that produces real-valued scores that do not scale conveniently and can be difficult to work with (LIU et al., 2016). The softmax layer converts the scores to a normalized probability distribution. For this reason, it is customary to add a softmax function as the final layer of the neural network.

The use of smaller, shared weights, and thus being easier to train than the multilayer perceptron (MLP), was an improvement that made CNN (BOTALB et al., 2018) a powerful tool for solving problems in computer vision, facial, speech, and character recognition.

2.2 Long short-term memory

Long short-term memory (LSTM) is a type of recurrent neural network (RNN). In recurrent neural networks, the output of a layer can be used as input to previous layers, forming a cycle (HOCHREITER; SCHMIDHUBER, 1997). It is used for processing, predicting, and classifying time-series data.

LSTM is designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well-suited for tasks such as language translation, speech recognition, and time series forecasting. A traditional RNN has a single hidden state that is passed through time, which can make it difficult for the network to learn long-term dependencies. LSTMs address this problem by introducing a memory cell, which is a container that can hold information for an extended period of time. The memory cell is controlled by three gates: the input gate, the forget gate, and the output gate. These gates control the information to be added or removed from the memory cell.

- **Input gate**: The input gate optionally classifies information that is relevant to the current cell state. It is the gate that determines which information is necessary for the current input by using the sigmoid activation function, σ. It then stores the information in the current cell state and the hyperbolic tangent function computes the vector representations of the input-gate values, which are added to the cell state.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_f) \tag{2.15}$$

$$\tilde{C}_t = \tanh(W_i[h_{t-1}, x_t] + b_f) \tag{2.16}$$

In equation 2.16 the i_t represents the input gate, W_i represents the weight of current gate neurons, h_{t-1} represents the previous LSTM state block, $\tilde{C}_t$ represents candidates for cell state in the timestamp t, and b_f represents the biases of the current gate (GRAVES; GRAVES, 2012).

- **Forget gate:** After getting the output of the previous state h_{t-1}, the forget gate, f_t, helps in the decisions about what must be removed from the h_{t-1} state and thus keeping only relevant stuff. After the forget gate receives the input x_t and output from h_{t-1}, it performs a pointwise multiplication with its weight matrix with an add-on of sigmoid activation, which generates probability scores. These probability scores help it determine what is useful information and what is irrelevant. The input gate remembers which tokens are relevant and adds them to the current cell state with hyperbolic tangent activation enabled. Also, the forget gate output, when multiplied with the previous cell state, memory, C_{t-1}, the forget gate discards the irrelevant information. Hence by combining these two gates the cell state is updated without any loss of relevant information or the addition of irrelevant ones.

$$f_t = \sigma(W_i[h_{t-1}, x_t] + b_f) \tag{2.17}$$

$$C_t = f_t(C_{t-1}) + i_t\tilde{C_t} \tag{2.18}$$

- **Output gate:** The output gate o_t decides what to output from our current cell state. The output gate also has a matrix where weights are stored and updated by backpropagation. This weight matrix takes in the input token x_t and the output from the previously hidden state h_{t-1} and does the same pointwise multiplication task. However, as said earlier, this takes place on top of a sigmoid activation as we need probability scores to determine what will be the output sequence.

$$o_t = \sigma(W_i[h_{t-1}, x_t] + b_f) \tag{2.19}$$

$$h_t = o_t \tanh(c^t) \tag{2.20}$$

After we get the sigmoid scores, we simply multiply them with the updated cell state, which contains some relevant information required for the final output prediction. A final hyperbolic tangent multiplication is applied at the very last, to ensure the values are in the range $[-1, 1]$, and our output sequence is ready.

3

Related works

The papers listed below were selected by the search string with the key concepts of this project. The search was performed in ACM Digital Library, Scopus, ScienceDirect (Elsevier), Web of Science, and Google Scholar. The selection was carried out based on the paper's abstract. If the paper's abstract did not mention anything referring to ECG signal analysis, the study was removed from the list. The search string used was

("convolutional neural network" OR "neural networks" OR "1D-CNN")
AND
("electrocardiogram" OR "ECG" OR "cardiac arrhythmia")

Shih (SHIH et al., 2009) proposed an embedded mobile ECG classification system that integrates ECG signal classification and radio-frequency identification (RFID) together to monitor elderly patients. The method has an accuracy of 92% in detecting cardiac arrhythmias and allows continuous monitoring and identification of elderly patients when alone.

Kiranyaz (KIRANYAZ; INCE; GABBOUJ, 2015) proposed a system for electrocardiogram classification and monitoring by adaptively implementing 1D CNNs. These networks are used to merge the two main sections of the classification of an ECG signal into a single learning corpus. Thus, for each patient, an individual 1D CNN was trained using a small common, patient-specific database, which improves the ability to extract patient-specific features, patient-specific heartbeats, and cardiac arrhythmia classification performance.

Kiranyaz (KIRANYAZ; INCE; GABBOUJ, 2017) proposed a personalized health monitoring system that can detect early occurrences of arrhythmias from a patient's ECG signal by modeling the common causes of arrhythmias in the signal domain as degradation from normal ECG beats to abnormal beats. Using the degradation models, abnormal beats were created from the patient's average normal beat. A CNN was trained using real normal beats and synthesized abnormal beats. The main contribution of this study is to model the causes of cardiac arrhythmias modeled by a set of filters and used to synthesize potential abnormal beats from a healthy person, thus discarding the need to have a set of abnormal beats already recorded.

Acharya (ACHARYA et al., 2017a) proposed a CNN technique to automatically detect the different segments of the ECG signal. The algorithm consists of an eleven-layer deep CNN with the output layer having four neurons, each representing the classes of an ECG normal beats, atrial fibrillation, atrial flutter and ventricular fibrillation.

Acharya (ACHARYA et al., 2017b) developed a nine-layer deep convolutional neural network (DCNN) to identify five different classes of heartbeats in ECG signals: non-ectopic beat, supraventricular ectopic beat, ventricular ectopic beat, fusion beat, and unknown beat. The experiment was conducted on noise-attenuated and non-attenuated data sets from a public database, MIT-BIH. The DCNN was artificially augmented to equalize the number of instances of the five heartbeat classes and filtered to remove high-frequency noise.

Li (LI et al., 2020) developed a deep learning method for cardiac arrhythmia classification based on ResNet. The design consists of a 1D 31-layer convolutional residual network. The algorithm includes four residual blocks, each of which consists of three layers of 1D convolutions, three layers of batch normalization (BN), three layers of the rectified linear unit (ReLU) activation function, and a structure of identity shortcut connections. The 2-lead ECG signals were used in combination with deep learning techniques to automatically identify the normal, left group, right group, premature atrial, and premature ventricular contraction heartbeats.

Han (HAN; SHI, 2020) presented a method to detect and localize myocardial infarction by combining a multiple-lead residual neural network (ML-ResNet) framework with three residual blocks and feature fusion using 12-lead ECG recordings. The single-lead feature branching network is trained to automatically learn local features of different levels between different layers, which can be used to characterize the spatial representation of the ECG. The main features are merged as global features. For the generalization and evaluation of the proposed method in clinics, intra-patient and inter-patient schemes were used.

Krizhevsky (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) trained a DCNN to classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 competition into 1000 different classes. Error rates of 37.5 percent and 17.0 percent were achieved, which was considerably better than the state of the art at the time of the work. The neural network, with 60 million parameters and 650,000 neurons, consists of five convolutional layers. Unsaturated neurons and a GPU implementation were used for training.

Huang. (HUANG et al., 2020) developed a deep neural network to predict the mortality of patients at Chang Gung Memorial Hospital using a 12-lead ECG. The clinical database MUSE of the Chang Gung Memorial Hospital was used to analyze 12-lead ECG voltage-time traces of 10s at rest, which were linked to the national death registry for all-cause and cause-specific mortality.

Ten thousand patients, out of the twenty thousand tested, who died within one year of

ECG examinations were randomly selected to train a 1D deep neural network to predict mortality risk. The data was divided into 16,000 patients for model training and 4,000 patients for the validation set. The model achieved an area under the receiver operating characteristic curve (ROC) of 0.85 on the tested data set.

Martis (MARTIS et al., 2014) investigated four different methods for atrial fibrillation and atrial flutter feature extraction: the principal components of discrete wavelet transform coefficients, independent components of discrete wavelet transform coefficients, principal components of discrete cosine transform coefficients, and independent components of discrete wavelet transform coefficients methods. Martis explored three different classification techniques: K-nearest neighbor, decision tree, and artificial neural network. The methodology used data from MIT-BIH arrhythmia and atrial fibrillation databases. Discrete cosine transform coupled with independent component analysis and K-nearest neighbor yielded the highest average sensitivity of 99.61%, average specificity of 99.99%, and classification accuracy of 99.45% using tenfold cross-validation.

Sellami (SELLAMI; HWANG, 2019) proposed a new type of deep convolutional neural network for heartbeat classification. A batch-weighted loss function was created to quantify the loss and to decrease the imbalance between classes. The loss weights change dynamically as the distribution of classes in each batch changes. Twenty-two heartbeat records from among the 48 available in the MIT-BIH database were used to increase the accuracy of the heartbeat classification. Although only one derivation of the ECG signal is used without data preprocessing, the method has higher accuracy and precision when compared to existing methods, at the time, of classifying heartbeats into 5 classes: non-ectopic beat, supraventricular ectopic beat, ventricular ectopic beat, fusion beat, and unknown beat. The accuracy, precision, sensitivity, and specificity obtained were 99.48%, 98.83%, 96.97%, and 99.87% for the intra-patient and 88.34%, the 48.25%, 90.90%, and 88.51% between patients.

Zhai (ZHAI; TIN, 2018) developed a high-performance ECG-based arrhythmic beat classification system. The classifier was designed based on a CNN. The single-channel ECG signal was segmented into heartbeats according to the change in beats. The beats were transformed into a double-beat coupling matrix with a CNN classifier with 2D inputs, which captured the beat morphology and beat-to-beat correlation in the ECG. A systematic training beat selection procedure was implemented, including the most representative beats in the training set to improve the classification. The CNN system improved the sensitivity and accuracy for ectopic beats by more than 12.2% and 11.9%, respectively.

Zhu. (ZHU et al., 2020) developed a deep learning approach for the automated diagnosis of multiple cardiac rhythm labels or conduction abnormalities by real-time ECG signal analysis. The dataset used was obtained from ECG data, in the 10s pattern and 12-channel format, from adult patients, with 21 distinct rhythm classes for the diagnosis of multilabel level cardiac arrhythmias.

ECG signal data were collected at three campi of Tongji Hospital (Huazhong University of Science and Technology, Wuhan, China) and annotated by cardiologists. The deep learning model used to diagnose abnormalities in the heart obtained an overall average F1-score of 0.887, a sensitivity of 0.867, and a specificity of 0.995.

Schwab (SCHWAB et al., 2017) proposed a machine learning approach based on recurrent neural networks (RNN) to analyze different cardiac arrhythmias with only a single lead and short ECG recordings, below 10s. To facilitate training dependencies on the temporal dimension, a new task formulation was introduced that takes advantage of the natural beat-based segmentation of ECG signals.

Heartbeat features that have proven useful for classification in previous work were used (SARKAR; RITSCHER; MEHRA, 2008; TATENO; GLASS, 2001; GARCÍA et al., 2016; RÓDENAS et al., 2015; ALCARAZ et al., 2006), in addition to stacked denoising autoencoder (SDAE) to capture differences in morphological structure. The RNNs were extended with a soft attention mechanism that allows them to select which segment of the ECG the RNNs prioritize for their decision-making. The model achieves an F1-score of 0.79 on the test set.

