Escuela Politécnica Superior de Linares

Master's Degree of Science in Telecommunication Engineering

Universidad de Jaén
Escuela Politécnica Superior de Linares

Retrieving heartbeat information from phonocardiogram

Student: Ivan Denisov

Tutor: Prof. Dr. Sebastián García Galán

Tutor: Prof. Dr. José Enrique Muñoz Expósito

Dept.: Departamento de Ingeniería de Telecomunicación

February, 2024

Universidad de Jaén

Escuela Politécnica Superior de Linares

Master Thesis

Retrieving heartbeat information from phonocardiogram

Student: Ivan Denisov

Tutor: Prof. Dr. Sebastián García Galán

Tutor: Prof. Dr. José Enrique Muñoz Expósito

DENISOV IVAN - Y6478858 W

Digitally signed by DENISOV IVAN - Y6478858W
Date: 2024.02.22 18:31:48 +01'00'

Student Signature | Tutor Signature | Tutor Signature

February, 2024

ABSTRACT

In this Master Thesis, several algorithms were implemented. In the first place, the electrocardiography (ECG) and phonocardiography (PCG) simultaneous data is acquired from open-source databases.

Then signal processing techniques are implemented to extract the necessary features from ECG and PCG signals. The processing is made using filters, sophisticated signal processing techniques, Pan Tompkins algorithm, bringing a stable and reliable solution to extract the features.

Once the necessary features are extracted from the signals, machine learning algorithms are implemented in order to predict the QT interval from PCG features. The results are compared to other previously obtained results.

RESUMEN

En este Trabajo Fin de Máster, se implementaron varios algoritmos. En primer lugar, se adquieren los datos simultáneos de electrocardiografía (ECG) y fonocardiografía (PCG) de bases de datos de acceso abierto.

Luego, se implementan técnicas de procesamiento de señales para extraer las características necesarias de las señales de ECG y PCG. El procesamiento se realiza utilizando filtros, técnicas sofisticadas de procesamiento de señales, el algoritmo de Pan Tompkins, brindando una solución estable y confiable para extraer las características.

Una vez que se extraen las características necesarias de las señales, se implementan algoritmos de aprendizaje automático para predecir el intervalo QT a partir de las características de PCG. Los resultados se comparan con otros resultados obtenidos previamente.

INDEX

Content:

INDEX OF FIGURES AND TABLES

Index of Figures

Index of Tables

ACRONYMS

API – Application Programming Interface

ECG – Electrocardiogram

EEG – Electroencephalogram

EMG – Electromyography

HR – Heart Rate

ICA – Independent component analysis

MECGB – Median ECG beat

PCG – Phonocardiogram

QT – Interval between the waves Q and T in the electrocardiography

QTc – Corrected interval between the waves Q and T in the electrocardiography

QRS – Interval of waves Q, R and S in the electrocardiography

STFT – Short-time Fourier transform

WFDB – Waveform-database

1. INTRODUCTION

In this chapter, an introduction to this Master Thesis is presented, defining the purpose and motivation, the objectives, and the initial hypothesis on which the development of the work will be based.

1.1 Objectives

The Goals of this Master Thesis were the next ones:

I. Use of public databases with Electrocardiogram (ECG) and Phonocardiogram (PCG)

II. Extraction of heartbeat information using electrocardiogram in order to get QRS complexes and QT intervals from ECG.

III. Extraction of sound heartbeat information from phonocardiogram getting sounds from every single heartbeat (S1 and S2) in order to estimate QT intervals. To do that information extracted from ECG signals will be considered.

IV. Estimation of QT intervals bearing in mind the aforementioned information. In order to estimate heartbeat information from PCG, some artificial intelligence techniques will be used considering information from ECG for training purposes.

1.2 Purpose and Motivation

The purpose and motivation of this Master's Thesis revolve around the development of a prediction model with the capability to accurately estimate the QT interval from phonocardiographic signals. The motivation is rooted in the complexity associated with acquiring electrocardiographic signals using advanced medical devices and personnel. In contrast, phonocardiographic signals can be obtained through simpler means, such as small wearable devices.

Prolongation of the QT interval is a crucial indicator, associated with early afterdepolarizations that can lead to cardiac arrhythmias and, in severe cases, ventricular fibrillation and sudden cardiac death. Conversely, a shortened QT interval is linked to increased heterogeneity of repolarization in time and space.

Given the complexity of obtaining the QT interval through traditional methods, this Master Thesis aims to explore the possibility of estimating it using phonocardiography, a more accessible and straightforward approach.

The proposed methodology involves extracting features from both electrocardiographic (ECG) and phonocardiographic (PCG) signals. The obtained data will be preprocessed and used as input for training and validating a prediction model. The goal is to establish the reliability of this prediction model as a means of estimating the QT interval.

If successfully implemented, this solution has the potential to simplify QT interval detection, providing valuable support to medical professionals and individuals with various medical conditions. Early identification of prolonged or shortened QT intervals could significantly contribute to preventing the risk of sudden cardiac death, thereby enhancing overall patient care and well-being.

1.3 Starting Hypothesis

Nowadays, a lot of different studies were made in electrocardiographic processing and phonocardiographic processing. The features of these signals separately or together were used to automatically find different types of diseases.

However, there are few studies in which using a phonocardiographic information, an electrocardiographic information was tried to be estimated. In some previous studies it was demonstrated that obtaining some information about QT interval from PCG is possible, but an error was considerable and not many features of phonocardiographic signals were used.

In this Master Thesis, the hypothesis is that implementing reliable processing techniques to extract ECG and PCG features, developing an accurate neural network and considering more information from PCG signal, can bring better results that actually can be considered to estimate the QT interval.

1.4 Organization of the document

The presented document consists of 10 numbered chapters. Below is a brief summary of each of these chapters:

- Chapter 1 provides an introduction to the work, defining the motivation, purpose, objectives, and starting hypothesis of the work.
- Chapter 2 presents the state of the art of the Master's Thesis, the theoretical information that may be useful for better understanding the work developed, and the scope in which it has been implemented.
- Chapters 3 and 4 present the list of software used, explaining the specific use given to each component. Additionally, the proposed solution is summarized with all the phases included in this project.
- Chapter 5 develops and explains the processing of the ECG and PCG signals, the way of handling the noise on these signals and the features extracted from both signals. Finally, the extracted features are processed and correlated in order to be ready to be the input of prediction techniques
- Chapter 6 develops the different regression models used to correctly predict the QT interval, using the information obtained in the previous chapter. .
- Chapter 7 develops an economic study of the project's costs.
- Chapters 8 and 9 develop the conclusions of the completed project and propose possible future lines of work that could be undertaken to expand the scope of the project.
- Chapter 10 lists the bibliographic references used throughout the work in the order of their use in the present document.

2. STATE OF THE ART

In the State of the Art of this Master Thesis is made an introduction to the main topics and tools that will be used and discussed in the subsequent chapters. Also, the previous studies related to the topic are presented.

2.1 Introduction to ECG

At every beat, the heart is depolarized to trigger its contraction. This electrical activity is transmitted throughout the body and can be picked up on the skin. This is the principle behind the ECG. An ECG machine records this activity via electrodes on the skin and displays it graphically. An ECG involves attaching 10 electrical cables to the body: one to each limb and six across the chest.

The ECG machine processes the signals picked up from the skin by electrodes and produces a graphic representation of the electrical activity of the patient's heart. The basic pattern of the ECG is logical:

- electrical activity towards a lead causes an upward deflection
- electrical activity away from a lead causes a downward deflection
- depolarization and repolarization deflections occur in opposite directions

The basic pattern of this electrical activity was first discovered over a hundred years ago. It comprises three waves, which have been named P, QRS (a wave complex), and T. [1]

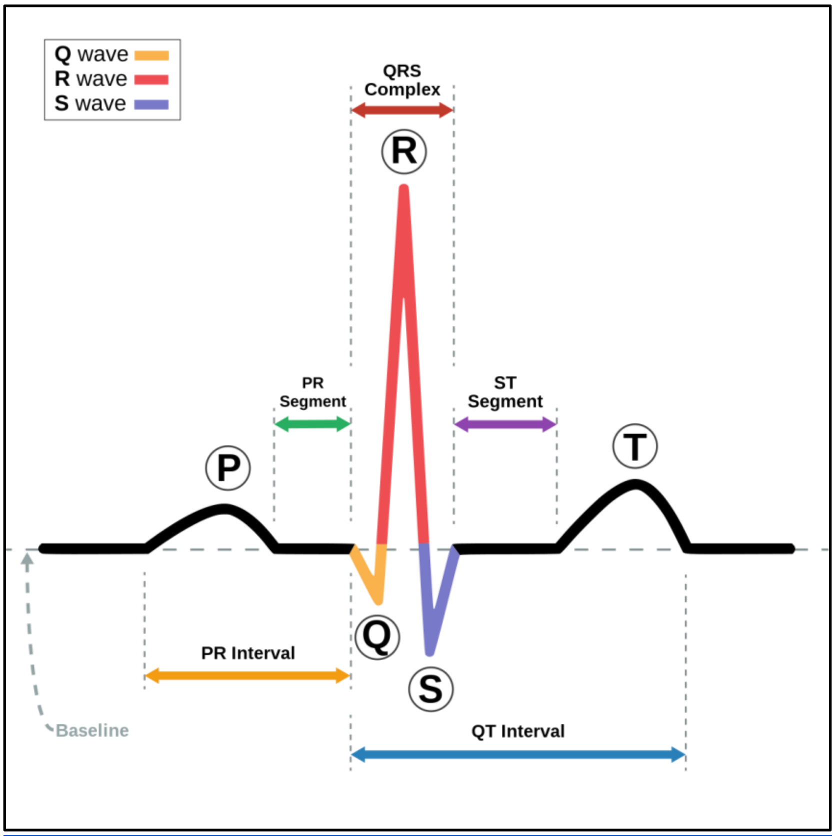

Figure 1 – Electrocardiography

P wave

The P wave is a small deflection wave that represents atrial depolarization.

PR interval

The PR interval is the time between the first deflection of the P wave and the first deflection of the QRS complex.

QRS wave complex

The three waves of the QRS complex represent ventricular depolarization. For the inexperienced, one of the most confusing aspects of ECG reading is the labeling of these waves. The rule is: if the wave immediately after the P wave is an upward deflection, it is an R wave; if it is a downward deflection, it is a Q wave:

- Small Q waves correspond to depolarization of the interventricular septum. Q waves can also relate to breathing and are generally small and thin. They can also signal an old myocardial infarction (in which case they are big and wide).
- The R wave reflects depolarization of the main mass of the ventricles –hence it is the largest wave.
- The S wave signifies the final depolarization of the ventricles, at the base of the heart.

ST segment

The ST segment, which is also known as the ST interval, is the time between the end of the QRS complex and the start of the T wave. It reflects the period of zero potential between ventricular depolarization and repolarization.

T wave

T waves represent ventricular repolarization (atrial repolarization is obscured by the large QRS complex).

ECG Obtaining

The standard ECG uses 10 cables to obtain 12 electrical views of the heart. The different views reflect the angles at which electrodes "look" at the heart and the direction of the heart's electrical depolarization.

This way, obtaining the ECG is not a simple task and must be carried out by a medical staff in special medical conditions. [1]

2.2 Introduction to PCG

A phonocardiogram (PCG) is a waveform of heart sounds. It’s a two-dimensional plot of sound intensity (loudness) versus time. Sound intensity is shown on the Y-axis, and time is shown on the X-axis. PCGs allow to visually group sounds with more ease than when using ear alone. [2]

Physiologically, the mechanical movement of the heart, and in particular the closing of the cardiac valves during each heartbeat, causes vibrations in the myocardium wall, which reflect as audible sounds, known as heart sounds. In each cardiac cycle, two main heart sounds are expected to subsequently occur, namely the first (S1) and second (S2) heart sounds, corresponding to the closing of the mitral and tricuspid valves, and of the aortic and pulmonary valves, respectively. Beyond them, two low-frequency sounds, named third (S3) and fourth (S4) heart sounds, may be present, in addition to murmurs generated by sudden flow turbulence, often associated with pathological conditions. [3]

The two fundamental heart sounds, S1 and S2, are shown on a PCG as large magnitude deflections occurring one after the other. S1 marks the beginning of systole; S2 marks the beginning of diastole. Depending upon where on the body the sounds were captured, the S1 deflection may be larger, S2 may be larger, or the sounds may be the same size (loudness). At other times the heart is silent, so the PCG will be flat. [2]

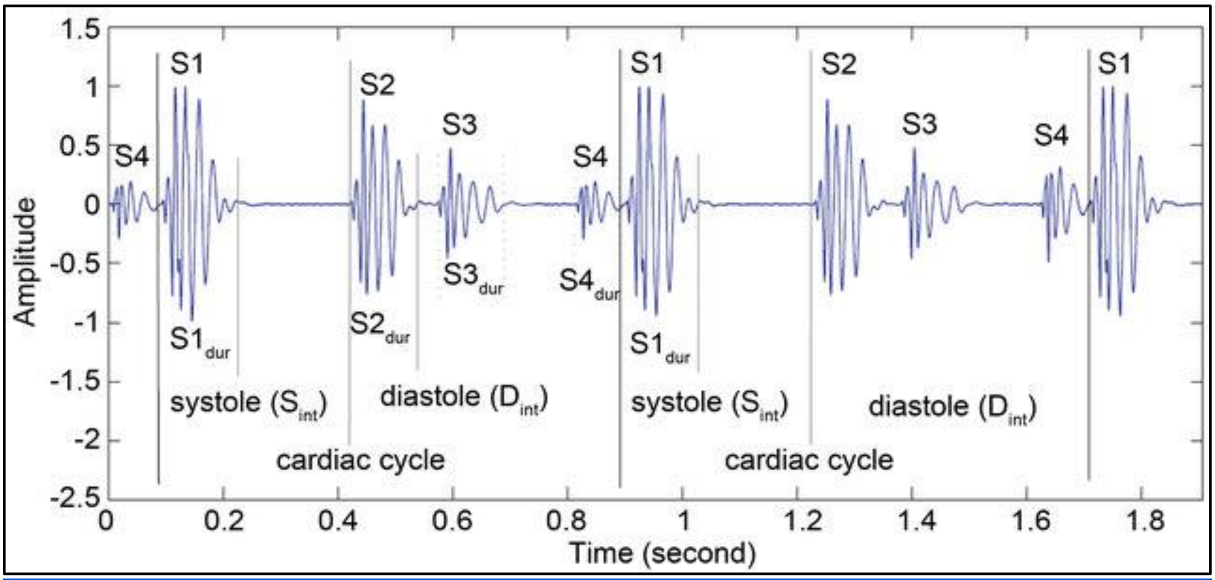

Figure 2 – Phonocardiography

A wide range of segmentation techniques has been implemented and described in the different papers. The methods that have been implemented were based on envelope-based approaches, wavelet-based approaches, machine learning techniques, methods based on time-frequency analysis and combinations of the mentioned approaches [4,5,6,7,8]. Nevertheless, no technique is as yet commonly in use in clinical practice.