Rahhal (RAHHAL et al., 2019) proposed a novel end-to-end architecture based on a dense convolutional network (DCN) for ECG signal classification. The architecture is based on two main modules: the first is a generative module and the second is a discriminative module. The generative module converts the 1D ECG signal into an image through fully connected up-sampling layers and convolutional layers. The discriminative module receives the image from the generative module and performs feature learning and classification.

To deal with the data imbalance problem that characterizes ECG data, focal loss (FL) was used, which is based on remodeling the standard cross entropy loss function that reduces the error assigned to correctly classified ECG beats. The experiments were validated using the MIT-BIH arrhythmia database in four different scenarios, using four classes in the first scenario, five in the second, and 12 in the third. The proposed method achieves a significant improvement in accuracy, obtaining an accuracy of 97%.

4

ECG signal acquisition board

This chapter will describe the steps necessary for prototyping the ECG signal acquisition circuitry and attaining the specific objectives listed in Section 1.2. These steps include designing circuits for amplifying and conditioning ECG signals and validating the printed circuit boards for these circuits. The code is processed by the Raspberry Pi (RASPBERRY PI (TRADING) LTD., 2019) board, which is connected to a circuit for ECG signal acquisition. The circuit performs the amplification and processing of the signal. For this, an AD620 amplifier (GOIS et al., 2017), a band-stop, and a low pass filter circuit are used to eliminate out-of-range signals from the ECG signal. The steps required to achieve this goal are presented below:

- Design and simulation of the ECG signal acquisition circuit.
- Implementation on a protoboard and performing tests of signal-to-noise ratio and gain of ECG signal on protoboard.
- Validation of the frequencies cut and gain of the ECG signal on the protoboard.

4.1 Acquisition and conditioning step

Figure 2 shows a block diagram with an overview of the project from the acquisition of the bioelectric signal to processing in the Raspberry Pi board. The implementation of the board and the acquisition and conditioning step bode plots are publicly available[1].

[1] https://github.com/Igor-Lopes-Souza/PCBlayout

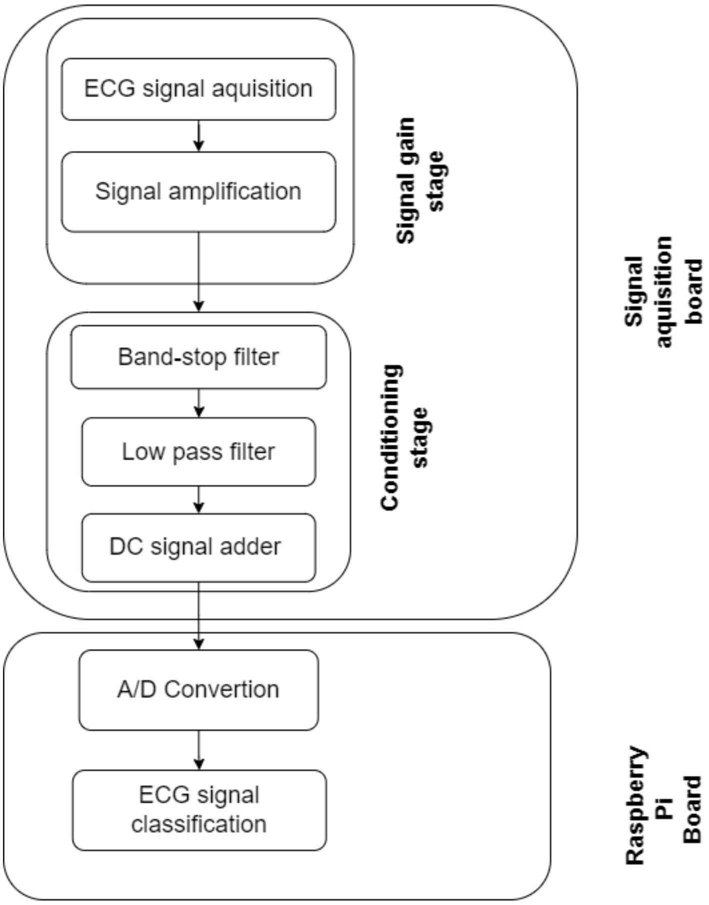


Figure 2 – Circuit step block diagram

4.1.1 Amplification step

The ECG signal has low voltage amplitude, in the order of millivolts, for our circuit to read the cardiac signal an amplification step is required. The bioelectric signal amplification step is composed of AD620 (ANALOG DEVICES, 2007). This is due to its high common mode rejection capability, and high input impedance, among other advantages. The electrodes that provide the differential output connected to the conditioning circuit input are connected to pins **3** and **2**, while a third reference electrode is connected in between R_{22} and R_{23}, as shown in Figure 3.

The amplification step creates a DC offset gain that adds noise to the circuit. In order to remove the offset, an AC coupling step was used (SPINELLI; MAYOSKY, 2000), as shown in Figure 3. For the AC coupling step, the ideal is to implement a filter with a frequency as close to 0Hz as possible. For this, it is necessary to use resistors and capacitors with high values. As electrolytic capacitors do not have a good frequency response, a ceramic capacitor was used whose highest value easily found was 4.4μF. With the capacitor limited, resistors of values in the order of megohms would reach cut-off frequencies very close to 0Hz, however, the current flowing through the AC coupling circuit could be in the order of the polarization currents of the circuit. For this reason, it was decided to use a 100kΩ resistor (JÚNIOR et al., 2019). The

Table 2 – Resistors and capacitors values used in the ECG acquisition board

Components	Values
R_1, R_2, R_7, R_{10}	10kΩ
R_3, R_{11}	220Ω
R_4, R_{12}	470Ω
R_5, R_{13}	33kΩ
R_9, R_{15}	1kΩ
R_8, R_{16}	820Ω
R_6	1.8MΩ
R_{14}	1.5MΩ
$C_1, C_2, C_3, C_4, C_6, C_7$	1μF
C_5	4.4μF
R_{17}	1.1kΩ
R_{18}	18kΩ
R_{19}	2.4kΩ
R_{20}	22kΩ
R_{21}	100kΩ
R_{22}, R_{23}	2.2kΩ
R_{24}	1.2kΩ

resulting cutoff frequency was approximately 0.4Hz, which effectively removed the DC level from the signal. In Figure 4 the design of the amplifier circuit in conjunction with the conditioning circuit is shown using the EAGLE simulator. The values of the resistors and capacitors used in Figure 4 and 6 are shown in Table 2. Let R_G be the value of the gain resistor in Figure 3, where R_G is R_{24}, the value of the gain G_1 (ANALOG DEVICES, 2007) is given by

$$\mathrm{G}_1 = \left(\frac{49,4k\Omega}{\mathrm{R_G}}\right) + 1 \tag{4.1}$$

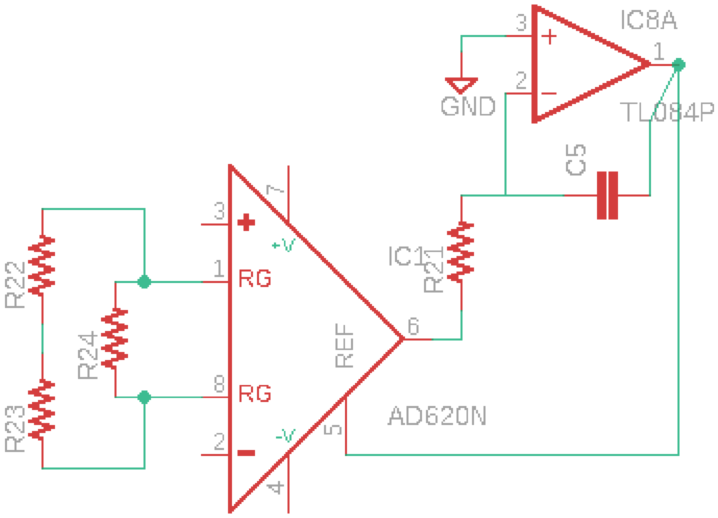


Figure 3 – Amplification step of the signal using AD620 (ANALOG DEVICES, 2007) connected to ad coupling

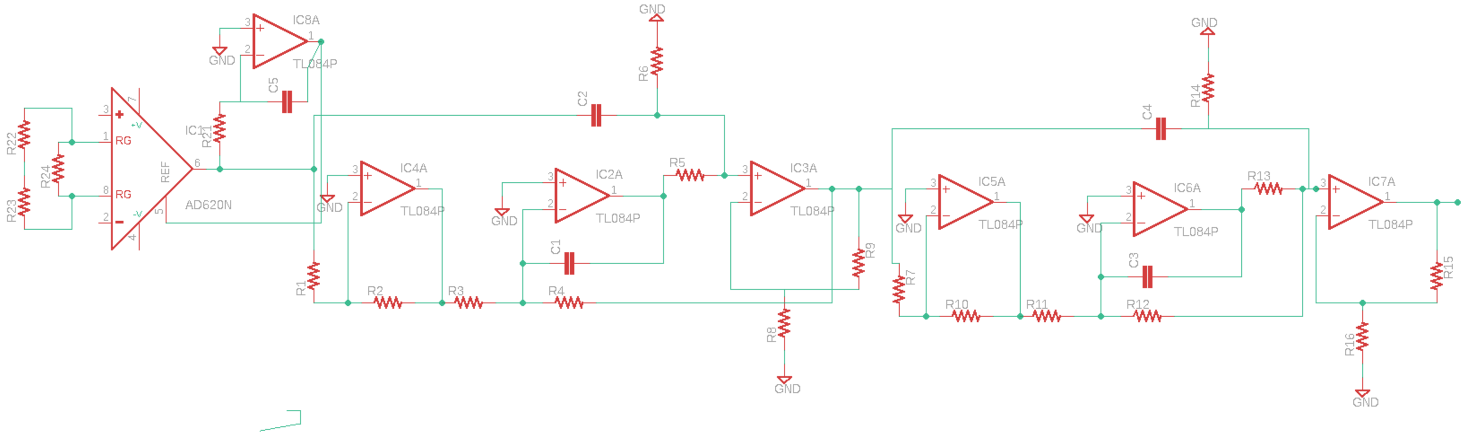


Figure 4 – Amplification step of the signal using AD620 (ANALOG DEVICES, 2007) connected to the conditioning step, simplified model (SPINELLI; MAYOSKY, 2000).