PCG Obtaining

The phonocardiogram is obtained either with a chest microphone or with a miniature sensor in the tip of a small tubular instrument that is introduced via the blood vessels into one of the heart chambers. [9]

2.3 QT Interval

Prolongation of the QT interval is associated with early afterdepolarizations. Early afterdepolarizations of sufficient amplitude can generate premature action potentials that lead to cardiac arrhythmias, which can progress to ventricular fibrillation and sudden cardiac death. By contrast, shortening of the QT interval is associated with exaggerated heterogeneity of repolarization in time and space. This exaggerated heterogeneity of the action potential duration creates a substrate for functional reentry similar to that of the long QT syndrome but with hastened recovery and reduced refractoriness in the ventricle. Arrhythmias are more likely to be malignant in short, compared with long, QT syndromes. [10]

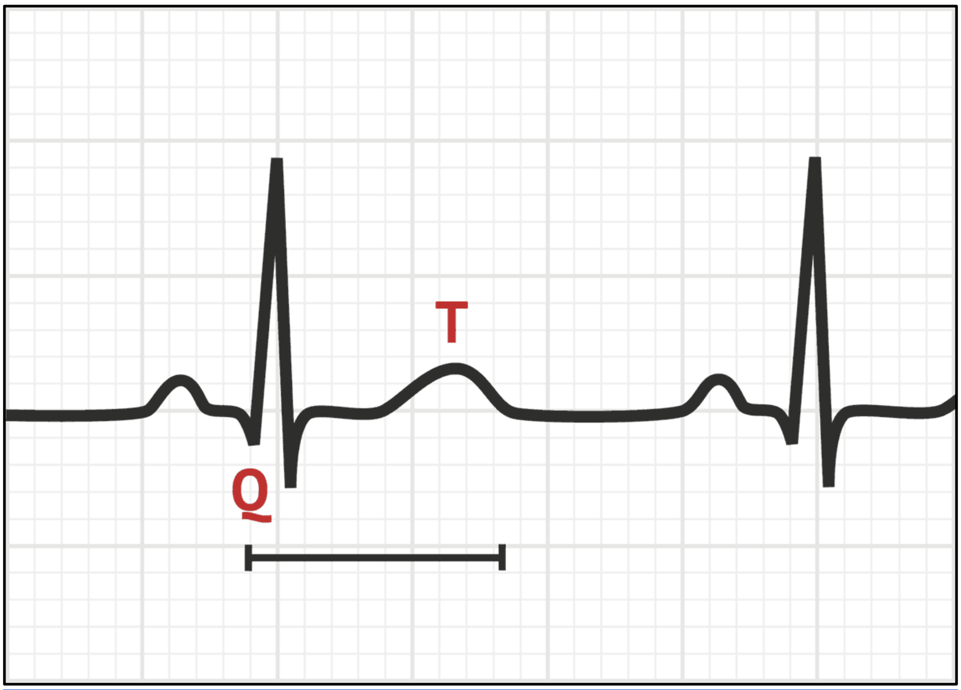

Figure 3 – QT Interval

QT interval has clinical significance and is widely used in pharmaceutical studies and clinical practice. For example, it is affected by various clinical conditions, including electrolyte abnormalities, antiarrhythmic drug effects, myocardial ischemia or infarction with deep T wave inversion, bradyarrhythmias, hypothermia, myocarditis, etc. [11]

QTC

The QT interval varies with the heart rate (HR) as is well known: normally the faster the HR (or the shorter the RR interval), the shorter the QT interval, and vice versa. For this reason, numerous clinical investigators have attempted to “correct” the QT interval to a value QTc which might be expected if the HR had been 60 beats per minute (bpm). [11]

The corrected QT interval has clinical significance and is widely used in pharmaceutical studies and clinical practice. For example, it is affected by various clinical conditions, including electrolyte abnormalities, antiarrhythmic drug effects, myocardial ischemia or infarction with deep T wave inversion, bradyarrhythmias, hypothermia, myocarditis, etc. [11]

There are numerous QT correction formulae that have been developed for research studies and clinical assessment with a variety of approaches being used in view of the complex relationship between QT interval and HR (or RR interval). Bazett’s formula is still the most popular in clinical practice, research, and education despite Simonson’s warning in 1961 and many others since then that it should not be recommended. Other approaches used to adjust QT for HR, like the Fridericia, Framingham, and Hodges formulae, are also popular in the field. [11]

There is increasing research interest in QT correction for each individual’s ECG. This approach is impractical for routine clinical ECG recording and is not considered further in this study. The hypothesis, therefore, is that the relationship between QT and HR can be mathematically or statistically described by a formula and applied to the general population. This assumption does not exclude the formula being HR and gender dependent. [11]

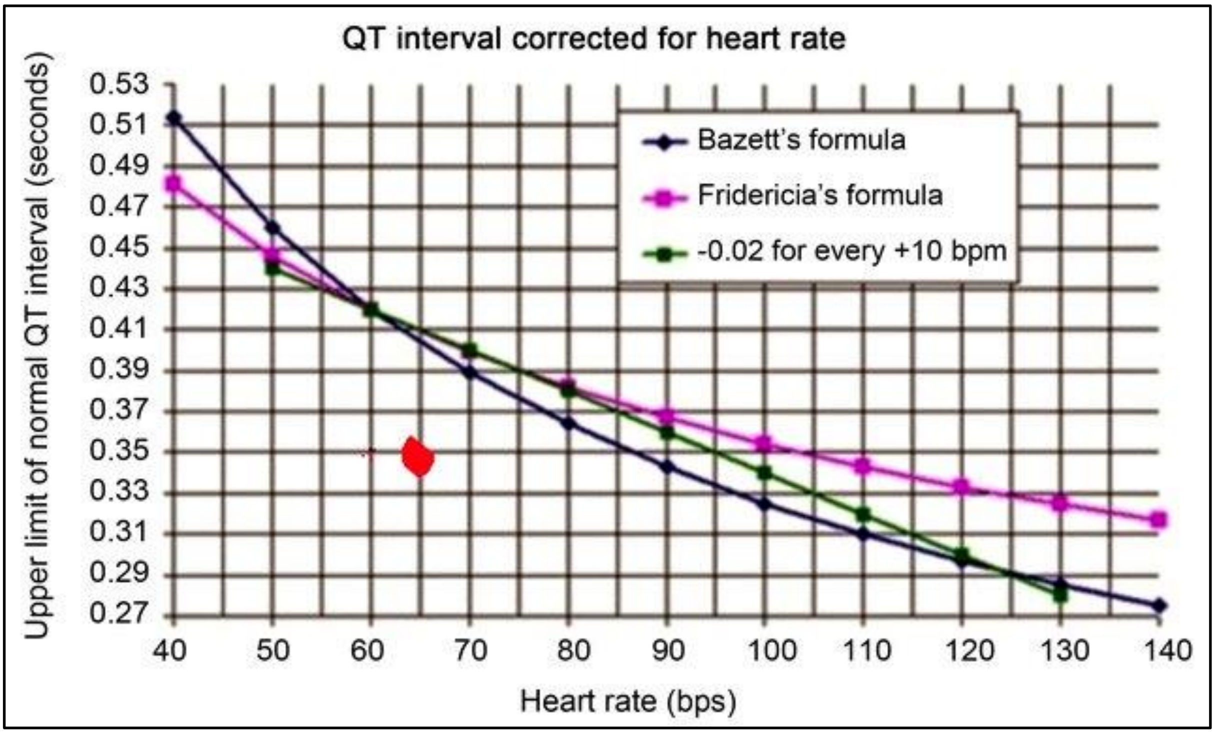

Figure 4 – Approximation Formulas

Normal values for the QTc range from 350 to 450 ms for adult men and from 360 to 460 ms for adult women; however, 10%-20% of otherwise healthy persons may have QTc values outside this range. Marked prolongations in the QT interval may be caused by genetic disorders (e.g., long QT syndrome), pharmacological agents (e.g., antiarrhythmics, antipsychotics, and antibiotics), electrolyte abnormalities (e.g., hypokalemia and hypomagnesemia), and their interactions. Other factors associated with QT interval length variability include age, sex, hypertension, body mass index, medication usage, low-calorie diets, serum potassium levels, and common genetic variants. Finally, within person variability and measurement error are additional sources of variability in the QT interval length. Bazett's correction is still recommended for the diagnosis of congenital long and short QT syndrome. [10]

2.4 Comparison of PCG and ECG data acquisition

In healthy conditions, the electrical activity of the heart drives its mechanical activity that directly causes the valves closure. [12] Thus, ECG and PCG are strictly related, as it can be easily observed by comparing two simultaneously acquired ECG and PCG recordings.

Obviously, both recordings contain the same number of cardiac cycles, since they are different representations of the same cardiac activity. Additionally, QRS complexes appear to be aligned with S1 occurrences and T waves offset appear to be close in time to S2 onset (a previous study of ours indicates that they are, on average, 5ms apart). [13]

Despite electrocardiography and phonocardiography being both noninvasive methods for cardiac monitoring, to acquire a PCG recording is usually much easier than to acquire an ECG recording, and PCG features identification is less sensitive to noise corruption. These characteristics make PCG recordings more reliable than ECG recordings, especially when obtained using wearable sensors that typically provide recordings affected by high levels of noise. On the other hand, ECG features, such the popular QT interval, are much more used in clinics than PCG features. [14]

2.5 T wave acquisition from Phonocardiogram

T-wave offset (Toff) identification is one of the most important challenges in ECG analysis: Toff is fundamental to compute the major features for diagnosis, but suffers from lack of reliability due to the presence of noise.

Thus, indirect measures of Toff are desirable. By observing simultaneous PCG and ECG recordings, it can be noted that Toff and S2 onset (S2on) seem to have the same time location in cardiac cycle. Previous studies used this property for evaluating the QT/QS2 index, defined as the ratio between QT interval (time distances between beginning of QRS complex and end of T wave) and QS2 interval (time distances between beginning of QRS complex and first major vibration of aortic component of S2, respectively). [14]

2.6 QT interval acquisition from Phonocardiogram

The previous to this project study [14] had as a goal a similar goal to predict the QT interval, using Phonocardiogram features. Further, is cited the article, describing the methodology implemented in this study. The results of this Master Thesis will be compared to the results obtained in this study.

Median ECG and median PCG cardiac beats were computed from each pair of simultaneously acquired signals. The electrocardiographic R-peak positions were taken as reference and used to compute the number N of cardiac beats included in each signal pair and the mean cardiac beat length (L; ms). Both cardiac signals were segmented into N cardiac beats; each beat was identified as the signal segment included between 250ms before the R-peak position and L-250ms after the same R-peak position. All cardiac beats segmented on the ECG were used to compute the median ECG beat (MECGB); analogously, all cardiac beats segmented in the PCG were used to compute the median PCG beat (MPCGB). [14]

Normality of feature distributions was evaluated by Lilliefors test; non-normal distributions were reported in terms of 50th (median value) [25th; 75th] percentiles. In order to obtain a PCG-based QT estimation, S1on, S2on and L were considered as inputs of the regression model.

QT' = p1·S1on +p2·S2on +p3·L +p4·S1on·S2on +p5, where p1, p2 and p3 are dimensionless, p4 is in ms-1 and p5 is in ms.

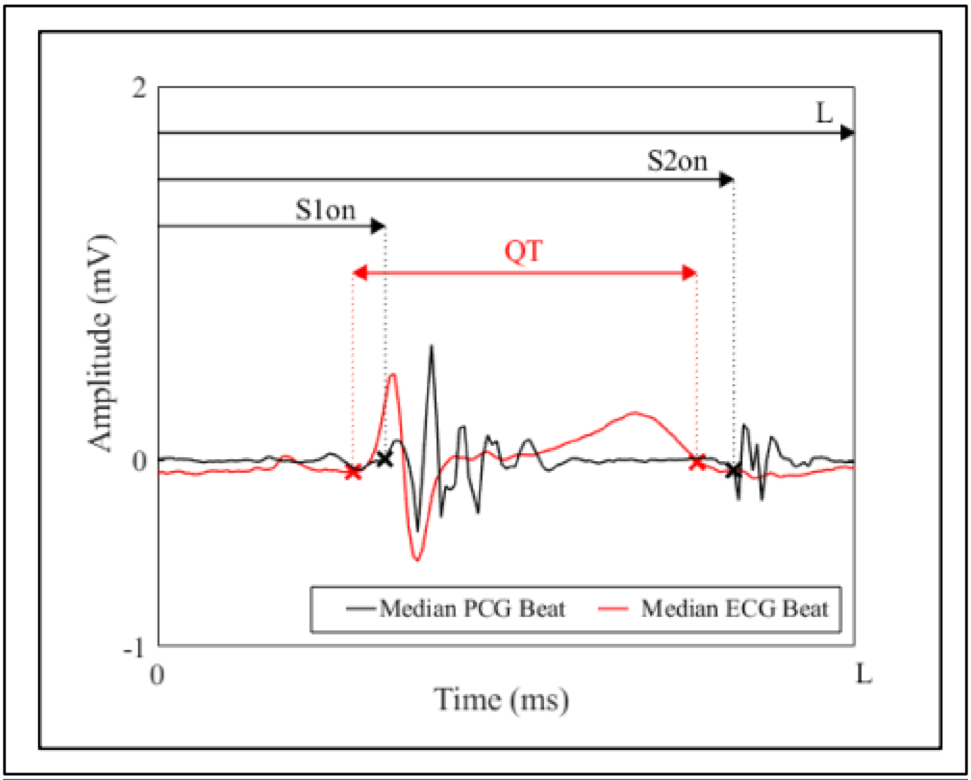

Figure 5 – QT plot in the previous study

The model was formulated by performing a regression analysis between QT and reference QT interval, in order to identify the numerical values of the five regression coefficients p1 to p5. Moreover, the model was validated by using the leave-one-out cross-validation algorithm. For both model formulation and validation, QT values were compared using the Wilcoxon rank sum test for equal medians. [14]

The absolute error (ε; ms) and the relative percentage error (ε%) were computed. Finally, the Pearson's correlation analysis was performed by computing the correlation coefficient (ρ) and regression line (QT=m·QT' + q) between predicted QT and reference QT.