4.1.2 Conditioning step

ECG signals monitoring needs a stable signal passing frequency range between 0.37Hz and 40Hz. This step is composed of a band-stop filter, a low-pass filter, and a DC signal adder circuit. The band-stop filter removes the effects of the electromagnetic interference from the electric grid by attenuating the 60Hz frequency, present in electronic devices, and is composed of two Bainter Chebyshev fourth-order filters, as shown in Figure 5. The low-pass filter cuts all frequencies above 40Hz to remove unnecessary signals and potential noise from our circuit, as shown in Figure 6. The cutoff frequencies of the band-stop (f_{c1}) and low-pass f_{c2}) filters are given by the equations 4.2 and 4.3 [2]. The band-stop filter also applies a gain (G_2) to its input,

[2] https://github.com/Igor-Lopes-Souza/PCBlayout/filters

given by equation 4.4.

$$\mathrm{f_{c1}} = \left(\frac{1}{2\pi} \sqrt{\frac{R_2}{R_1 R_3 R_5 C_1 C_2}} \right) \tag{4.2}$$

$$\mathrm{f_{c2}} = \left(\frac{1}{2\pi R_{17} R_{18} C_6 C_7} \right) \tag{4.3}$$

$$\mathrm{G_2} = \left(\frac{s^2 + \frac{R_2}{R_1 R_3 R_5 C_1 C_2}}{s^2 + [\frac{R_5 + R_6}{R_5 R_6 C_2}] + [\frac{(1 + \frac{R_7}{R_8})}{R_4 R_5 C_1 C_2}]} \right)^2 \tag{4.4}$$

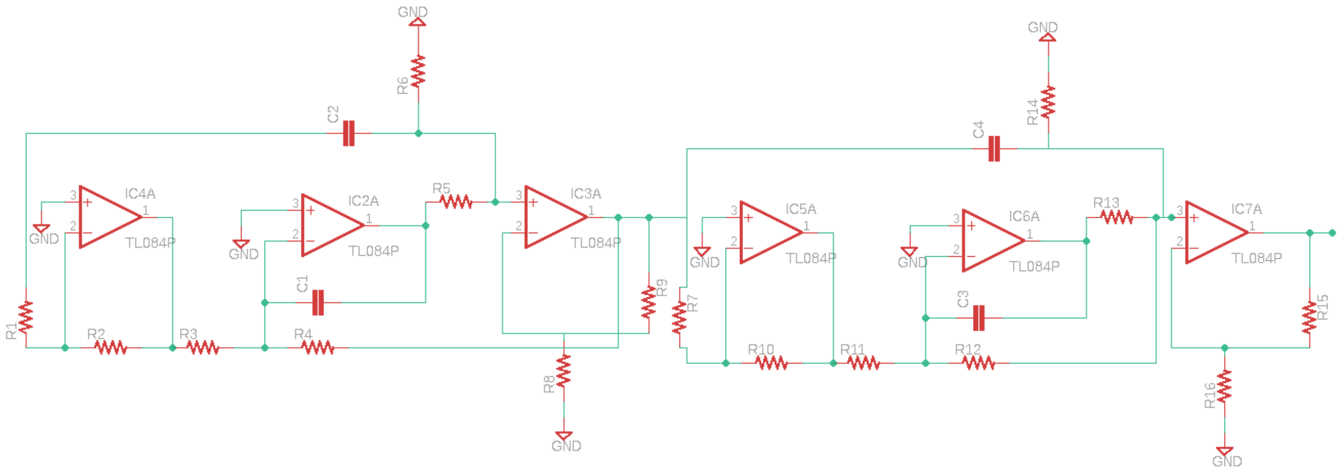


Figure 5 – Band-stop filter

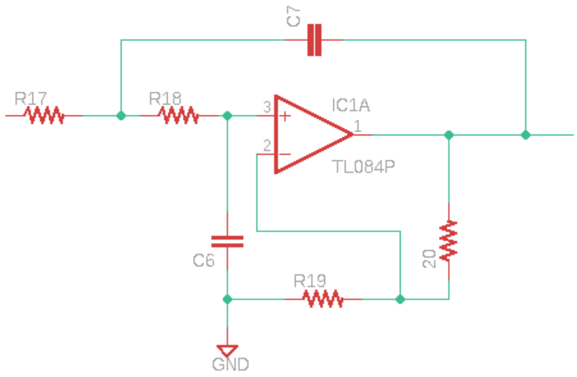


Figure 6 – Low pass filter

From the values shown in Table 2, the frequencies of the equations 4.2 and 4.3 from Figures 5 and 6 are approximately 60Hz and 39Hz, with the gains G_1 and G_2 of the equations 4.1 and 4.4 equal to 40V/V and 5V/V. With this, the gain is given by

$$G = (G_1 G_2) = 200 \tag{4.5}$$

The myoelectric signal can present negative voltages, but the Raspberry Pi board pins available for A/D conversion require positive voltage values. As a conditioning step, it is necessary to add a DC voltage level to the output of the conditioning step to detect the entire excursion of the signal. In this step, an amplifier in the inverter adder configuration and the voltage regulator LM317 (SEMICONDUCTOR, 2016) were used (SILVA et al., 2018).

The inverter adder was implemented with the CI TL084 which has four simple operational amplifiers (AMPLIFIERS, 1999), which receive the signal frequency conditioning output and the DC level as inputs, provided by the voltage regulator, as shown in Figure 7. The myoelectric signal is multiplied by a unitary gain G = $R_7/R_5 = 1$, we altered the R_7 value to 1kΩ, while the DC level is multiplied by a unit module gain. The voltage regulator LM317, as shown in Figure 8, receives a positive DC voltage as input and outputs another positive DC voltage level, as long as the difference between these voltages is in the range of 3V to 40V. The adjustable output voltage is defined by Equation 4.6. According to the LM317 datasheet, the values of $\mathrm{V_{ref}}$ and $\mathrm{I_{adj}}$ are 1.25 V and 50μA, respectively (SEMICONDUCTOR, 2016). Therefore, the DC level provided by the regulator will be approximately 1.6 V. This value was chosen because the Raspberry Pi analog pins detect values in the range from 0V to 3.3V.

The inverter adder inverts the signal after the DC level addition applies a 180-phase inversion. Therefore, the offset after this step is -1.6 V. In the next conditioning step, another phase inversion will be used, canceling the phase inversion.

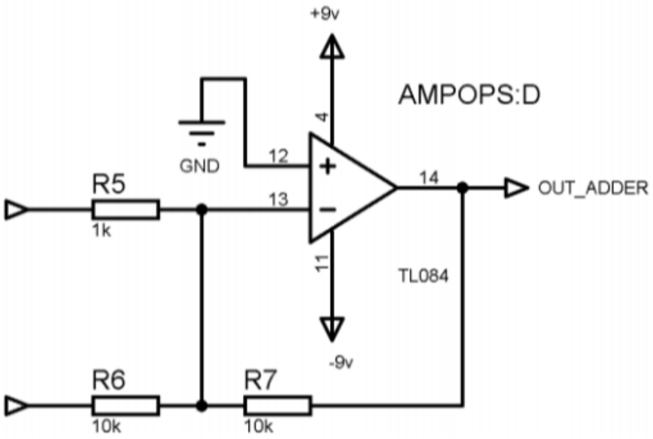


Figure 7 – Inverter adder amplifier

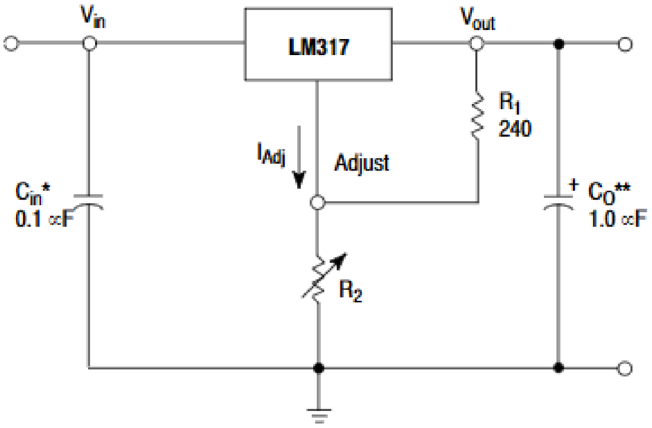


Figure 8 – Voltage Regulator LM317 (SEMICONDUCTOR, 2016)

$$\mathrm{V}_{\mathrm{out}} = \mathrm{V}_{\mathrm{ref}} \left(1 + \frac{R_2}{R_1}\right) \left(\mathrm{I}_{\mathrm{adj}}\, X R_2\right) \tag{4.6}$$

4.1.2.1 Rectification

The desired voltage restrictions may not always be respected even if the DC level is added to the signal. It is possible for inputs with greater amplitude than the myoelectric signal to be unintentionally supplied into the circuit during system test procedures. The inverting adder can produce a high-amplitude signal that could harm the Raspberry Pi board following the two gain steps. Applying voltage levels outside of this range to the analog pins of the Raspberry Pi A/D converter can compromise its functionality, even if the device ignores values outside of the 0 V to 3.3 V range.

A precision rectifier circuit was introduced in the last step of signal conditioning to ensure that it does not present voltage values less than zero as output, as seen in Figure 9. This circuit was built using an operational amplifier CI TL084 and two diodes 1N4007. After the precision rectifier, the Zener diode BZX79C3V3 which has a voltage threshold of 3.3 V, was used to avoid voltage levels above 3.3 V. Figure 9 shows the precision rectifier circuit diagram and the gain of its amplifier is given by Equation 4.7. As the resistors R_{10} and R_{11} have the same impedance, the gain of the amplifier is negative and unitary, thus canceling the phase inversion obtained in the previous step. The resistor R_{12} was inserted only to ensure the current flow necessary for operating the Zener diode D3 in reverse polarization.

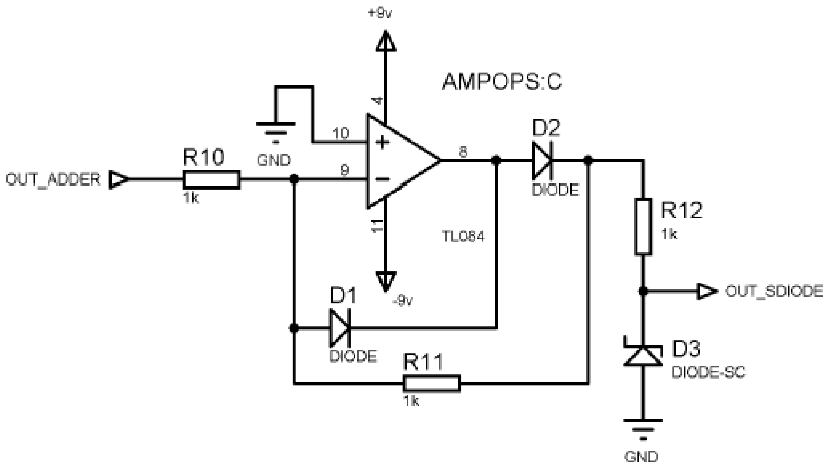


Figure 9 – Precision rectifier

$$V_0 = -\left(\frac{R_{11}}{R_{10}}\right) V_i \tag{4.7}$$

4.2 Processing board

Once the output of the ECG signal acquisition circuit is obtained, it will be sent to the MCP3008 (MICROCHIP TECHNOLOGY, 2008) analog-to-digital converter, which will convert it to classify the acquired signal. The Raspberry Pi makes the signal classification process easier, due to its processing power and low cost (SHAH et al., 2016).

The Raspberry Pi connection to the ECG signal acquisition board has not been tested. To do this, it would be necessary to write the firmware for the A/D conversion and communication with the USB port of the computer. That will be a future job.

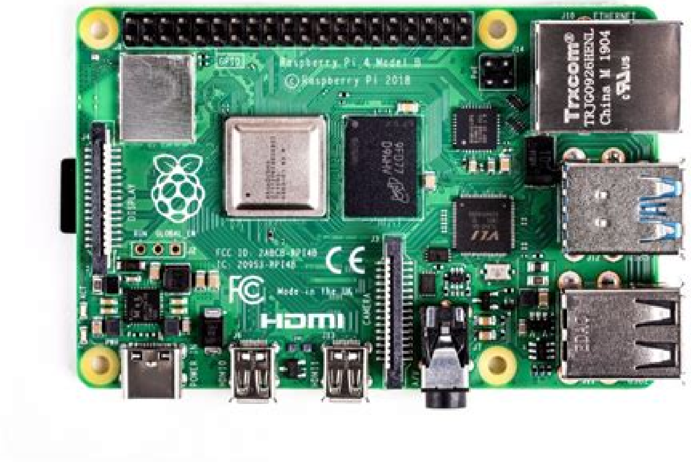


Figure 10 – Raspberry Pi (RASPBERRY PI (TRADING) LTD., 2019)

5

Device validation

The validation of this circuit included 3 steps: amplification test, low-pass filter frequency response test, and band-stop 60Hz filter test. The purpose of the first test was to confirm that the amplifier was offering a 40 V/V gain. The plot of the instrumentation amplifier output from the function generator signal, as recorded by the oscilloscope, is displayed in Figure 11. The result was G = (2,88)/(0,10) = 28,8 V/V.