The validation on the article lead to the next graphical result:

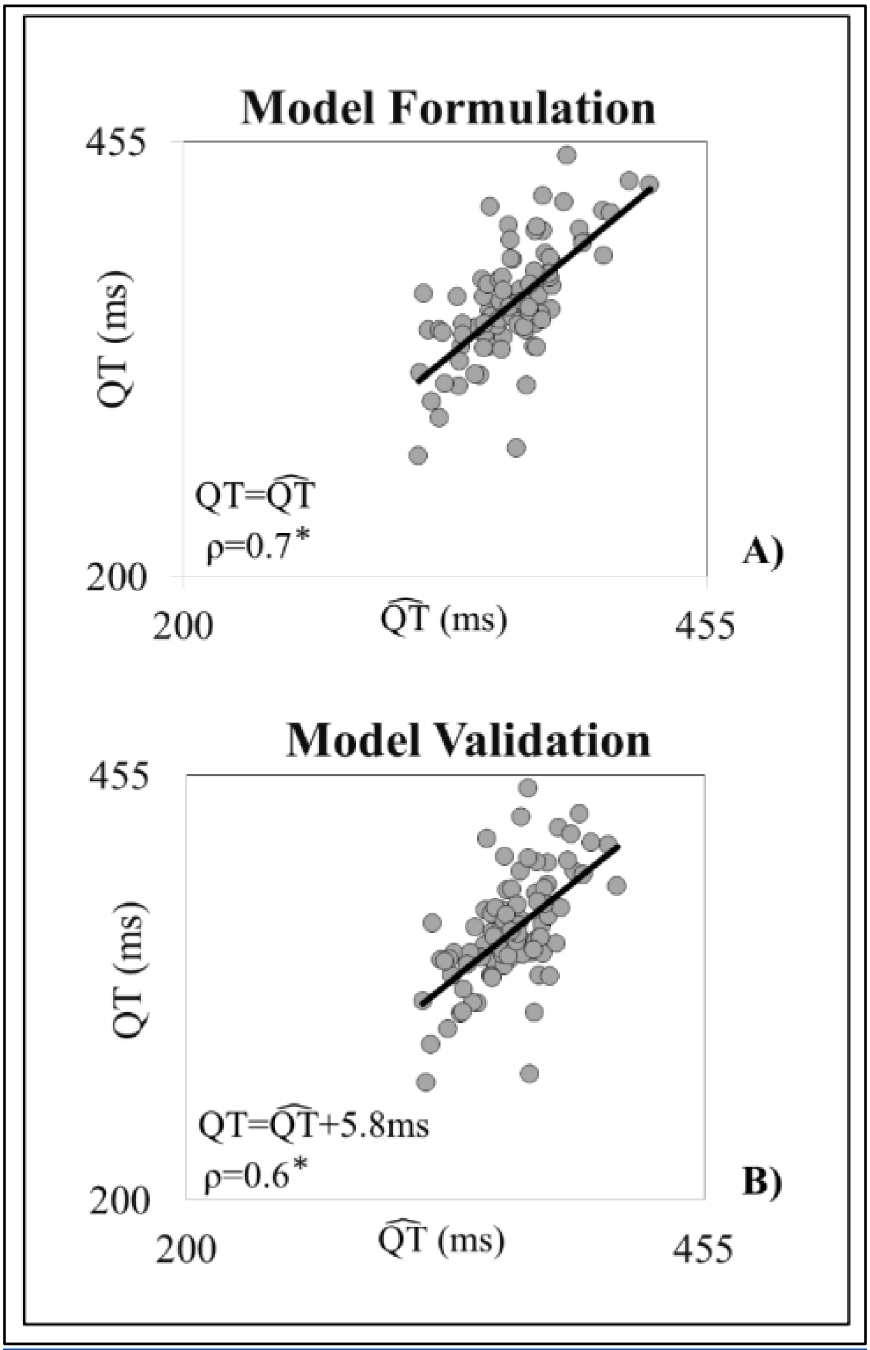

Figure 6 – Model Validation previous study

The obtained plot and other results from this study will be finally compared with the results obtained in this Master Thesis. The goal consists in achieving similar results or improving the results of the study and make a QT prediction from the PCG recording.

2.6 Introduction to neural networks

The neural networks will be used in the last step of the processing of this project, so a small introduction to this topic is given in this chapter.

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously. Thus, artificial neural networks attempt to solve complicated problems, like summarizing documents or recognizing faces, with greater accuracy.

Neural networks can help computers make intelligent decisions with limited human assistance. This is because they can learn and model the relationships between input and output data that are nonlinear and complex. [15]

How do neural networks work?

The human brain is the inspiration behind neural network architecture. Human brain cells, called neurons, form a complex, highly interconnected network and send electrical signals to each other to help humans process information. Similarly, an artificial neural network is made of artificial neurons that work together to solve a problem. Artificial neurons are software modules, called nodes, and artificial neural networks are software programs or algorithms that, at their core, use computing systems to solve mathematical calculations.

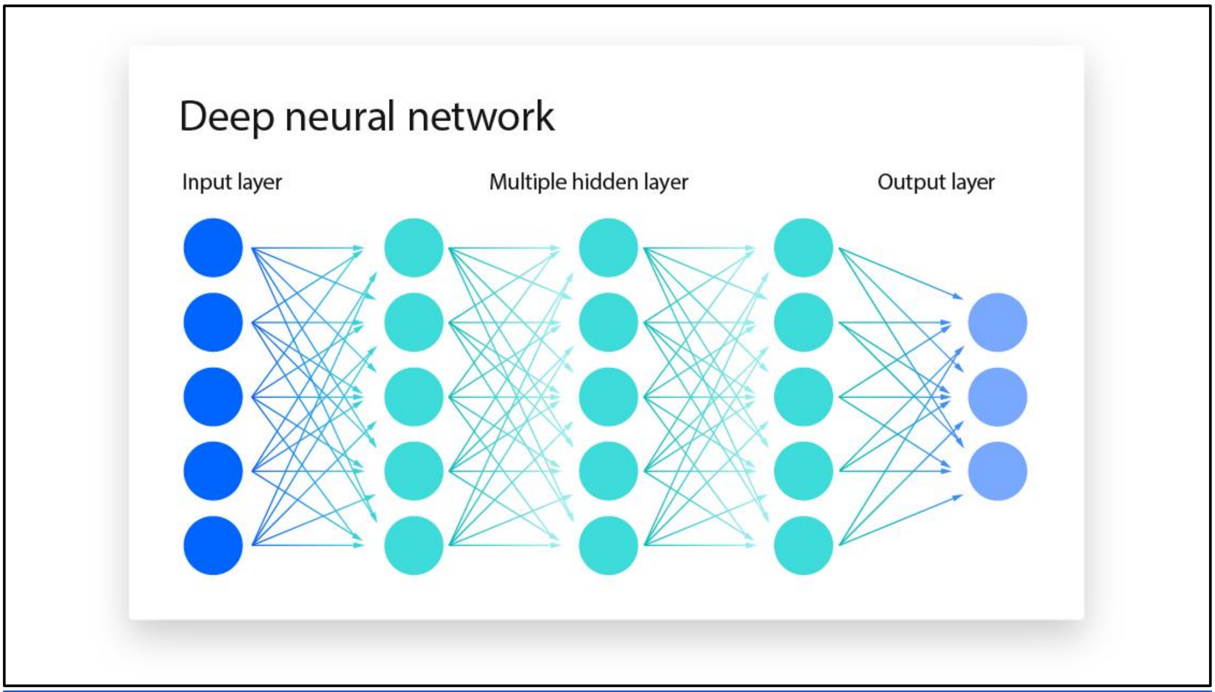

Figure 7 – Neural Networks

A basic neural network has interconnected artificial neurons in three layers:

- Input Layer

Information from the outside world enters the artificial neural network from the input layer. Input nodes process the data, analyze or categorize it, and pass it on to the next layer.

- Hidden Layer

Hidden layers take their input from the input layer or other hidden layers. Artificial neural networks can have a large number of hidden layers. Each hidden layer analyzes the output from the previous layer, processes it further, and passes it on to the next layer.

- Output Layer

The output layer gives the final result of all the data processing by the artificial neural network. It can have single or multiple nodes. For instance, if we have a binary (yes/no) classification problem, the output layer will have one output node, which will give the result as 1 or 0. However, if we have a multi-class classification problem, the output layer might consist of more than one output node. [15]

How to train neural networks?

Neural network training is the process of teaching a neural network to perform a task. Neural networks learn by initially processing several large sets of labeled or unlabeled data. By using these examples, they can then process unknown inputs more accurately.

- Supervised learning

In supervised learning, data scientists give artificial neural networks labeled datasets that provide the right answer in advance. For example, a deep learning network training in facial recognition initially processes hundreds of thousands of images of human faces, with various terms related to ethnic origin, country, or emotion describing each image.

The neural network slowly builds knowledge from these datasets, which provide the right answer in advance. After the network has been trained, it starts making guesses about the ethnic origin or emotion of a new image of a human face that it has never processed before. [15]

3. MATERIALS AND METHODS

3.1 Database used

Data considered in this study belong to the training set A of PhysioNet/CinC Challenge 2016 database, consisting in 409 pairs of short (around 30s) PCG and ECG signal. Both signals were simultaneously recorded by the Welch Allyn Meditron electronic stethoscope. ECGs sampling frequency was 1000 Hz, while PCG sampling frequency was 2000 Hz. [16]

This way, using this dataset we will be able to extract features from PCG and ECG simultaneously. Finally, after the processing of the signals, the correlated dataset can be obtained, which can be used to train the prediction model.

Also, to work with this database the “The WFDB Toolbox for MATLAB and Octave” has been used. The WFDB Toolbox is a collection of functions for reading, writing, and processing physiologic signals and time series in the formats used by PhysioBank databases (among others). The Toolbox is compatible with 64-bit MATLAB and GNU Octave on GNU/Linux, Mac OS X, and MS-Windows. [17]

3.2 Software used

MATLAB

MATLAB is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models.

MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook. [18]

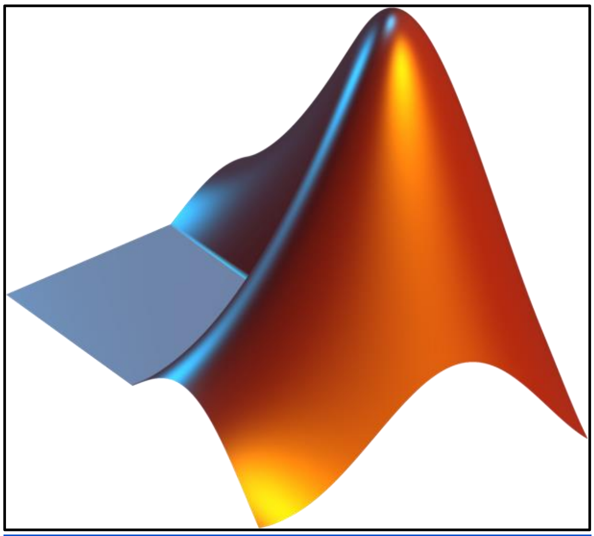

Figure 8 – Matlab

Biomedical signal processing involves acquiring and preprocessing physiological signals and extracting meaningful information to identify patterns and trends within the signals.

Sources of biomedical signals include neural activity, cardiac rhythm, muscle movement, and other physiological activities. Signals such as electrocardiogram (ECG), electroencephalogram (EEG), electromyography (EMG) can be captured non-invasively and used for diagnosis and as indicators of overall health. [18]

The biomedical signal processing workflow involves:

- Signal Acquisition
- Signal Visualization and Annotation
- Artifact Removal and Preprocessing
- Feature Extraction

MATLAB provides built-in apps to help you analyze and visualize signals in time, frequency, and time-frequency domains without writing any code. Signals can be labeled manually or using algorithms like those that find peaks and transition points. Biomedical signals often contain noise or unwanted artifacts that can distort the analysis of the signals. One of the main challenges in preprocessing biomedical signals is to remove unwanted artifacts while preserving the sharp features within signals. Most popular techniques for artifact removal are digital filtering, adaptive filtering, independent component analysis (ICA), and recursive least squares. A combination of preprocessing techniques may also be used to address the limitations of individual techniques. [18]

Feature extraction can be accomplished manually or automatically. Signal processing techniques like AR modeling, Fourier analysis, and spectral estimation can be used to manually compute key features from signals. Time-frequency transformations, such as the short-time Fourier transform (STFT) can be used as signal representations for training data in machine learning and deep learning models. Automatic feature extraction techniques like wavelet scattering can be used to reduce dimensionality and extract important features. These features can be used directly for diagnosis or as input to machine learning and deep learning classifiers. [18]

Python

Figure 9 – Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. [19]

Keras

Keras is a deep learning API written in Python and capable of running on top of either JAX, TensorFlow, or PyTorch.

Figure 10 – Keras

Keras is:

- Simple – but not simplistic. Keras reduces developer cognitive load to free you to focus on the parts of the problem that really matter.
- Flexible – Keras adopts the principle of progressive disclosure of complexity: simple workflows should be quick and easy, while arbitrarily advanced workflows should be possible via a clear path that builds upon what you've already learned.
- Powerful – Keras provides industry-strength performance and scalability: it is used by organizations including NASA, YouTube, or Waymo. [20]

4. PROPOSED SOLUTION

The proposed solution can be summarized with the next scheme:

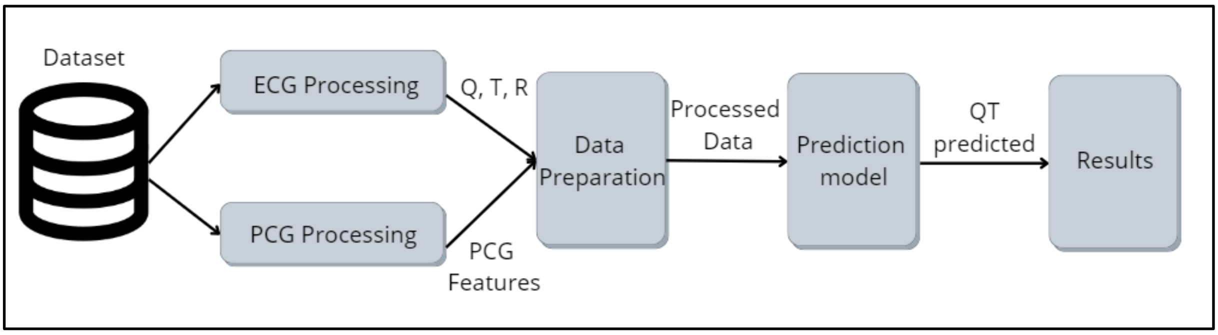

Figure 11 –Proposed Solution

We will extract the synchronously recorded ECG and PCG signals. Each signal will be processed, using the implemented algorithms for processing.