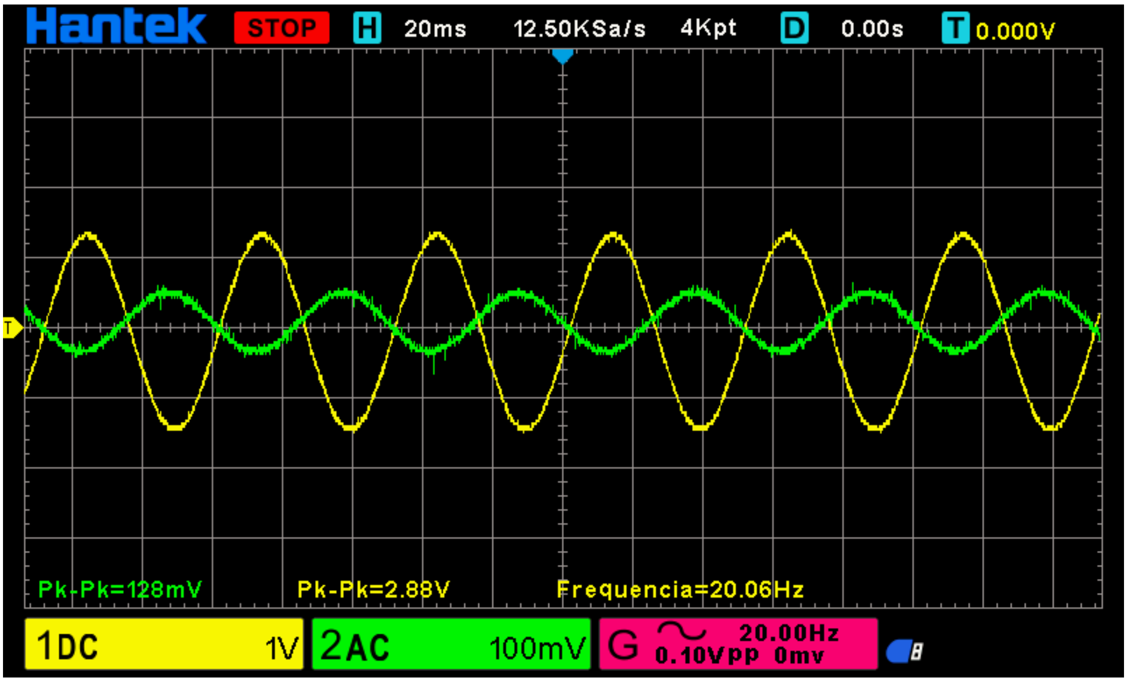


Figure 11 – Amplification step result using AD620 (ANALOG DEVICES, 2007)

A function generator was used in sweep mode on a logarithmic scale to generate a plot showing the gain as a function of the frequency of the filters connected in series for the second test, which was to confirm the behavior of the implemented filters. As shown in Figure 12, the sweep's outcome. The oscilloscope's cursor mode was employed to depict the cutoff frequencies.

The cursors on the left and right were placed at different frequencies: 0.37 Hz and 40 Hz, respectively. This graphic demonstrates how the filters are operating as predicted, particularly the low-pass filter, which displays the two cutoff frequencies of their second-order sections (JÚNIOR et al., 2019). Figures 13 and 14 show the low-pass filter signal gain in 20Hz and 40Hz, where peak-to-peak voltage is attenuated from 3.44V in 20Hz to 2.16V in 40Hz.

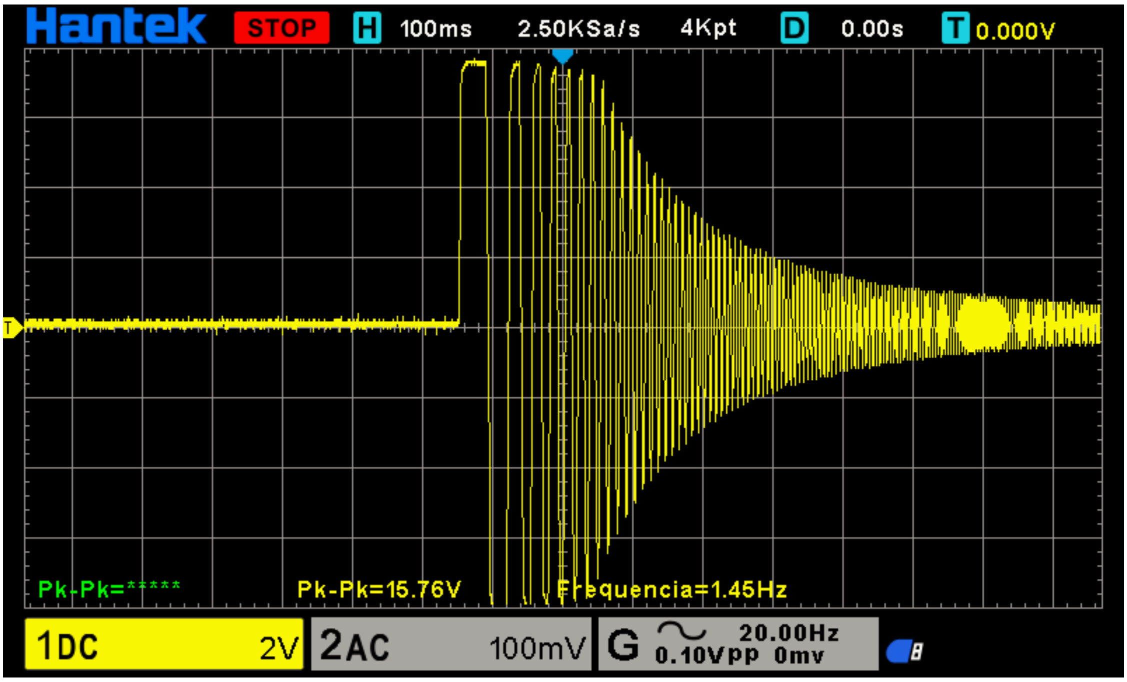


Figure 12 – Response of the gain as a function of frequency in logarithmic scale of the implemented filters. The cursor on the left indicates 0.37 Hz and the cursor on the right indicates 40Hz.

The third test was made to validate the operation of the band-stop filter. Since the band-stop filter's cutoff frequency was so restricted, it was required to check the amplifier's gain, exclusively for 60 Hz. Figures 15 and 16 show the band-stop filter signal gain in 20Hz and 60Hz, where peak-to-peak voltage is attenuated from 504mV in 20Hz to 424mV in 60Hz. Figure 17 shows the signal acquisition output and Figure 18 shows the typical ECG signal, where we can see that our circuit results show the typical ECG signal.