From the ECG processing the features as a start of Q-wave, end of T-wave and R-peak positions will be extracted and used to train the prediction model further.

In case of PCG processing, we are not limited in the features that could be obtained, so the maximum possible relevant information will be extracted from the PCG Processing step.

Once all the signals have been processed, the Data will be prepared as a Dataset of related Data with the information about each segment of the signal.

The prepared dataset will be used to feed a prediction model, that after training should be capable of predicting the QT interval from the PCG.

The results obtained will be discussed and compared to some previous results obtained in different studies and the conclusion of the effectiveness of this method will be deduced.

5. DATA ACQUISITION

This chapter describes the processing of the ECG and PCG signals and the extraction of the necessary features to predict the ECG features from PCG features in the future. Finally, the extracted segments are prepared and organized in a data frame, where each will contain the information related to one segment of heartbeat.

5.1 ECG processing

The full code of the Matlab algorithm that implements the functionality of ECG processing can be found in the annexes. As a code is large and sophisticated it was decided to exclude it out of the memory for the readability of the solution. Here is the detailed scheme of the code general blocks:

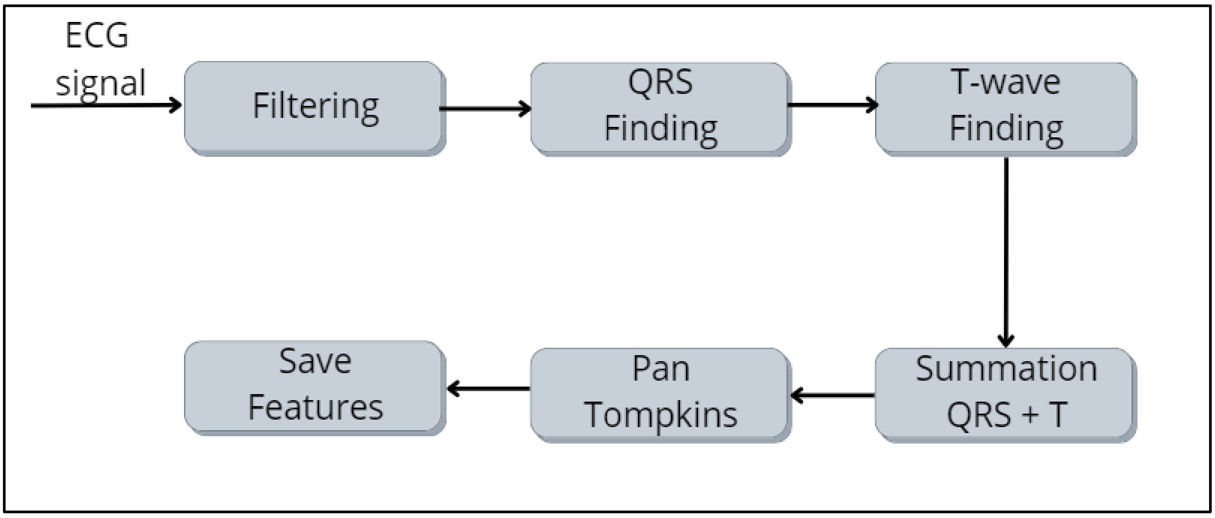

Figure 12 – ECG Processing

Inputs	Outputs
Input Signal ECG	Q
ECG signal polarity	T
T-wave polarity	-

Table 1 – Inputs and outputs of the ECG Processing

The ECG processing is a necessary step to obtain the data for the training and validation. The final goal consists in deleting the need of the step of ECG processing, as it is a more complicated medical procedure to obtain this type of signals.

In this step we don't need all the possible features from the ECG, as only the points of Q wave start and T wave finish will be needed. Also, the R-peak will be extracted, however it is not a primary target, but it could be used also for validation.

Filtering

In a first step the code discards the first and last 100 samples of the audio signal, as it is more probable for these samples to be noisy.

Then, a bandpass filter is applied to the signal using the "eegfilt" function. In this step 2 different signals will be made. One signal will be obtained with lower and upper cutoff frequencies of 7 and 70 Hz, respectively, and the second will be filtered with 7 and 20 Hz.

With these 2 obtained signals the further processing will be made.

QRS Finding

After applying the initial preconditioning of the signal, the next step consists in finding the QRS complex. The implemented algorithm for finding the energy correctly is based on the previous study: "ECG denoising algorithm based on group sparsity and singular spectrum analysis". [21] However, the modifications to the initially proposed algorithm will be done.

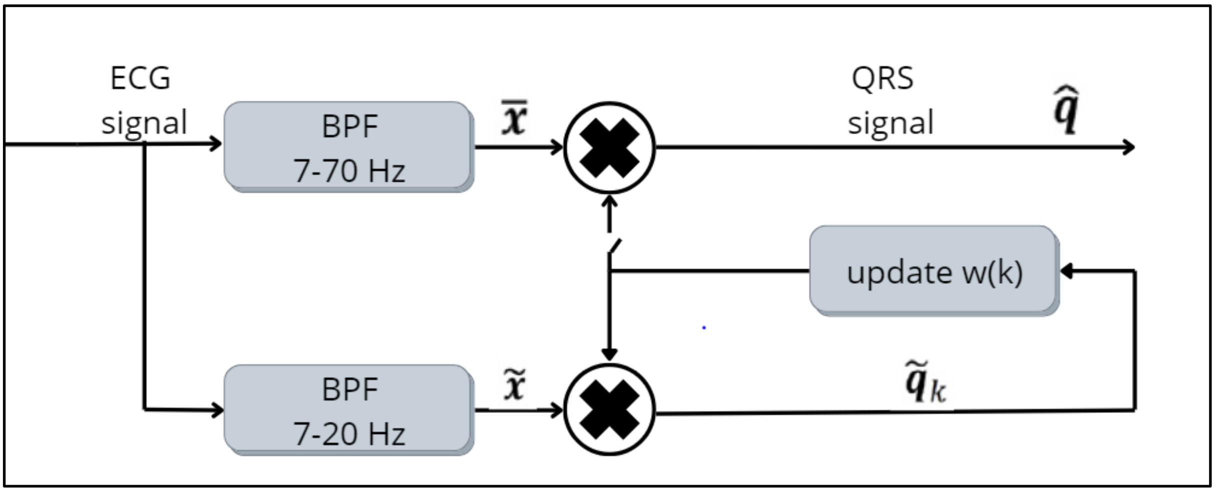

Figure 13 – QRS Finding

After filtering the signal by 2 bandpass filters the signal filtered between 7 and 20 Hz is multiplied by the signal w(k), that initially is a vector of ones, but then it is updated based on the algorithm that will be described.

After reaching a determined ratio, the signal resulting from the multiplication of the iterative part is multiplied by the signal filtered between 7 and 70 Hz and finally the resulting QRS signal is obtained, which ideally will only contain the denoised QRS complex and in the other parts will be flat:

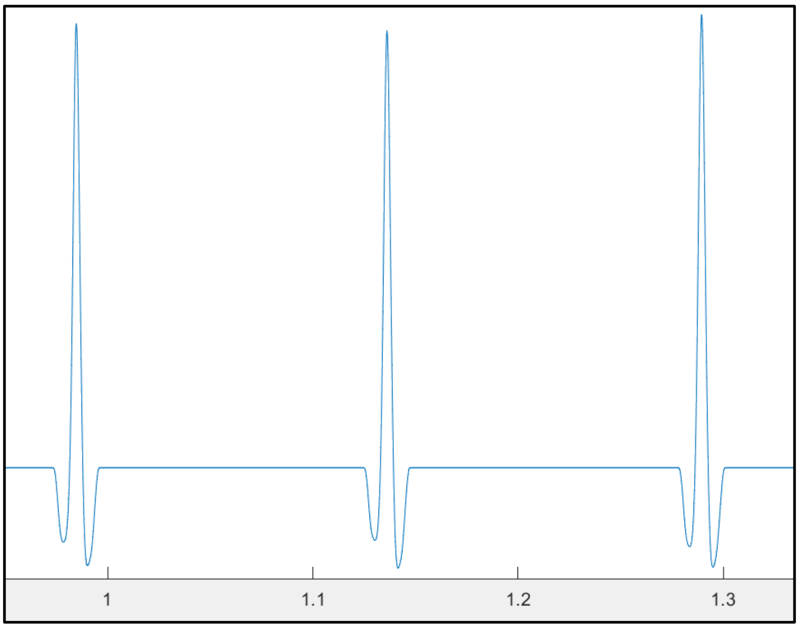

Figure 14 – QRS complex

Let's define the vector f = [f1. . . fT], where each t entry is defined as the energy of the group of samples of q(t) confined within a short rectangular window extending from *t-m* to *t+m* and centered at the t sample, that is:

$$f_t = \sum_{j=t_i}^{t_f} q_j^2 \qquad (1)$$

Normally, the size L_t the window equals (2m + 1) samples,where the value of m > 0 is to be determined. However, To cope with the samples at the beginning and the end of the analyzed signal, the window size is adjusted by neglecting its portion that does not overlap with the signal. Therefore, t_i= max(1, t − m), t_f= min(t + m, T), and, hence, the window size L_t has the following bounds (m + 1) ≤ Lt≤ (2m + 1). To determine the value of m, it is obvious that m should be selected based on the size of the QRS-complex. Since the duration of normal QRS–complex approximately equals 100 ms, it is recommended to set m = 50 ms.

Than the algorithm as suggested by the cited paper [] is the next one:

First, we initialize the values:

$$\widehat{q_0} = \hat{x}$$

$$\widehat{\alpha_0} = 1$$

While the needed ratio is not reached or the maximum number of retries in not exceeded we implement the next algorithm considering k as iterated value:

1. $f_t = \sum_{j=t_i}^{t_f} q_j^2$ (2)

2. $\sigma_t = \sum_{j=t_i}^{t_f} \frac{1}{f_j + \alpha}$ (3)

3. Resulting σ is sorted in descending order.

4. It is constructed a line from the maximum value of sorted vector to the minimum:

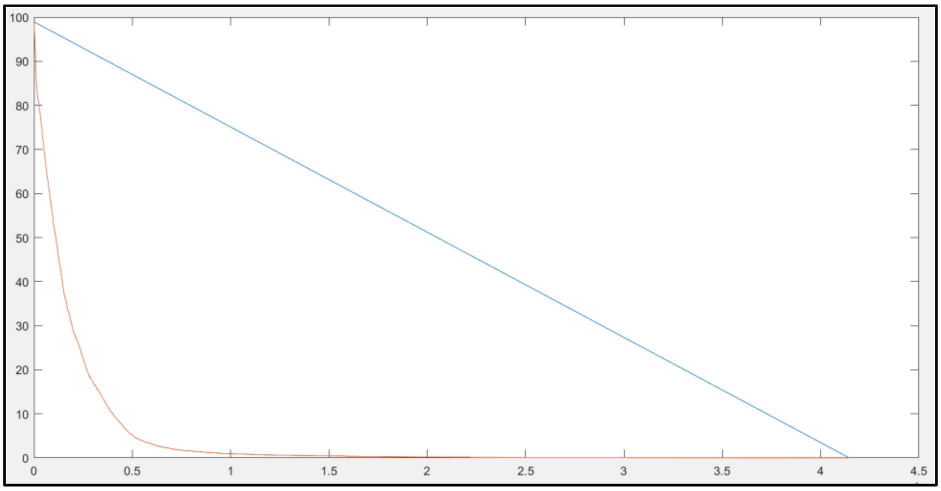

Figure 15 – Sorted Vector

5. The distances from the sorted vector to the line are calculated in each position and the index of the maximum distance is obtained and λ becomes a value of the sorted sigma in this index.

6. $w = \frac{1}{(1+\lambda)} * \sigma$ (4)

7. Smooth out the sharp transitions in w.

8. $\widehat{q_{k+1}} = w * \hat{x}$ (5)

9. if $||\widehat{q_{k+1}} - \widehat{q_k}||_2 / ||\widehat{q_k}||_2 > 0.3$ break. (6)

10. α is updated as the maximum between 0.0001 and $\alpha/2$

After that process the values of the resulting omega signals are set to exactly 0 when they don't correspond to the QRS complex interval.

Finally, we get a vector w that could be represented like this and it specifies the amount of energy obtained from the previous algorithm and determines the location of a QRS complex:

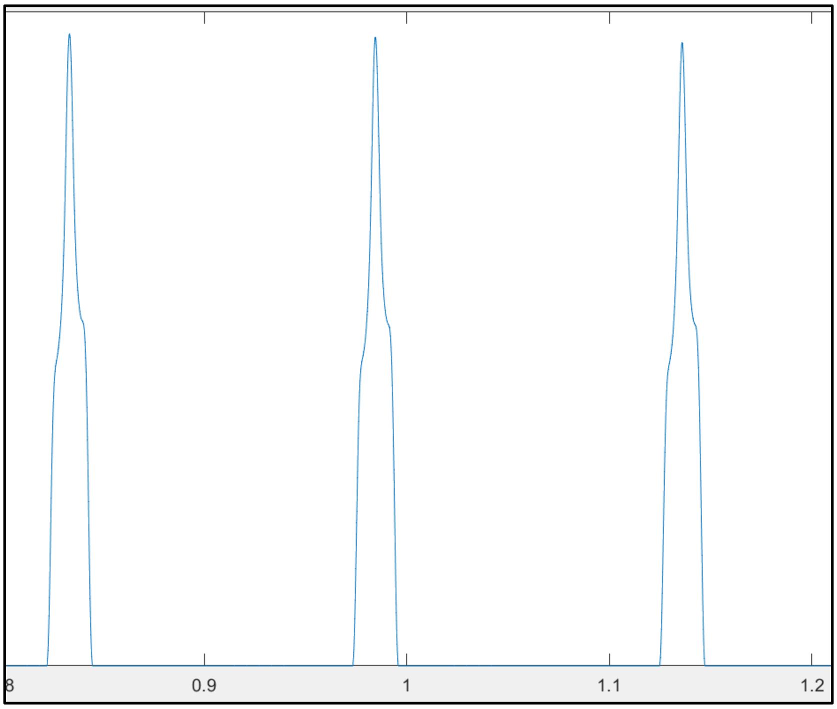

Figure 16 – Omega vector

Then this resulting signal is multiplied by the signal filtered between 7 and 70 Hz, finally obtaining the QRS complex.