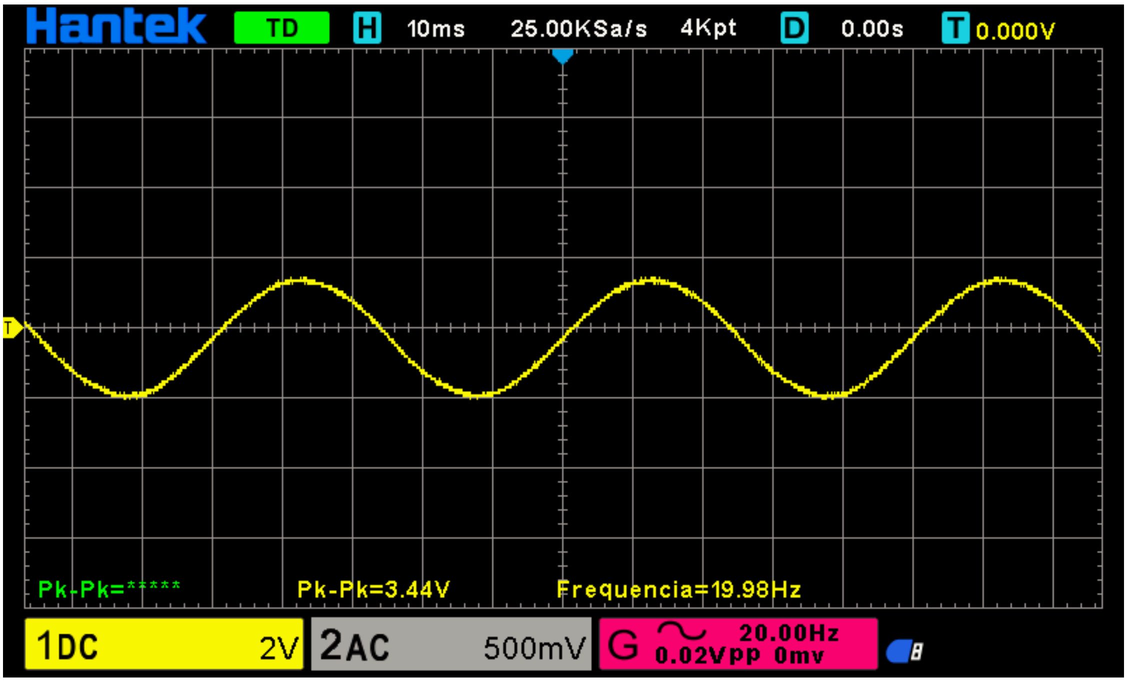


Figure 13 – Low pass filter response to 20Hz

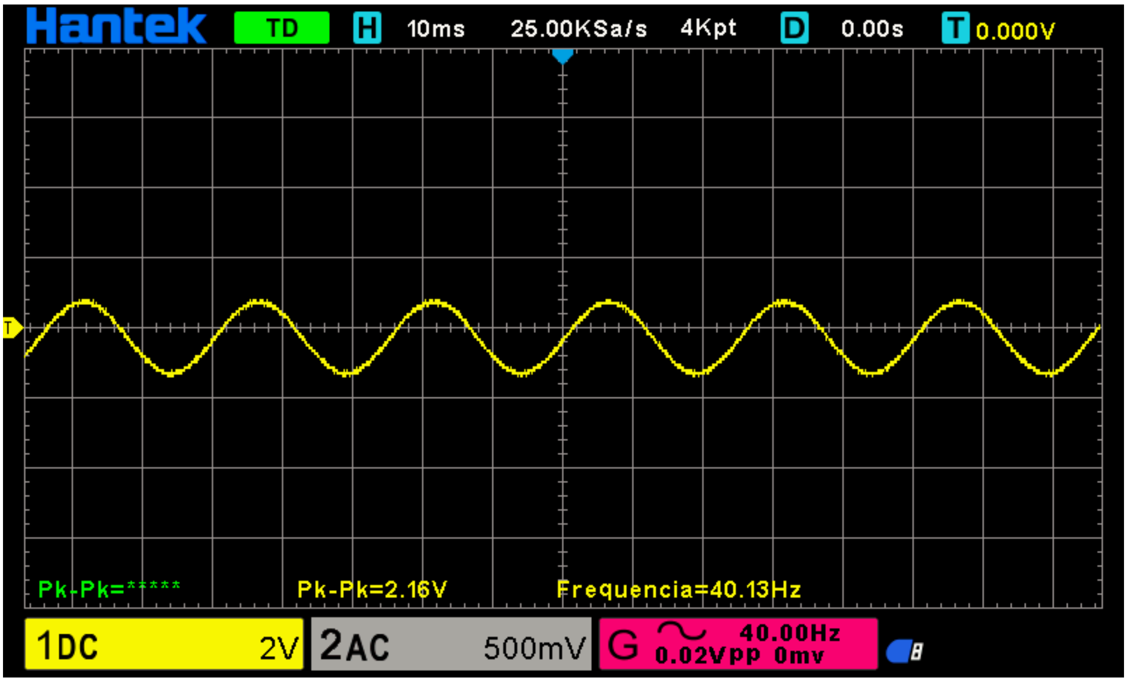
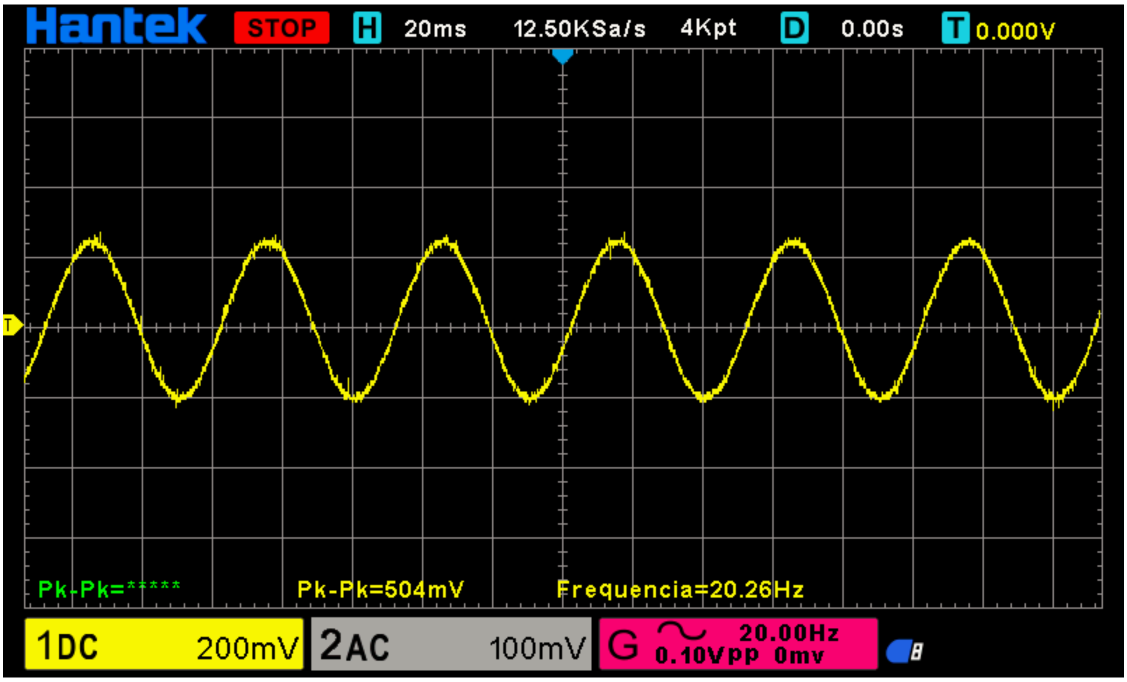


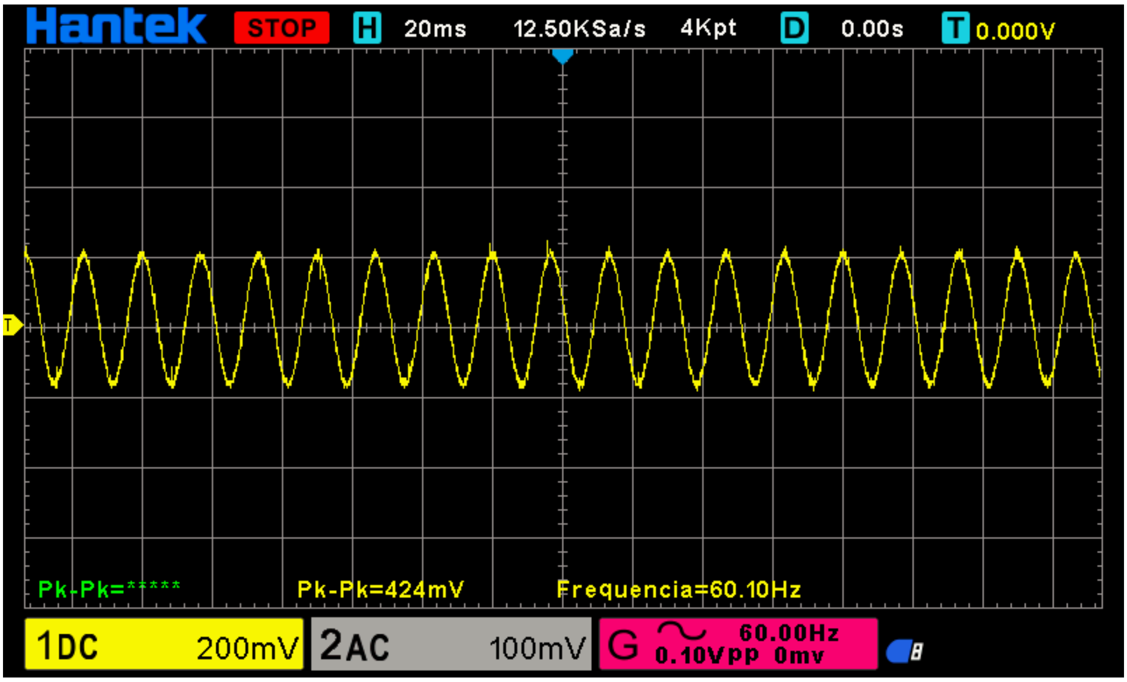
Figure 14 – Low pass filter response to 40Hz


Hantek STOP H 20ms 12.50KSa/s 4Kpt D 0.00s T 0.000V

Pk-Pk=***** Pk-Pk=504mV Frequencia=20.26Hz

1DC 200mV 2AC 100mV G 20.00Hz 0.10Vpp 0mv

Figure 15 – Band-stop filter response to 20Hz


Hantek STOP H 20ms 12.50KSa/s 4Kpt D 0.00s T 0.000V

Pk-Pk=***** Pk-Pk=424mV Frequencia=60.10Hz

1DC 200mV 2AC 100mV G 60.00Hz 0.10Vpp 0mv

Figure 16 – Band-stop filter response to 60Hz

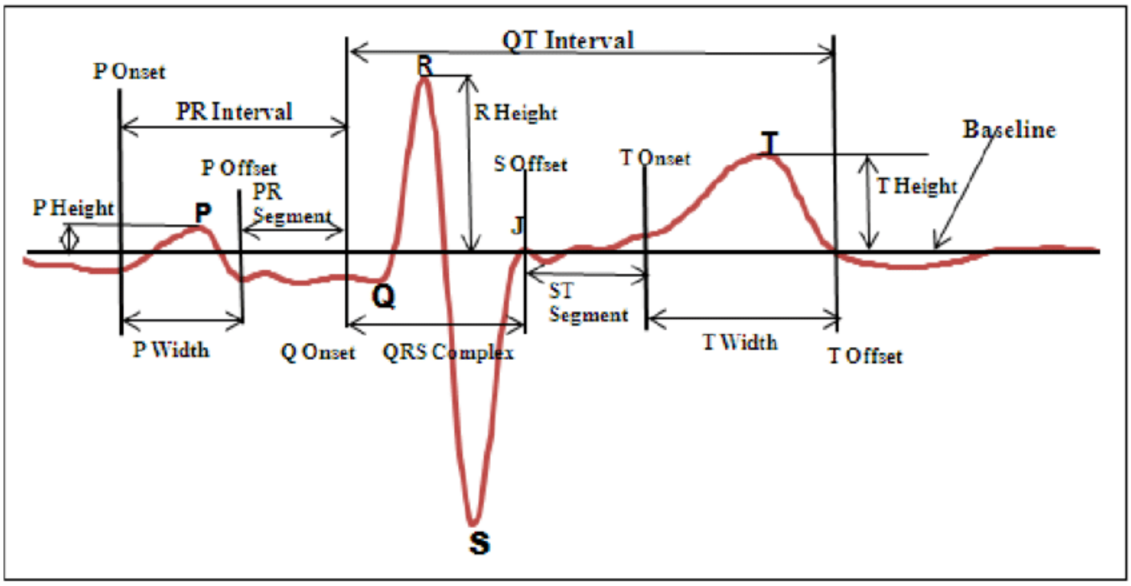


Figure 17 – Typical ECG signal (SACOMANO et al., 2018)

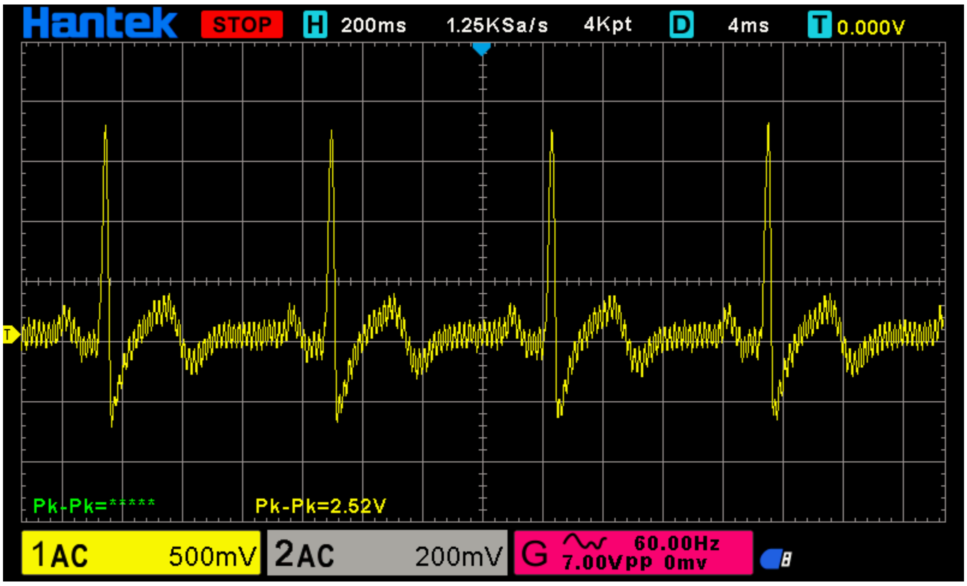


Figure 18 – Captured ECG signal

6 Classification of ECG signals

In this chapter, we will see the steps needed for heartbeat classification as discussed in Section 1.2.

- Develop the preprocessing code for the ECG signal.
- Develop the architecture of the CNN classifier.
- Develop the training method for the classifier.
- Evaluate the trained classifier by comparing the results of test accuracy, sensitivity, specificity, precision, and F1-score with those of other works.

Chapter 4 will show the prototyping step of the ECG signal acquisition board. Initially, a cardiac arrhythmia detection software will be developed through neural networks, using convolutional neural networks for continuous analysis of the data input, model training, and data classification.