T-wave Finding

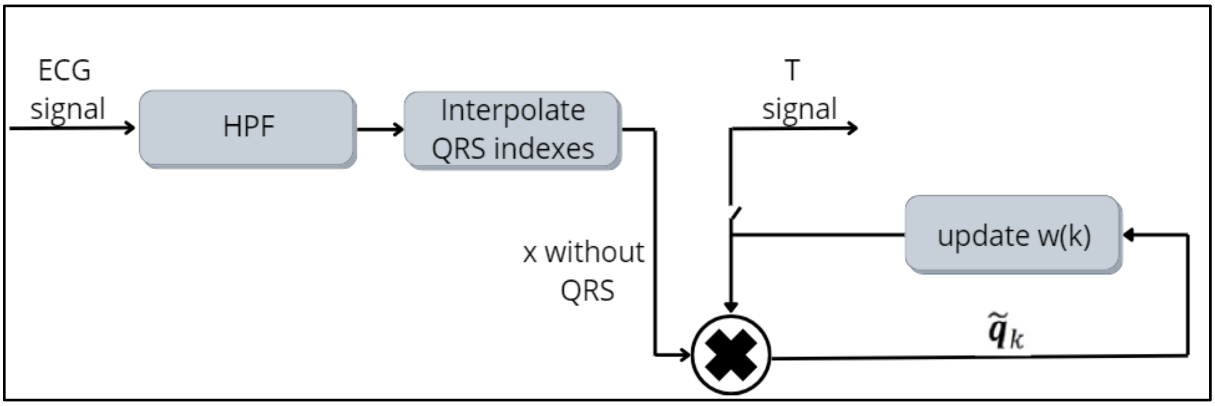

Figure 17 – T-wave Finding

To obtain the signal corresponding to the T-wave first the initial ECG signal is filtered, using a high-pass filter with butter function in Matlab of order 4 and a high-pass frequency as a half of the sampling frequency.

Than the indexes corresponding to the QRS complex are interpolated resulting in the ECG signal without QRS complex:

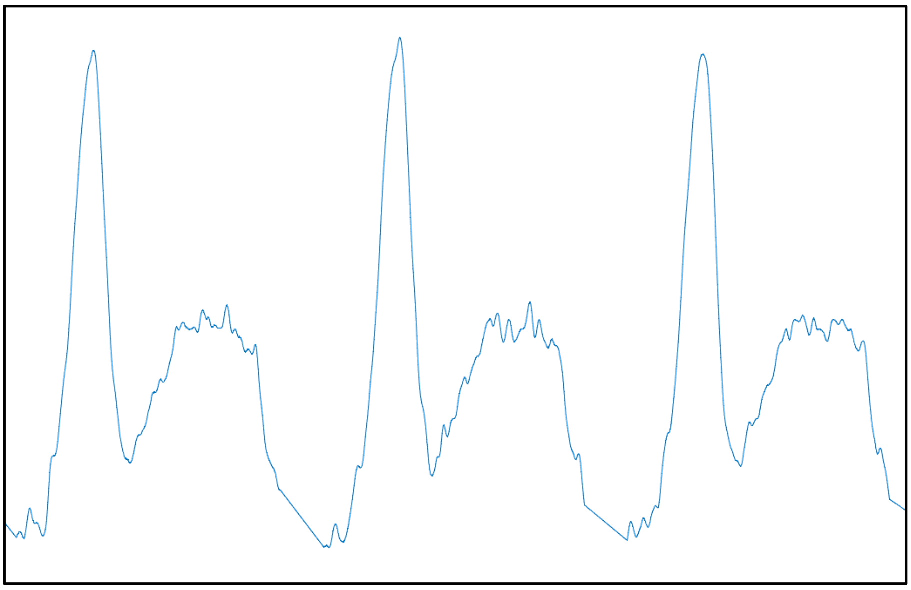

Figure 18 – Signal without QRS

After that, it is implemented a similar algorithm for the energy searching as in the previous step. But in this case, the algorithm is simpler, as the energy found can be directly used as the T-signal, as in this case the T-wave is one-directional, meaning that it only can be fully negative or fully positive.

The algorithm described for the QRS complex obtaining is executed with the next modifications:

- m (size of the window) is incremented to 70
- The step 9: if $\left|\left|\widehat{q_{k+1}} - \widehat{q_k}\right|\right|_2 / \left|\left|\widehat{q_k}\right|\right|_2 > 0.8$ is modified to the threshold of 0.8.
- The resulting signal is directly the amount of energy found with the algorithm, deleting the noise from the original signal.

As a result, a processed signal for the T-wave is obtained with clear T-wave peaks:

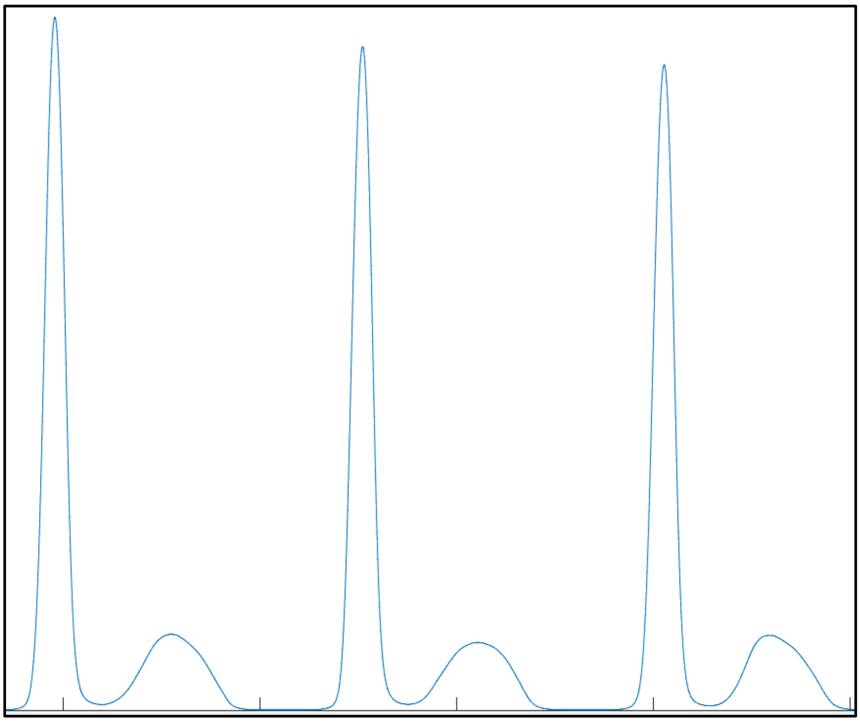

Figure 19 – T-wave signal

Summation QRS + T

Once the signals for the QRS complex and for the T-wave are obtained a simple sum is performed. And after that the values between the end of T-wave and the start of Q-wave are interpolated to make more reliable a further processing with the Pan Tompkins algorithm, deleting the P-wave energy and other noise:

After running the described, finally a clear ECG can be obtained:

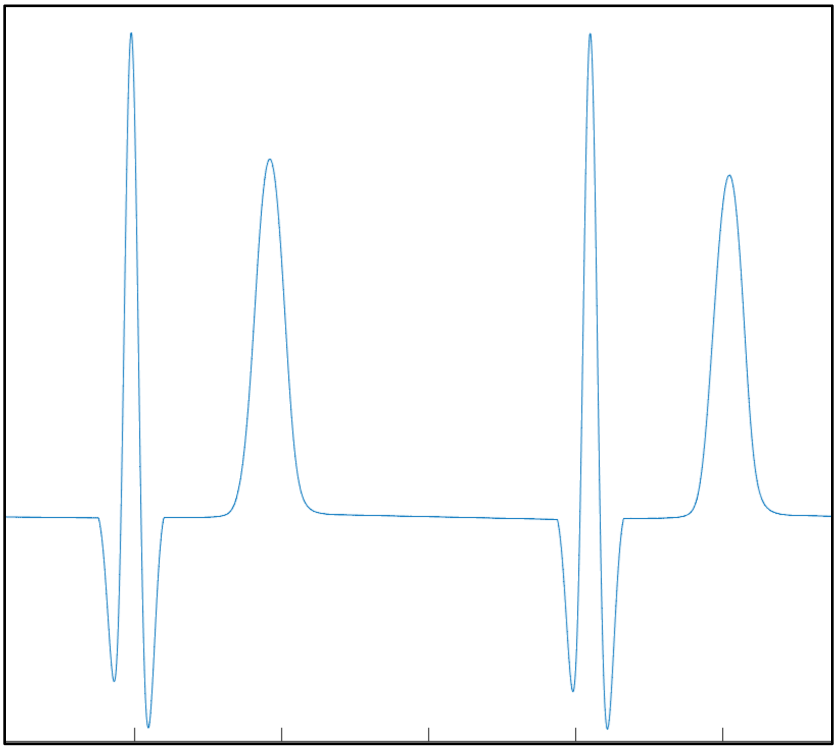

Figure 20 – Processed signal

Pan Tompkins

Once the processed signal is obtained the Pan Tompkins algorithm is executed, using Matlab function “ecgpuwave”.

ECGPUWAVE analyzes an ECG signal from the specified record 'recordName', detecting the QRS complexes and locating the beginning, peak, and end of the P, QRS, and ST-T waveforms. The output of “ecgpuwave” is written as a standard WFDB-format annotation file associated with the specified annotator 'outputAnnFileName'.

Once the algorithm is executed, it is possible to plot the resulting signal with the annotations made by the algorithm:

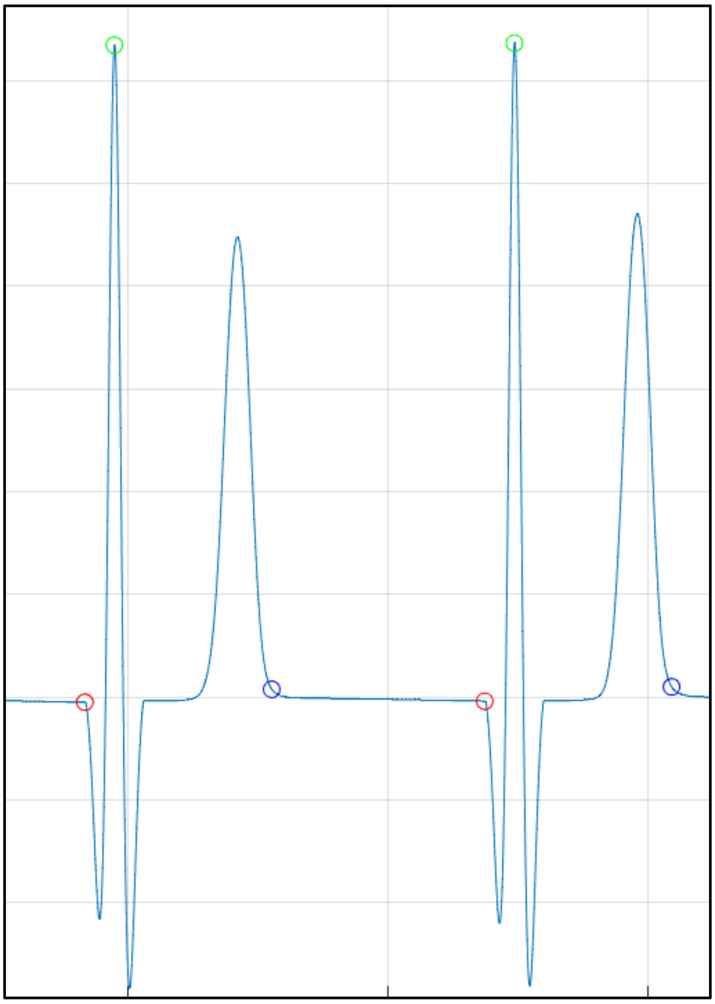

Figure 21 – Pan Tompkins annotations

Save Features

Once all the locations of Q, R and T waves are sound the result is stored and associated for each signal the processing has been performed.

5.2 PCG processing

Being the PCG signal non-stationary and typically affected by a wide range of artifacts, its processing and interpretation is not a trivial issue. A critical step of all PCG signal-processing tools is the segmentation of heart sounds, i.e., the partitioning of the signal into segments corresponding to S1, systole, S2 and diastole intervals. This task requires identifying on the signal the time of occurrence and duration of the main heart sounds, as well as their classification. [22]

A variety of segmentation techniques have been extensively discussed in the literature, encompassing envelope-based, wavelet-based, and machine learning approaches, as well as methods grounded in time-frequency analysis and hybrids of these methodologies [5,6,7,8,9].

However, as of now, no single technique has gained widespread adoption in clinical practice. The precise and consistent identification of the onset, peak, and cessation points of heart sounds remains a formidable challenge from a technical standpoint. Factors such as the low amplitude of the signal, the presence of diverse artifacts, and the inherent non-stationarity of the signal all contribute to the complexity of the task.

The full code of the Matlab algorithm that implements the functionality of PCG processing can be found in the annexes. As a code is large and sophisticated it was decided to exclude it out of the memory for the readability of the solution. Here is the detailed scheme of the code general blocks:

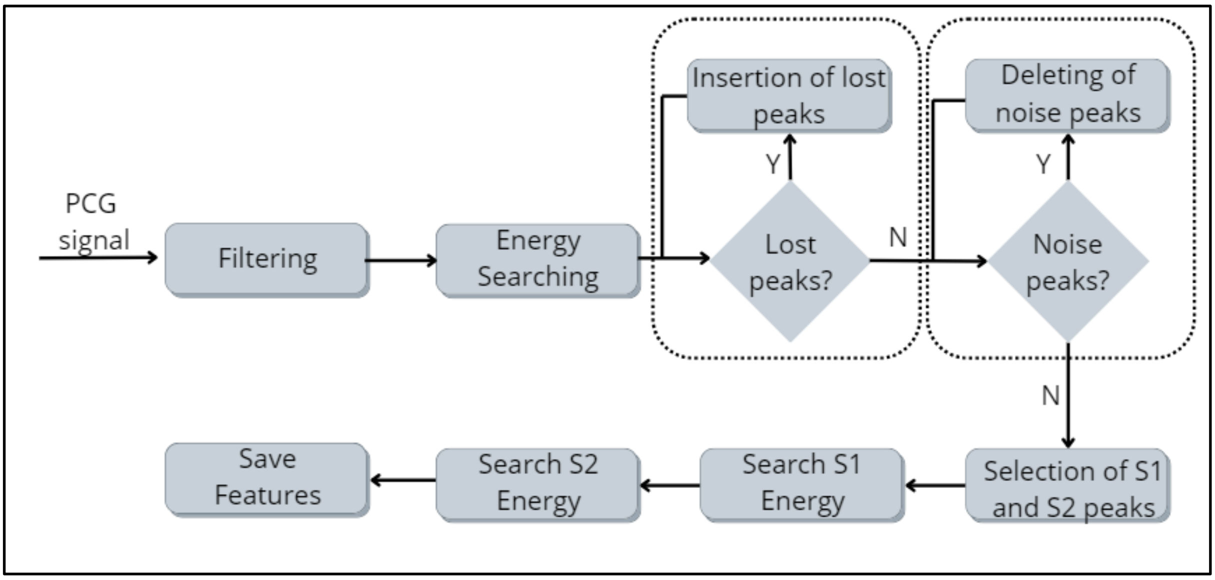

Figure 22 – PCG Processing

Inputs	Outputs
Input Signal PCG	Indexes start S1
manual indexes set to 0	Indexes end S1
	Indexes start S2
	Indexes end S2
	Energy vector S1
	Energy vector S2

Table 2 – Inputs and outputs of the ECG Processing

The PCG processing constitutes a pivotal component of our proposed solution, given its paramount role in detecting the QT or QTC interval. Achieving precision and adaptability in signal processing is crucial, particularly when dealing with diverse signal types from patients with conditions such as arrhythmia and other diseases.

Our approach emphasizes extracting maximum and accurate information from PCG data. This rich dataset serves as a foundation for training a prediction model, with the ultimate goal of reliably predicting the QT interval in future applications. The adaptability of our solution is paramount, catering to various signal characteristics while excluding extremely noisy signals.

Recognizing the inherent variability in signals, our solution is designed to be flexible, offering reliable outcomes across a spectrum of inputs. While maintaining this flexibility, some manual adjustments may be applied to enhance processing efficiency in cases where the code encounters challenges. This approach ensures that, even when faced with diverse signal profiles, our solution can be fine-tuned to yield optimal results.

Extensive testing across a diverse dataset has demonstrated the efficacy of the code, resulting in a commendable success rate. However, acknowledging the potential for further refinement, we aspire to enhance the automated processing of PCG, aiming to minimize or eliminate the necessity for manual adjustments in handling specific signals. This ongoing improvement process is geared towards achieving even higher levels of accuracy and automation in PCG analysis.

In order to train the model we will be searching for some specific points and values, the most relevant data for the future comparison will be S1 and S2 intervals and all the information we can get from them, for example, the length, the periodicity, start and finish position, the location of the maximum energy peak. In more detail the information that will be used from the processed PCG will be explained in a Data Preparation section.

Filtering

The first block of code implements the next steps:

The code discards the first and last 100 samples of the audio signal, as it is more probable for these samples to be noisy.

The code enables the possibility of some manual adjustments of the signal in order to improve the final result of the code, for example it is possible to provide an array of indexes, where the signal is set to 0, if it complicated in some cases to distinguish automatically, using processing techniques the actual S1 and S2 intervals from the noisy intervals.

Apart from the zero setting, we can multiply or divide some specific intervals by a constant as a part of manual adjustment.

In order to have more similar signals in terms of amplitude, we remove the Mean from each signal, obtaining a normalized audio signal.

Then, a bandpass filter is applied to the signal using the “eegfilt” function with lower and upper cutoff frequencies of 20 and 200 Hz, respectively, as it is suggested in the literature for the PCG processing.

In summary, the code prepares the audio signal by removing unnecessary portions, zeroing specific indexes, adjusting amplitudes, and applying a bandpass filter for subsequent analysis.

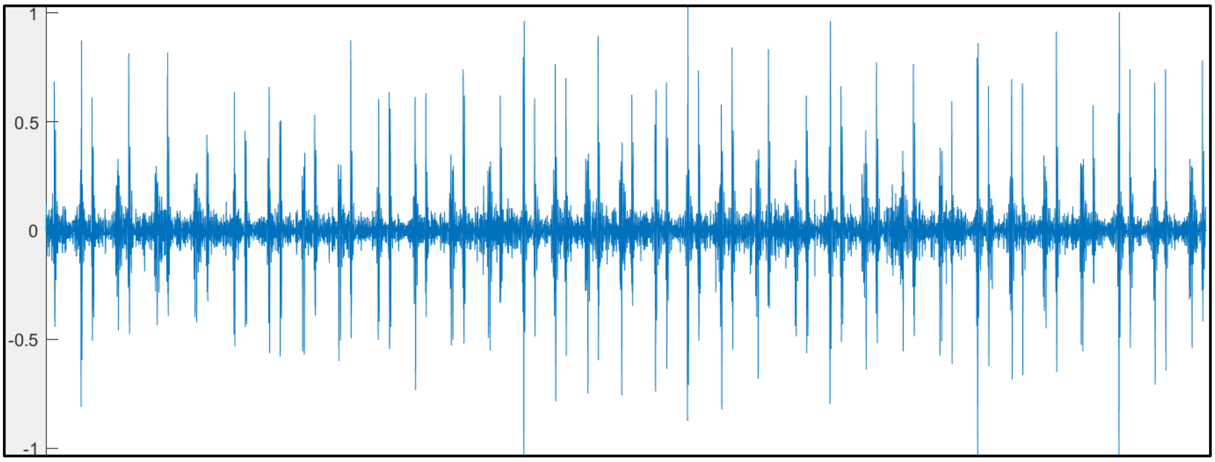

Figure 23 – Filtered PCG

Generic Function

The mechanism of searching the energy will be based on the algorithm implemented also for the ECG. However, through the code this algorithm will be parametrized. As the input to the algorithm several parameters can be modified, such as the length of the window, the ratio threshold, the threshold of found energy that should be saved or discarded.

This function, named “energy_finding”, computes the energy vector omega from the input signal using a series of energy-related calculations. The detailed explanation of the algorithm was made in the ECG processing chapter.

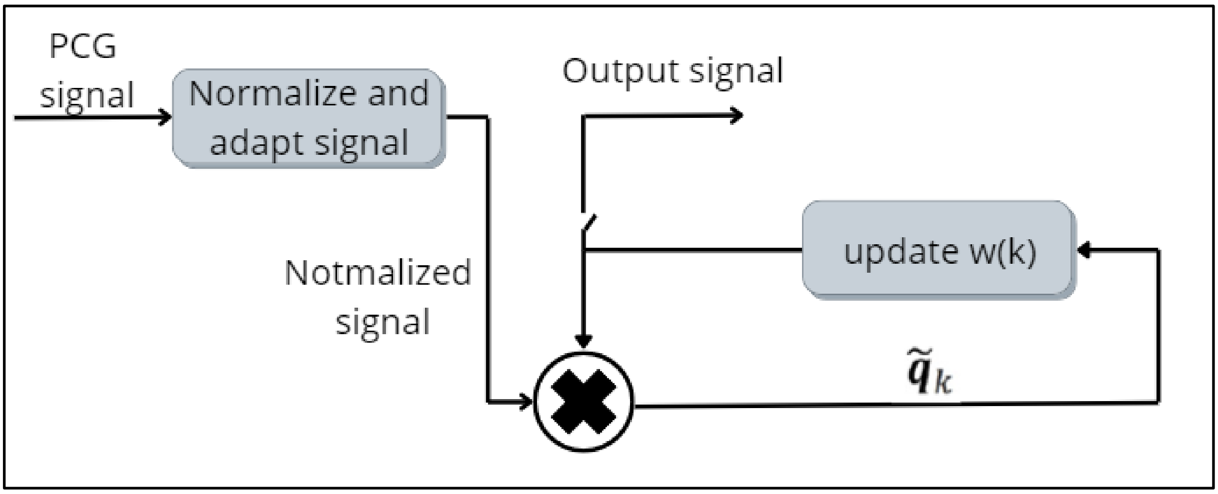

Figure 24 – Generic function

Inputs and outputs of this function are presented in the table.

Inputs	Outputs
Input Signal	Energy calculated for the input signal
Size of the window	
Divisor to divide the input signal by	
Threshold to break the for loop	
Threshold of the start and end of the peak	
Threshold of energy value to reach to save the window	

Table 3 – Inputs and outputs for the energy calculation

The function can be described by the next steps:

- Iterate through the first stage calculations up to a predefined maximum number of iterations.
- Update matrices f, sigma, lambda, w, and q based on the input signal and intermediate results as described in the QRS finding.
- Adjust parameters and calculate intermediate results to converge toward the desired energy vector.

Second Stage Iterative Calculation:

- The function includes a loop for the second stage, which iteratively adjusts the weights and calculates the energy vector.
- It dynamically adjusts parameters based on the calculated values and convergence criteria.
- If a certain ratio condition is met, the loop breaks.

Post-Processing:

- Normalize the resulting weight vector (w) and apply additional filtering using filtfilt.

Thresholding and Windowing:

- The function applies thresholding to the resulting energy vector and sets values below the threshold to zero.
- Additional processing involves determining and preserving specific windows based on threshold criteria.
- The resulting energy vector is saved in the variable omega.

Output:

- The function returns the final energy vector omega.

Energy Finding

This code segment focuses on computing the energy distribution of a signal and identifying peaks in this distribution.

Initialization:

- result_of_code is set to 0.
- tries is initialized to 0.

Energy Distribution Calculation:

- A while loop runs as long as result_of_code is 0 and tries is less than 1.
- The loop iterates at most three times, adjusting “ratio_threshold” and “save_window_threshold” on the second and third tries.
- The energy distribution is calculated using the “energy_finding” function, applied to the absolute values of the filtered signal.

Peak Detection:

- The maximum value of the energy distribution is determined.
- Peaks in the energy distribution are identified using the “findpeaks” function, with specific constraints on peak distance, height, and prominence.

Ideally this code already finds all the peaks corresponding to the S1 and S2 intervals and doesn’t detect any of the noise peaks:

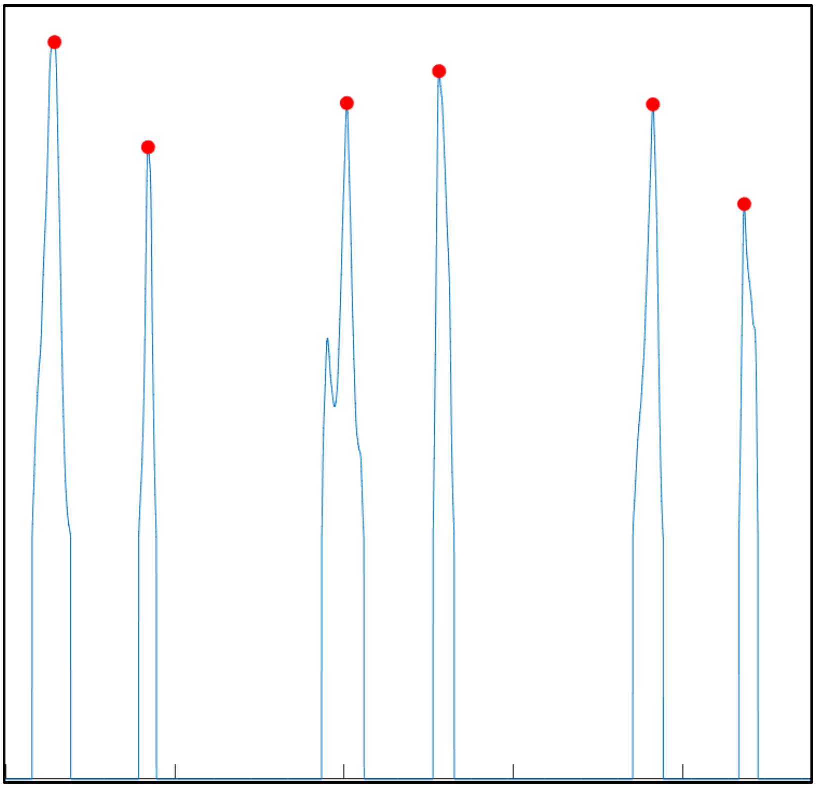

Figure 25 – Peaks S1-S2

However, this situation is not always reached, as in some cases the noise energy can be even higher than an energy of S1 or S2 intervals. This leads to 2 possible situations:

- Actual peaks associated with S1 or S2 interval are lost.
- Noise peaks are not deleted successfully.