The preprocessing step receives the MIT-BIH database and upsamples the data rows to have the same number of data rows for every heartbeat type. The CNN, CNN-LSTM and AlexNet classifiers will convolve the input through a unidimensional filter and will obtain the feature map, then reduce the size of the input and the number of parameters needed for computation in the network and prepare the data for training. In our CNN model, we convolve the input three times, triple-convolution, to obtain better accuracy. In our CNN-LSTM model, we use triple-convolution and LSTM to obtain better accuracy. In our AlexNet model, we doubled the number of triple-convolutions and used batch-normalization (BALESTRIERO; BARANIUK, 2022) to obtain better accuracy. The training model will process the input data using Adam optimizer (Equation 6.1) and will compare the processed output with the expected output. The result of this correlation is used to feed back new information and modify the model. We then compare the results of the modified model with other works to validate our classifier, chapter 7 will show the classifiers comparison.

6.1 Methodology

In this study, we created three classifiers based on CNN, CNN-LSTM and AlexNet for classifying arrhythmia. The architectures were fine-tuned to achieve the highest validation accuracy and F1-score, and were compared to the decision tree, random forest and extra trees classifiers (KUMAR, 2022). The implementation of the CNN[1], CNN-LSTM and AlexNet[2] are publicly available and were coded in Python using Tensorflow, Keras, and Numpy. The following topics describe the content of the next sections:

- The dataset section will describe the used dataset for the validation, training, and testing of our classifier.
- The preprocessing section will explain the ECG signal adjustments and dataset augmentation.
- The classifier architecture section will explain the architecture of our classifiers.
- The training method section will explain the steps taken for the fine-tuning and training of the classifiers.

6.1.1 Dataset

In this study, we used the ECG Heartbeat Categorization Dataset, freely available on the Internet[3]. We used only the portion of the dataset derived from the Physio Bank MIT-BIH Arrhythmia database (MARK; MOODY, 1988). This database consists of 48 half-hour long ECG recordings from 47 subjects—obtained with a Lead II ECG configuration—that were band-pass filtered over the frequency range from 0.1 to 100Hz and digitized at 360 samples per second. Furthermore, these recordings were interpreted and validated by at least two cardiologists. The database consists of annotations for both heartbeat class information and R-peak position information verified by two or more expert cardiologists. The 17 beat types can be grouped into five beat classes defined by the Association of Advancement for Medical Instrumentation (AAMI) which follows the American National Standard for Ambulatory ECGs (ANSI/AAMI EC38:2007) recommendations. The five beat types are the non-ectopic beat (N), supraventricular ectopic beat (S), ventricular ectopic beat (V), fusion beat (F), and unknown (Q).

[1] https://github.com/Igor-Lopes-Souza/VISAPP-2023
[2] https://github.com/Igor-Lopes-Souza/2023-CNN-LSTM
[3] https://www.kaggle.com/datasets/shayanfazeli/heartbeat

Table 3 – Number of samples in the training, validation, and test sets

	Before Data Augmentation			After data augmentation		
	Training	Validation	Test	Training	Validation	Test
N	57974	14493	18118	57974	14493	-
S	1778	450	556	57974	14493	-
V	4630	1155	1448	57974	14493	-
F	520	127	162	57974	14493	-
Q	5141	1286	1608	57974	14493	-
Total	70043	17511	21892	289870	72465	-

6.1.2 Preprocessing

The MIT-BIH dataset is unbalanced, difficulting the analysis of the signal. The original dataset contains a total of 109,446 data rows. Each data row contains a fraction of the ECG signal with a duration of 10 seconds and its class, specified in the last column by a number from 0 to 4 representing N, S, Q, F, and Q respectively. There are 70,043 data rows for training, 17,511 for validation and 21,892 for testing, making the proportions 65/15/20. We augmented the training and validation datasets to match the number of rows of the biggest class from the five types of heartbeats using the bootstrap process[4], as seen in Table 3.

6.1.3 Classifier architecture

Our study proposes the implementation of three architectures based on CNN, CNN-LSTM and AlexNet models.

Figure 19 shows the schematic of our CNN classifier. The network is composed of convolutional layers, subsampling layers, activation layers, merge layers, flatten layers, fully connected layers, and a softmax layer. The merge layer adds two layers, in our case the second convolution layer and the first activation function of each execution. Usually, each convolution layer is followed by a subsampling layer. In order to facilitate mapping between the heartbeat category and its waveform, we use a triple-convolution structure to achieve a better fitting capability (UCHIDA; TANAKA; OKUTOMI, 2018) in our CNN model. We tested eight CNN models with preprocessing and without preprocessing, these are divided into with subsampling and without subsampling, and tested with triple and simple convolution. The preprocessing tests compare the results of balanced datasets and unbalanced datasets, along with the subsampling tests that compare the results of using max-pooling without using max-pooling. In all tests, we compared the results of the metrics using our triple convolution and simple convolution. Figure 20 shows the structure of a triple-convolution layer sequence.

Figure 21 shows the CNN-LSTM classifier architecture, based on the CNN architecture,

[4] https:https://github.com/Igor-Lopes-Souza/2023-CNN-LSTM

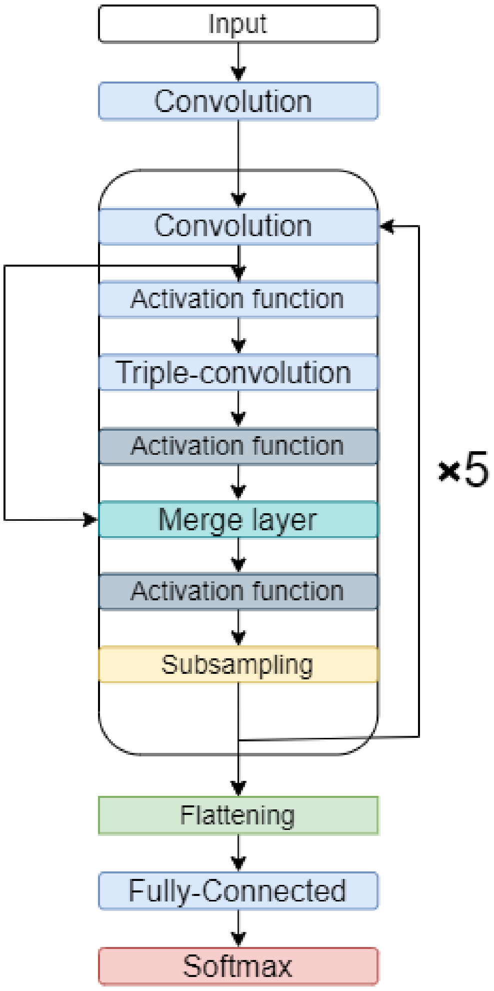


Figure 19 – CNN architecture

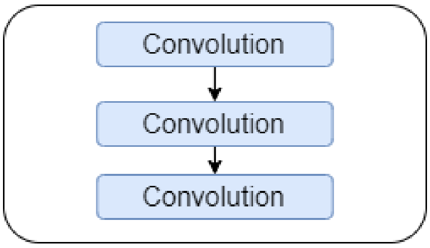


Figure 20 – Triple-convolution

which comprises convolutional layers, subsampling layers, fully connected layers, batch normalization layers, LSTM layers, and a dropout layer. We use the triple-convolution and the LSTM to obtain better accuracy results in our CNN-LSTM model. Figure 22 shows the schematic of our AlexNet classifier. The standard AlexNet classifier (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) is used for 2D image classification, while we modify its architecture for the analysis of ECG signals, which are 1D. The AlexNet classifier comprises convolutional layers, subsampling layers, fully connected layers, flatten layers, batch normalization layers, and dropout layers. In our AlexNet architecture when compared to the standard format, we doubled the number of

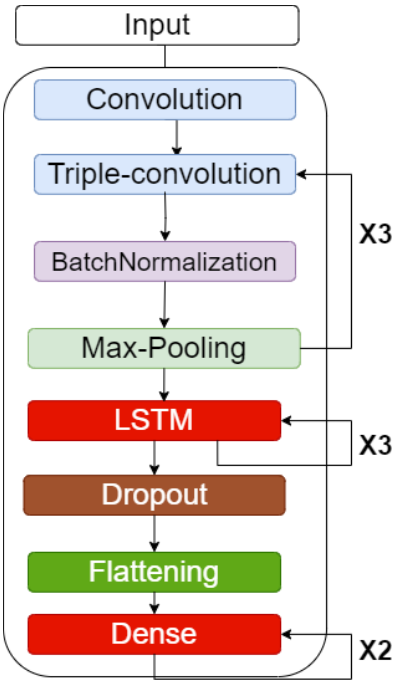


Figure 21 – CNN-LSTM architecture

triple-convolutions to obtain better accuracy and used the batch-normalization layer to normalize the interlayer outputs into a standard format.

The CNN, CNN-LSTM and AlexNet models were compared with extra trees, random forest and decision tree classifiers (ALOM et al., 2018; YU et al., 2019) that were trained with the sklearn default configuration and our training dataset (KRAMER; KRAMER, 2016). Among our models, the CNN-LSTM obtained the highest F1-score, the CNN obtained the highest accuracy and AlexNet obtained the highest sensitivity, chapter 7 will show the results of each classifier. The extra trees, decision trees and random forest classifiers architecture use the following parameters with their default values:

- minimal number of leaves:1
- minimal number of samples split: 2
- criterion: gini,
- maximum depth: None,
- maximum number of features': sqrt,
- maximum number of leaf nodes: None,
- maximum number of samples: None,
- number of estimators: 100

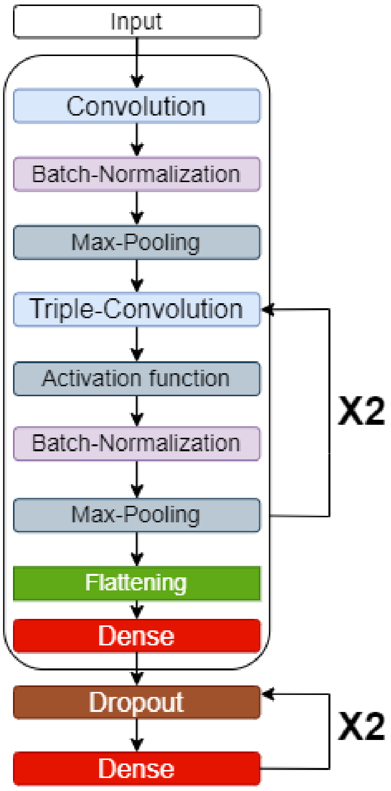


Figure 22 – AlexNet architecture

In this study, we use the ReLu activation function in both convolutional layers and fully connected layers (NAIR; HINTON, 2010; GIROSI; JONES; POGGIO, 1995). In the output layer, we use the softmax activation function to obtain the five heartbeat classes.

6.1.4 Training method

The goal of training is to reduce the value of the loss function L, i.e., to decrease the CNN, CNN-LSTM and AlexNet models loss and adjust the weights and biases so that Equation 6.1 fits the model training set. The cross-entropy function is used as the loss function (XU; LIU, 2020):

We update the weights and offsets using the Adam optimizer (KINGMA; BA, 2014). First, a batch of samples was sent to calculate the gradient of the Equation 6.1.

$$g = \left(\frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)}; \theta), y^{(i)}) \right). \tag{6.1}$$

where g is the gradient value, m is the batch size, θ is the parameter to be updated, $f(x^{(i)}; \theta)$ is the heartbeat type predicted by the i-th sample, $y^{(i)}$ is the actual type of the i-th sample, and L is the loss function.

The CNN-LSTM and AlexNet models training time of each epoch was approximately 20s, and the maximum epoch number was set to 100. In the CNN model, the training time of each epoch was approximately 5s, and the maximum epoch number was set to 75. We used an

Table 4 – Hyperparameter values chosen in CNN, CNN-LSTM and AlexNet classifiers fine-tuning.

Parameters	Values	Chosen Value
Dropout	0.10, 0.20, 0.30, 0.40, 0.50	0.50
Optimizer	Adam, Adamax, SGD	Adam
Activation function	Relu, Softmax, Softplus	Relu
Batch size	10, 32, 54, 76, 98	98
Loss function	Binary cross-entropy, Categorical cross-entropy, Poisson, Kullback-Leibler divergence, Huber	Categorical cross-entropy

NVIDIA GeForce GTC 1050 for the training. After defining the architecture, fine-tuning was performed to obtain the best values of dropout, optimizer, activation function, loss function, and batch size. A grid search in the hyperparameter space tested each possible combination with 20 epochs. Table 4 shows the tested hyperparameter values and the ones that maximized accuracy.

7

Classifier results

We performed classification experiments on 44 recordings from the MIT-BIH arrhythmia database, among the 48 recordings obtained from 47 patients studied by the BIH arrhythmia laboratory, and the heartbeats were classified according to the recommendation of the AAMI (KLIGFIELD et al., 2007), such as testing the classifier with signals with frequency between 40Hz and 0.4Hz.

The training dataset contains a total of 109,466 data rows of representative beats from all classes: type-N, non-ectopic beats; type-S, supraventricular ectopic beats; type-V, ventricular ectopic beats; type-F, fusion beats; and type-Q, unknown beats. Classification performance is measured using the statistical error metrics found in the literature (CHEN et al., 2022): accuracy (ACC), sensitivity (SEN), specificity (SPE), precision (PRE), and F1-score (F1S). The F1-score measures the overall performance of the beat classification, as shown in Table 5.

We made eight experiments with our CNN model, with all combinations of the following three parameters: with preprocessing or without preprocessing; with subsampling or without subsampling; with triple convolution or simple convolution. Table 6 shows the results of different proposed CNN architectures.

Table 5 shows that the CNN-LSTM model has an F1-score value comparable to those of other studies, presenting the second best results. Table 7 shows the results of the different proposed CNN architectures implementations. Table 8 shows the results of the comparison between the five types of heartbeats using the CNN, CNN-LSTM and AlexNet classifier. Our CNN (SOUZA; DANTAS, 2023) model achieved an accuracy of 99.33%, sensitivity of 99.59%, specificity of 99.30%, precision of 99.12% and F1-score of 99.44%. Our CNN-LSTM model achieved an accuracy of 98.12%, sensitivity of 99.00%, specificity of 98.85%, precision of 99.39%, and F1-score of 99.72%. Our AlexNet model achieved an accuracy of 96.66%, sensitivity of 99.45%, specificity of 99.10%, precision of 96.85%, and F1-score of 80.07%.

Figure 23 shows the confusion matrix of the classification results of the CNN-LSTM

Table 5 – Comparison of the proposed algorithm classification using ventricular ectopic beats (V)

	ACC	SEN	SPE	PRE	F1S
Martis (MARTIS et al., 2014)	99.45%	**99.61%**	**99.99%**	**99.99%**	**99.80%**
Proposed classifier: CNN-LSTM	98.12%	99.00%	98.85%	99.39%	99.72%
Proposed classifier: CNN (SOUZA; DANTAS, 2023)	99.33%	99.59%	99.30%	99.12%	99.44%
Sellami (SELLAMI; HWANG, 2019)	**99.48%**	96.97%	99.87%	98.83%	97.80%
Acharya (ACHARYA et al., 2017b)	94.03%	96.71%	91.54%	97.85%	97.27%
Zhai (ZHAI; TIN, 2018)	99.10%	96.40%	99.50%	96.40%	96.40%
Jiang (JIANG; KONG, 2007)	98.80%	94.30%	99.40%	95.30%	94.70%
Xiang (XIANG et al., 2018)	99.20%	93.70%	99.60%	94.80%	94.20%
Ince (INCE; KIRANYAZ; GABBOUJ, 2009)	97.60%	83.60%	98.10%	87.40%	85.40%
Proposed classifier: AlexNet	96.66%	99.45%	99.10%	96.85%	80.07%

Table 6 – Comparison of proposed CNN architecture implementations

			ACC	SEN	SPE	PRE	F1S
With preprocessing	With subsampling	Triple convolution	**99.33%**	**99.59%**	**99.30%**	**99.12%**	**99.44%**
		Simple convolution	95.32%	95.73%	98.83%	96.39%	95.40%
	Without subsampling	Triple convolution	95.40%	95.27%	94.70%	95.25%	95.35%
		Simple convolution	90.45%	90.14%	92.83%	90.89%	90.44%
Without preprocessing	With subsampling	Triple convolution	89.65%	89.00%	97.41%	91.63%	89.50%
		Simple convolution	87.85%	83.25%	96.06%	90.63%	87.56%
	Without subsampling	Triple convolution	86.68%	88.69%	96.80%	90.05%	86.67%
		Simple convolution	87.20%	90.40%	90.50%	89.20%	85.80%

test set. The model is able to make accurate predictions and distinguish different classes. The main reason behind this might be due to the fine-tuning of our model, as unrefined tests with ventricular ectopic beats (V) obtained an average accuracy of 89.99% and an F1-score of 88.54%.

Table 7 – Comparison of proposed implementations using ventricular ectopic beats (V)

	ACC	SEN	SPE	PRE	F1S
CNN-LSTM	98.12%	99.00%	98.85%	99.39%	**99.72**%
CNN	**99.33**%	**99.59**%	**99.30**%	**99.12**%	99.44%
Decision tree classifier	99.11%	99.22%	99.16%	99.00%	99.36%
Random forest classifier	95.32%	95.73%	98.83%	96.39%	95.40%
Extra trees classifier	95.40%	95.27%	94.70%	95.25%	95.35%
AlexNet	96.66%	99.45%	99.10%	96.85%	80.07%

Table 8 – Comparison of the types of heartbeats

	ACC	SEN	SPE	PRE	F1S
Normal (N)	99.45%	99.98%	92.83%	90.89%	99.44%
Supraventricular ectopic beats (S)	97.39%	88.61%	98.92%	88.92%	95.25%
Ventricular ectopic beats (V)	98.12%	99.00%	98.85%	99.39%	99.72%
Fusion Beats (F)	87.40%	77.27%	84.70%	95.25%	82.35%
Unknown Beats (Q)	99.32%	99.65%	98.72%	97.20%	99.10%

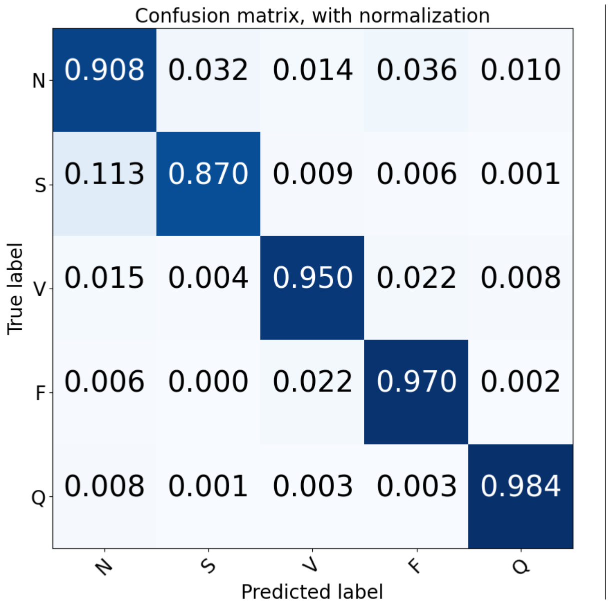


Figure 23 – Confusion matrix for heartbeat classification on the test set.

8
Conclusions

This project aims to classify ECG signals for cardiac arrhythmia detection using CNNs and the implementation of an electrocardiogram signal acquisition circuit. We developed a CNN classifier for ECG signal classification with an accuracy of 98.12% and an F1-score of 99.72% in the classification of ventricular ectopic beats (V). In order to optimize our model, we fine-tuned our variables and functions, the selected values compose our final version of the classifier and are displayed in Table 4. Compared with the methods in previous literature, our model performed better in terms of V classification F1-score, only being surpassed by Martis (MARTIS et al., 2014).

We designed an ECG acquisition board circuit and the test results show that our signal gain and cut frequency are operating in the ECG delimited parameters and are capturing ECG signals, as shown in Figure 17. Our tested gain output is 28.8V/V and the frequency cut is 40Hz.

Our study may be refined by using a better set of hyperparameter values and different augmentation strategies for our classifier. Other projects may implement, test, and validate the circuit board for signal gain and noise reduction. Future work may test the connection of the Raspberry Pi to the ECG signal acquisition board.

Bibliography

ACHARYA, U. R. et al. Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network. *Information sciences*, Elsevier, v. 405, p. 81–90, 2017. Citado na página 17.

ACHARYA, U. R. et al. A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, Elsevier, v. 89, p. 389–396, 2017. Citado 2 vezes nas páginas 17 and 42.

ALCARAZ, R. et al. Wavelet sample entropy: A new approach to predict termination of atrial fibrillation. In: IEEE. *2006 Computers in Cardiology*. [S.l.], 2006. p. 597–600. Citado na página 19.

ALOM, M. Z. et al. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018. Citado na página 38.

AMINE, K.; REDOUANE, K. M.; NARIMA, Z. ECG signal protection for telemedicine applications. *Circuits, Systems, and Signal Processing*, Springer, p. 1–16, 2022. Citado na página 4.

AMPLIFIERS, T. J.-I. O. Datasheet; texas instruments. *Inc.: Dallas, TX, USA*, 1999. Citado na página 25.

ANALOG DEVICES. *10 MHz,* $20V/\mu s$*, G = 1, 10, 100, 1000 iCMOS®*. [S.l.], 2007. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/200399/AD/AD620.html>. Citado 4 vezes nas páginas 21, 22, 23, and 29.

BALESTRIERO, R.; BARANIUK, R. G. Batch normalization explained. *arXiv preprint arXiv:2209.14778*, 2022. Citado na página 34.

BERRAR, D. *Cross-Validation.* 2019. Citado na página 9.

BOTALB, A. et al. Contrasting convolutional neural network (cnn) with multi-layer perceptron (mlp) for big data analysis. In: IEEE. *2018 International conference on intelligent and advanced system (ICIAS)*. [S.l.], 2018. p. 1–5. Citado na página 14.

CHEN, S. W. et al. Review of ECG detection and classification based on deep learning: Coherent taxonomy, motivation, open challenges and recommendations. *Biomedical Signal Processing and Control*, Elsevier, v. 74, p. 103493, 2022. Citado 3 vezes nas páginas 4, 5, and 41.

GARCÍA, M. et al. Application of the relative wavelet energy to heart rate independent detection of atrial fibrillation. *computer methods and programs in biomedicine*, Elsevier, v. 131, p. 157–168, 2016. Citado na página 19.

GIROSI, F.; JONES, M.; POGGIO, T. Regularization theory and neural networks architectures. *Neural computation*, MIT Press, v. 7, n. 2, p. 219–269, 1995. Citado na página 39.

GOIS, D. A. S. et al. An experiment to assess an acquisition platform and biomedical signal conditioning. In: SCITEPRESS. *International Conference on Enterprise Information Systems*. [S.l.], 2017. v. 2, p. 75–86. Citado na página 20.

GRAVES, A.; GRAVES, A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, Springer, p. 37–45, 2012. Citado na página 14.