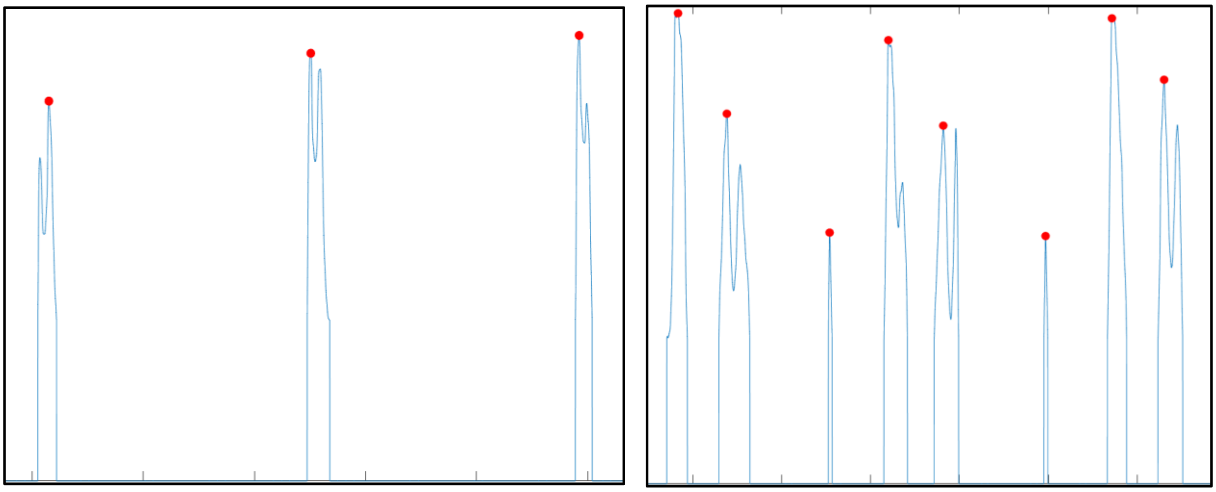

Figure 26 – Lost Peaks and no peaks

As it is not possible to implement a perfect processing only using the energy searching techniques, more sophisticated algorithms are implemented to insert or delete peaks, if the distances between the peaks don't have the next shape:

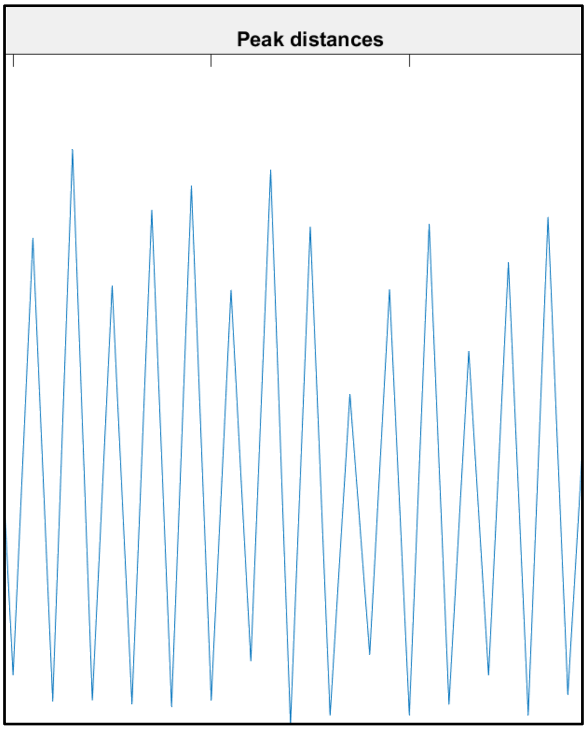

Figure 27 – Ideal Peak Distances

That means that for the PCG signals the peak distances must decrease and increment in each step.

Two consequently increasing values indicate that a peak was lost. Two consequently decreasing values indicate that a noise peak was detected. This way it is needed to implement different code blocks to handle these situations and solve them automatically.

Insertion of lost peaks

This code segment is responsible for checking if there are missed S2 intervals in the signal and, if needed, inserting peaks to rectify these misses.

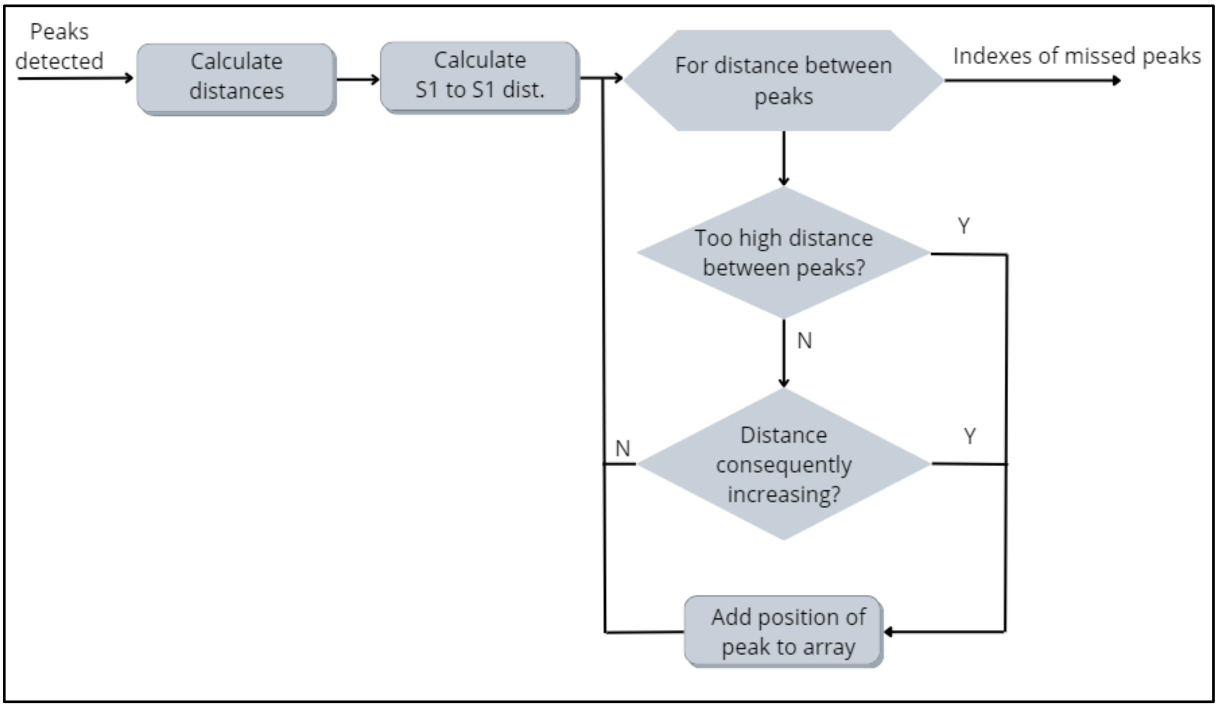

Figure 28 – Missed peaks detection

Calculate distances:

- Differences between consecutive peak locations are computed.

Calculate distances from S1 to S1:

- It is calculated using a histogram the distance between S1 interval and S1 distance that is normally in the signal. For arrhythmic signals this distance can vary, but the mean distance is computed. This calculated value can be used in order to determine, at which distance 2 consecutive S1 peaks are located and where this distance is too high, so we can determine the location of a lost S2 peak.

Once these distances are computed the next steps can be performed:

- The missed peaks are detected based on various conditions. It can be defined that a peak is missed if the distance between two other peaks is too high or it is increasing in two consecutive steps. The indexes, at which the missed peaks are detected are appended to an array for future use. In S2 peaks are lost the shape of peak distances that will indicate that peaks need to be inserted is:

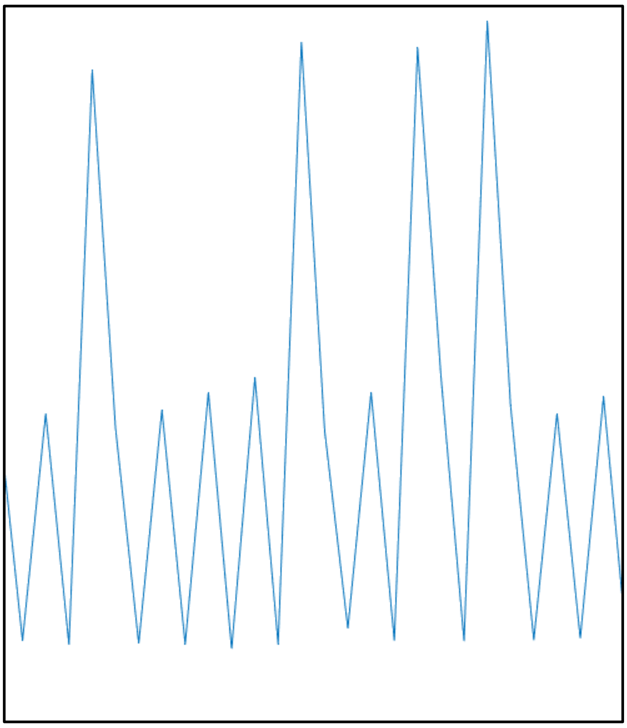

Figure 29 – Peak Distances indicating lost peaks

Once these indexes are detected a new step to insert the peaks is implemented.

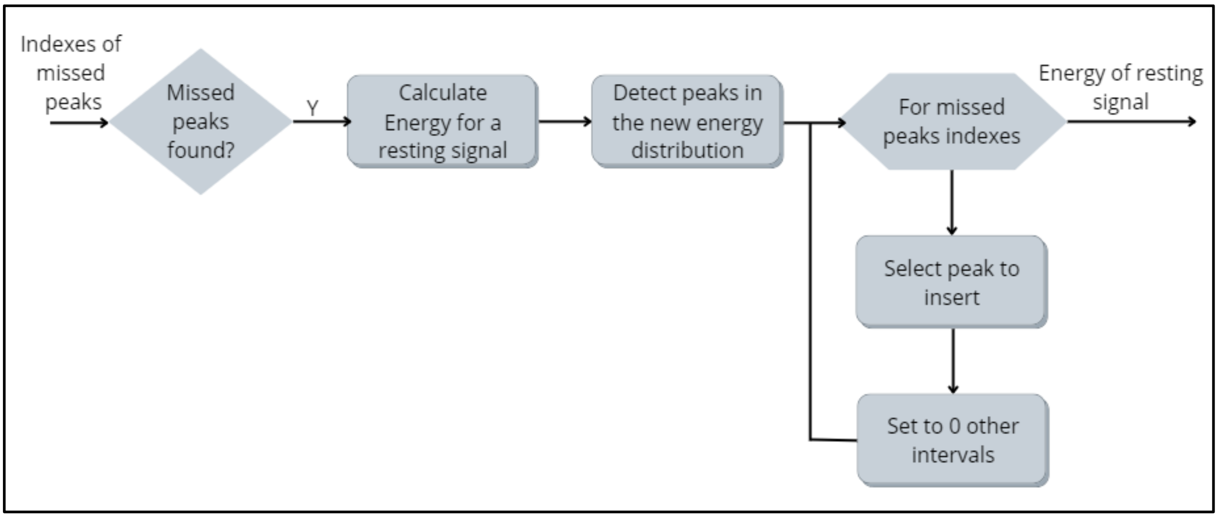

Figure 30 – Selecting lost peaks

Energy Distribution Calculation:

- Calculation of a new energy distribution for the modified signal, that won't contain the intervals of the originally found peaks.

- The peaks will be detected in the new energy distribution.

Inserting Peaks:

- Identification and selection of peaks to insert based on conditions, handling scenarios where multiple peaks are found in a missed interval. The peak should be located in an interval, where it is detected that a peak is lost, located in a reasonable distance based on the histogram calculation of most frequent distances and should be large and high enough. In the figure in blue there are already found peaks. In red the discarded candidates to be inserted and in yellow the selected peaks to be inserted:

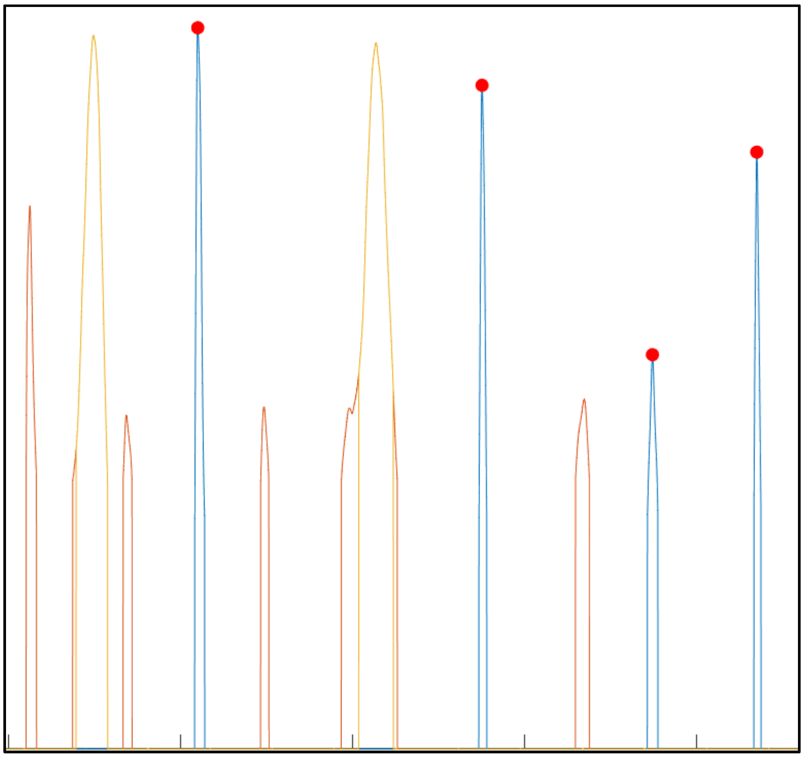

Figure 31 – Missed Peaks selection

The next steps of the algorithm are:

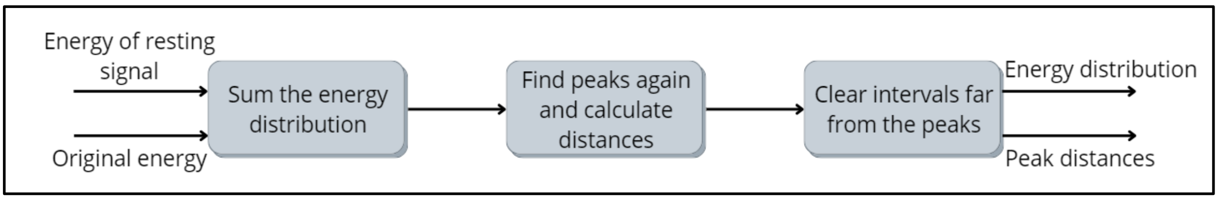

Figure 32 – Summing the original and new energy distributions

The modified energy distribution is considered, such that it will contain the originally calculated peaks and the peaks detected in this stage will be added to the distribution.

The peaks are refined by finding peaks again in the integrated energy distribution. The intervals around the refined peak locations are cleared to avoid redundant peaks.

Deleting of noise peaks

This code segment is responsible for refining the identification of peaks in the signal by considering subsequences of consecutive peaks. This code aims to refine the identification of peaks in the signal by considering subsequences of consecutive peaks, using an automated refinement process and deleting the peaks not corresponding to S1 or S2. The goal is to improve the accuracy of identifying S1 and S2 peaks in the signal.