Guimarães, L. M. S.; Meireles, M. R. G.; Almeida, P. E. M. de. Avaliação das etapas de pré-processamento e de treinamento em algoritmos de classificação de textos no contexto da recuperação da informação. Perspectivas em Ciência da Informação, Belo Horizonte, v. 24, n. 1, p. 169–190, 05 2019. Citado na página 8.

HAGIWARA, Y. et al. Computer-aided diagnosis of atrial fibrillation based on ecg signals: A review. *Information Sciences*, Elsevier, v. 467, p. 99–114, 2018. Citado na página 4.

HAN, C.; SHI, L. Ml–resnet: A novel network to detect and locate myocardial infarction using 12 leads ecg. *Computer methods and programs in biomedicine*, Elsevier, v. 185, p. 105138, 2020. Citado na página 17.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 14.

HUANG, Y. C. et al. Prediction of mortality using standard 12-lead electrocardiogram with deep neural network. *Circulation*, Am Heart Assoc, v. 142, n. Suppl_3, p. A15588–A15588, 2020. Citado na página 17.

INCE, T.; KIRANYAZ, S.; GABBOUJ, M. A generic and robust system for automated patient-specific classification of ECG signals. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 56, n. 5, p. 1415–1426, 2009. Citado na página 42.

JIANG, W.; KONG, S. G. Block-based neural networks for personalized ECG signal classification. *IEEE Transactions on Neural Networks*, IEEE, v. 18, n. 6, p. 1750–1761, 2007. Citado na página 42.

JÚNIOR, J. D. L. S. et al. Projeto e desenvolvimento de um sistema de eletromiografia de superfície. In: *Congresso Brasileiro de Automática-CBA*. [S.l.: s.n.], 2019. v. 1, n. 1. Citado 2 vezes nas páginas 21 and 30.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 39.

KIRANYAZ, S. et al. 1d convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, Elsevier, v. 151, p. 107398, 2021. Citado na página 4.

KIRANYAZ, S.; INCE, T.; GABBOUJ, M. Real-time patient-specific ECG classification by 1-d convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 63, n. 3, p. 664–675, 2015. Citado na página 16.

KIRANYAZ, S.; INCE, T.; GABBOUJ, M. Personalized monitoring and advance warning system for cardiac arrhythmias. *Scientific reports*, Nature Publishing Group, v. 7, n. 1, p. 1–8, 2017. Citado na página 16.

KLIGFIELD, P. et al. Recommendations for the standardization and interpretation of the electrocardiogram: part i: the electrocardiogram and its technology: a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology. *Circulation*, Am Heart Assoc, v. 115, n. 10, p. 1306–1324, 2007. Citado na página 41.

KRAMER, O.; KRAMER, O. Scikit-learn. *Machine learning for evolution strategies*, Springer, p. 45–53, 2016. Citado na página 38.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012. Citado 2 vezes nas páginas 17 and 37.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 60, n. 6, p. 84–90, may 2017. ISSN 0001-0782. Citado na página 13.

KUMAR, V. Analysis of cnn features with multiple machine learning classifiers in diagnosis of monkeypox from digital skin images. *medRxiv*, Cold Spring Harbor Laboratory Press, p. 2022–09, 2022. Citado na página 35.

LI, Z. et al. Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram. *Journal of Electrocardiology*, Elsevier, v. 58, p. 105–112, 2020. Citado na página 17.

LIU, W. et al. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*, 2016. Citado na página 14.

LOWN, M. et al. Machine learning detection of atrial fibrillation using wearable technology. *PLoS One*, Public Library of Science San Francisco, CA USA, v. 15, n. 1, p. e0227401, 2020. Citado na página 4.

MANT, J. et al. Accuracy of diagnosing atrial fibrillation on electrocardiogram by primary care practitioners and interpretative diagnostic software: analysis of data from screening for atrial fibrillation in the elderly (safe) trial. *Bmj*, British Medical Journal Publishing Group, v. 335, n. 7616, p. 380, 2007. Citado na página 4.

MARK, R.; MOODY, G. Mit-bih arrhythmia database directory. *Cambridge: Massachusetts Institute of Technology*, 1988. Citado na página 35.

MARSLAND, S. *Machine learning: an algorithmic perspective*. [S.l.]: CRC press, 2015. Citado 2 vezes nas páginas 7 and 11.

MARTIS, R. J. et al. Computer aided diagnosis of atrial arrhythmia using dimensionality reduction methods on transform domain representation. *Biomedical signal processing and control*, Elsevier, v. 13, p. 295–305, 2014. Citado 3 vezes nas páginas 18, 42, and 44.

MCALOON, C. J. et al. The changing face of cardiovascular disease 2000–2012: An analysis of the world health organisation global health estimates data. *International journal of cardiology*, Elsevier, v. 224, p. 256–264, 2016. Citado na página 4.

MICROCHIP TECHNOLOGY. *2.7V, 4-Channel/8-Channel 10-Bit A/D Converterswith SPI Serial Interface*. [S.l.], 2008. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/pdf/304549/MICROCHIP/MCP3008.html>. Citado na página 27.

Mitchell, T. M. *Machine Learning*. 1. ed. Nova York: McGraw-Hill International Editions, 1997. 14 p. ISBN 978-00-7115-467-3. Citado na página 7.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: *Icml*. [S.l.: s.n.], 2010. Citado na página 39.

Neves, R. de Cássia David das. *Pré-processamento no processo de descoberta de conhecimento em banco de dados*. [S.l.]: Dissertação de Mestrado - UFRGS Lume, 2003. Citado na página 8.

POLLOCK, K. G. et al. Application of a machine learning algorithm for detection of atrial fibrillation in secondary care. *IJC Heart & Vasculature*, Elsevier, v. 31, p. 100674, 2020. Citado na página 4.

RAHHAL, M. M. A. et al. Dense convolutional networks with focal loss and image generation for electrocardiogram classification. *IEEE Access*, IEEE, v. 7, p. 182225–182237, 2019. Citado na página 19.

RASPBERRY PI (TRADING) LTD. *Raspberry Pi 4 Model B*. [S.l.], 2019. Disponível em: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>. Citado 2 vezes nas páginas 20 and 28.

ROCKWELL, K. L.; GILROY, A. S. Incorporating telemedicine as part of covid-19 outbreak response systems. *Am J Manag Care*, v. 26, n. 4, p. 147–148, 2020. Citado na página 4.

RÓDENAS, J. et al. Wavelet entropy automatically detects episodes of atrial fibrillation from single-lead electrocardiograms. *Entropy*, MDPI, v. 17, n. 9, p. 6179–6199, 2015. Citado na página 19.

Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. USA: Prentice Hall Press, 2010. ISBN 0136042597. Citado na página 8.

SACOMANO, J. B. et al. *Indústria 4.0*. [S.l.]: Editora Blucher, 2018. Citado na página 33.

SARKAR, S.; RITSCHER, D.; MEHRA, R. A detector for a chronic implantable atrial tachyarrhythmia monitor. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 55, n. 3, p. 1219–1224, 2008. Citado na página 19.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, Elsevier BV, v. 61, p. 85–117, Jan 2015. ISSN 0893-6080. Citado na página 13.

SCHWAB, P. et al. Beat by beat: Classifying cardiac arrhythmias with recurrent neural networks. In: IEEE. *2017 Computing in Cardiology (CinC)*. [S.l.], 2017. p. 1–4. Citado na página 19.

SELLAMI, A.; HWANG, H. A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. *Expert Systems with Applications*, Elsevier, v. 122, p. 75–84, 2019. Citado 2 vezes nas páginas 18 and 42.

SEMICONDUCTOR, O. *LM317 datasheet*. [S.l.]: January, 2016. Citado 2 vezes nas páginas 25 and 26.

SHAH, D. et al. Iot based biometrics implementation on raspberry pi. *Procedia Computer Science*, Elsevier, v. 79, p. 328–336, 2016. Citado na página 27.

Shao, L.; Zhu, F.; Li, X. Transfer learning for visual categorization: a survey. *IEEE transactions on neural networks and learning systems*, v. 26, n. 5, p. 1019—1034, May 2015. ISSN 2162-237X. Citado na página 12.

SHIH, D.-H. et al. An embedded mobile ECG reasoning system for elderly patients. *IEEE Transactions on Information Technology in Biomedicine*, IEEE, v. 14, n. 3, p. 854–865, 2009. Citado na página 16.

SHOEMAKER, M. B. et al. How will genetics inform the clinical care of atrial fibrillation? *Circulation research*, Am Heart Assoc, v. 127, n. 1, p. 111–127, 2020. Citado na página 4.

SILVA, I. A. R. da et al. Low cost hardware and software platform for multichannel surface electromyography. In: IEEE. *2018 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.], 2018. p. 01114–01119. Citado na página 25.

Simard, P.; Steinkraus, D.; Platt, J. Best practices for convolutional neural networks applied to visual document analysis. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* [S.l.: s.n.], 2003. p. 958–963. Citado na página 12.

SOUZA, I.; DANTAS, D. Cardiac arrhythmia classification in electrocardiogram signals with convolutional neural networks. In: *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science and Technology Publications, 2023. Disponível em: <https://doi.org/10.5220%2F0011682800003411>. Citado 2 vezes nas páginas 41 and 42.

SPINELLI, E. M.; MAYOSKY, M. A. Ac coupled three op-amp biopotential amplifier with active dc suppression. *IEEE transactions on biomedical engineering*, IEEE, v. 47, n. 12, p. 1616–1619, 2000. Citado 2 vezes nas páginas 21 and 23.

Tan, P.-N.; Steinbach, M.; Kumar, V. *Introdução a Datamining Mineração de Dados*. 2. ed. Rio de Janeiro: Person Education, Inc., 2006. 3,23,27,171-174,176-178,204-207,218-222,291-292,303-306 p. ISBN 978-85-7393-761-9. Citado 3 vezes nas páginas 7, 8, and 10.

TATENO, K.; GLASS, L. Automatic detection of atrial fibrillation using the coefficient of variation and density histograms of rr and δrr intervals. *Medical and Biological Engineering and Computing*, Springer, v. 39, n. 6, p. 664–671, 2001. Citado na página 19.

UCHIDA, K.; TANAKA, M.; OKUTOMI, M. Coupled convolution layer for convolutional neural network. *Neural Networks*, Elsevier, v. 105, p. 197–205, 2018. Citado na página 36.

XIANG, Y. et al. ECG-based heartbeat classification using two-level convolutional neural network and rr interval difference. *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, v. 101, n. 4, p. 1189–1198, 2018. Citado na página 42.

XU, X.; LIU, H. ECG heartbeat classification using convolutional neural networks. *IEEE Access*, IEEE, v. 8, p. 8614–8619, 2020. Citado na página 39.

Yamashita, R. et al. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, v. 9, 06 2018. Citado na página 12.

YU, Y. et al. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 31, n. 7, p. 1235–1270, 2019. Citado na página 38.

ZHAI, X.; TIN, C. Automated ECG classification using dual heartbeat coupling based on convolutional neural network. *IEEE Access*, IEEE, v. 6, p. 27465–27472, 2018. Citado 2 vezes nas páginas 18 and 42.

ZHIQIANG, W.; JUN, L. A review of object detection based on convolutional neural network. In: IEEE. *2017 36th Chinese control conference (CCC)*. [S.l.], 2017. p. 11104–11109. Citado na página 4.

ZHU, H. et al. Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: a cohort study. *The Lancet Digital Health*, Elsevier, v. 2, n. 7, p. e348–e357, 2020. Citado na página 18.