The code first checks if the peak distances exhibit two consecutive values in descending order, meaning the peak distance would have the next shape:

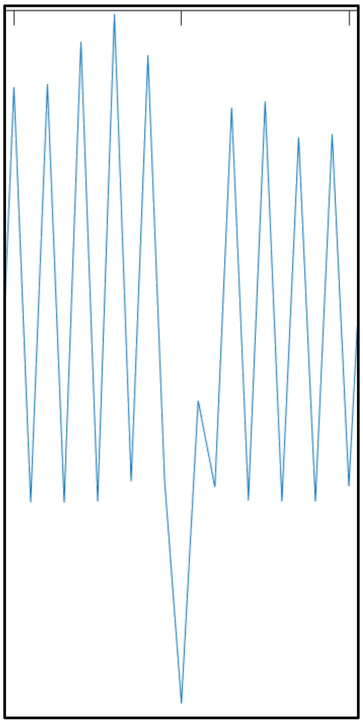

Figure 33 – Noise Peak Distances

In the first place If consecutive values are found, an array is initialized to store indexes corresponding to consecutive peak distances.

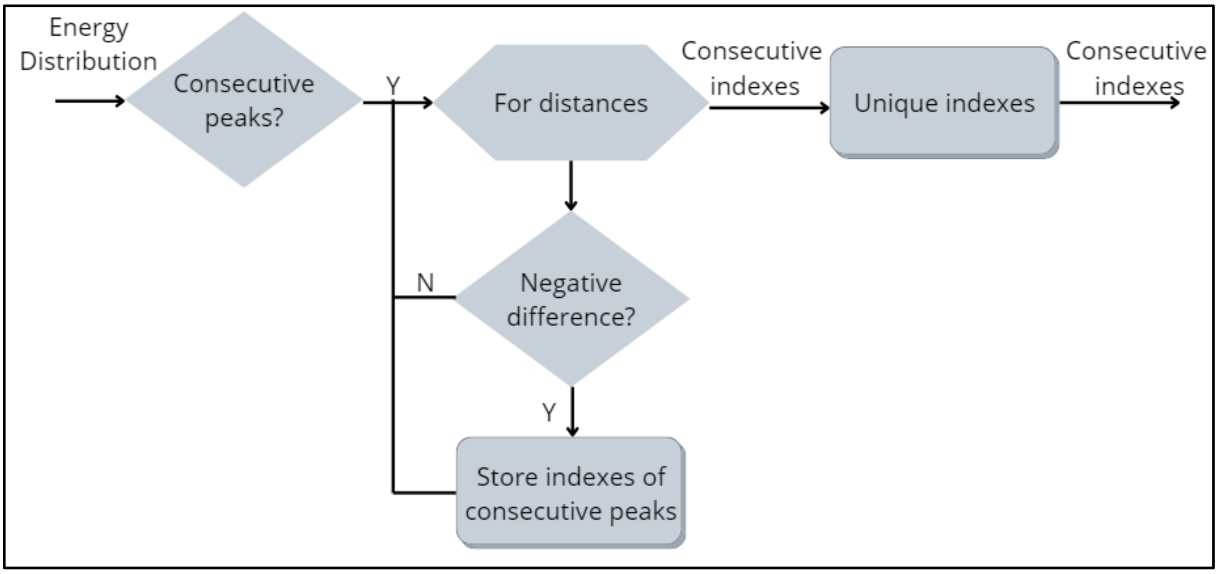

Figure 34 – Consecutive indexes searching

Iterating Through Peak Distances:

- The code iterates through the differences between consecutive peak distances (difference array) and identifies consecutive peaks with negative differences.
- For negative differences, the indexes of three consecutive peaks are stored in the new array. The array of consecutive indexes is then filtered to keep unique indexes.

The next step consists in removing the peaks that are responsible for generating the array of consecutive indexes.

A histogram is computed for the peak distances, and the distance corresponding to the difference between S1 and S2 peak is found.

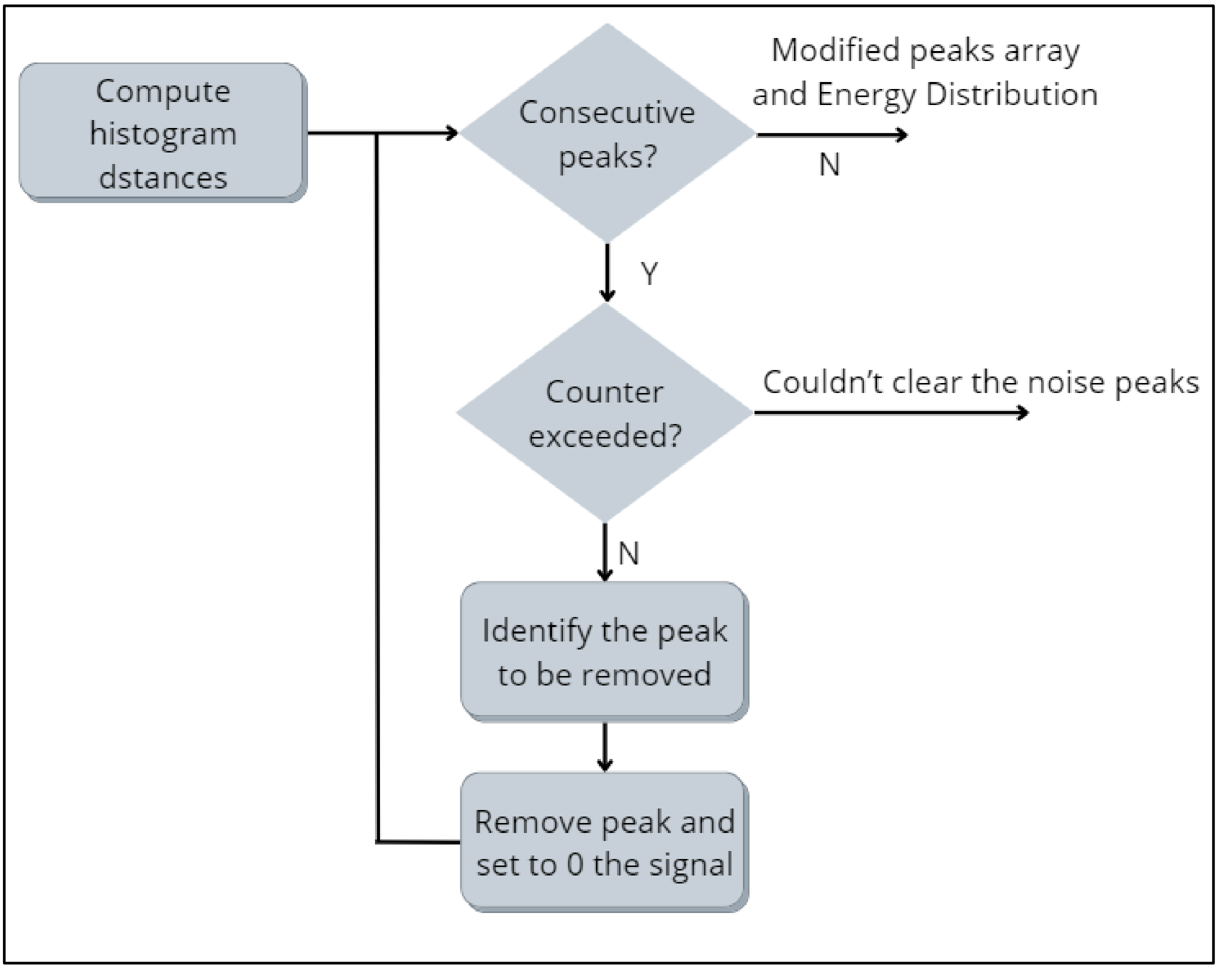

Figure 35 – Removing of noise peaks

Refinement Loop:

- A loop is initiated to refine the peak identification process. It continues refining as long as the peak distances do not exhibit two consecutive values in descending order.
- The loop attempts to refine the identified peaks based on certain conditions, such as the distance between the peaks and the energy contained in each peak. The loop has a counter, and the refinement process is limited to 10 iterations. If the condition is not satisfied within these iterations, the loop exits.

Peak Removal and Zeroing:

- The identified peaks, which need to be removed, are deleted from the list of peaks. In this example the noise peak is much smaller than the other peaks and it is located in a distance that doesn't much with the histogram values, so this peak is deleted:

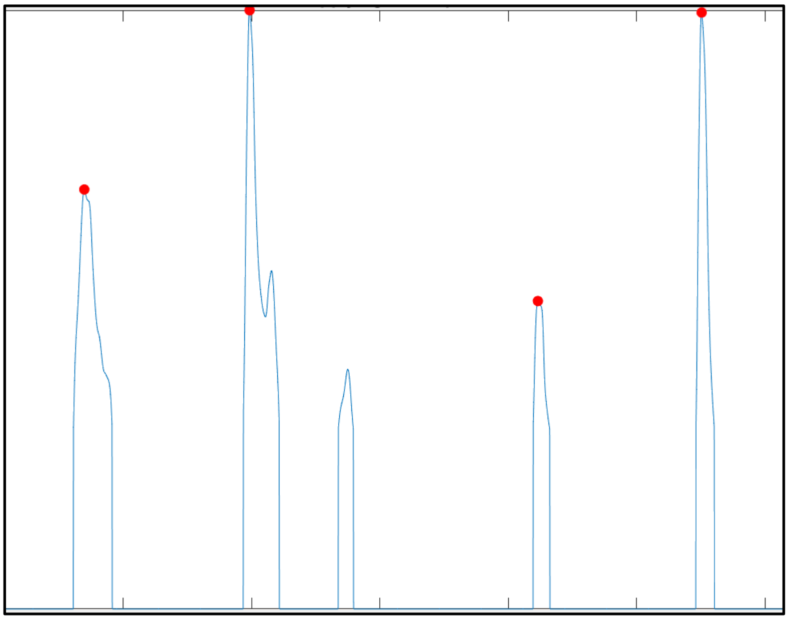

Figure 36 – Deleted Peak

- The signal is modified by zeroing out the regions around the removed peaks.

Selection of S1 and S2 peaks

This code segment focuses on the detection of S2 peaks in the filtered signal. This code identifies S2 peaks in a given signal. It creates separate signals (x_s2 and x_s1) corresponding to S2 and S1 peaks, respectively. The detection process involves finding peaks in the difference signal (x_s1) with a refined mean distance calculation.

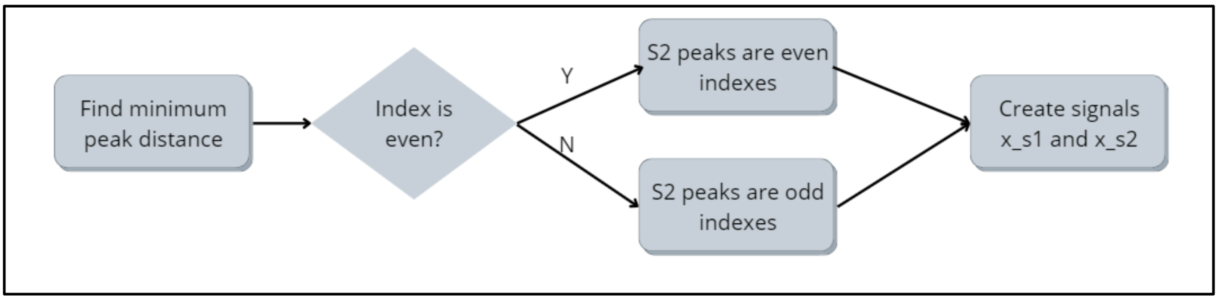

Figure 37 – Selection of S1 and S2 peaks

Finding Minimum Peak Distance:

- It is calculated the minimum value and its index in the array of peak distances.

Determining S2 Peak Locations:

- It is checked if the index corresponding to this distance is even or odd. If even, S2 peaks are selected from even indices of peak locations (every second element starting from the second element). If odd, S2 peaks are selected from odd indices of peak locations (every second element starting from the first element).

Creating S2 Signal (x_s2):

- A new signal (x_s2) is created by setting values away from each S2 peak to 0.

Creating S1 Signal (x_s1):

- A signal representing S1 peaks (x_s1) is obtained by subtracting x_s2 from the original signal.

Search S1 energy

This code segment is responsible for detecting S1 intervals in a signal (x_s1). This code calculates and concatenates energy vectors associated with S1 intervals in the x_s1 signal. It addresses short intervals by identifying and removing them based on a specified difference threshold.

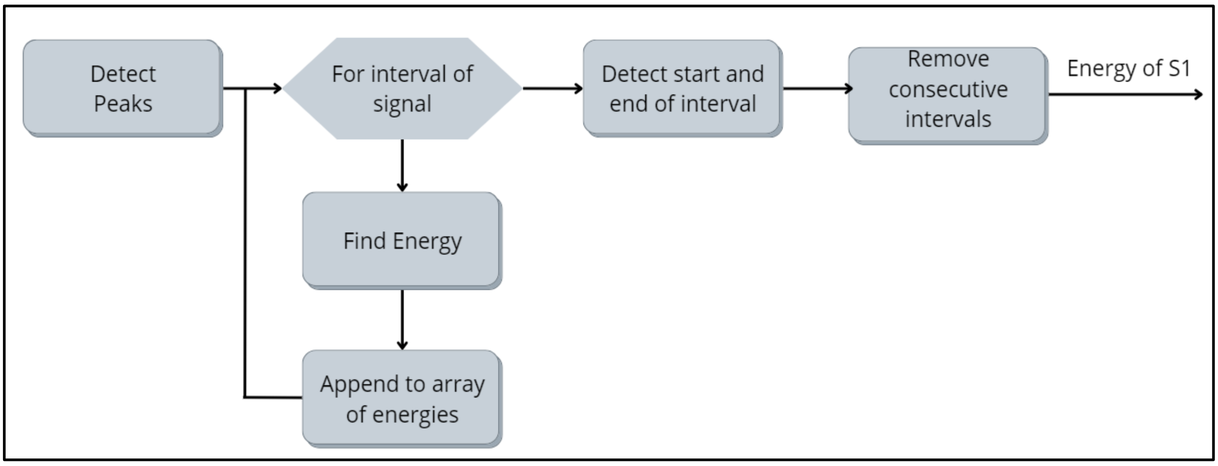

Figure 38 – Search S1 energy

Peak Detection:

- Peaks in absolute value of x_s1 are detected using the “findpeaks” function with specified parameters (“MinPeakDistance” and “MinPeakHeight”).

Concatenated_vector_S1 is initialized as an empty vector to store energy vectors related to S1 intervals.

Vector Calculation for S1 Intervals:

- A loop iterates over the S1 intervals using peak locations. For each interval, an energy vector is computed using the function “energy_finding”. The signal used for this calculation is the portion of x_s1 between the midpoints of consecutive S1 peaks. The resulting vector is appended to concatenated_vector_S1.

Detection of indexes:

- Indices of omegaS1 with absolute values less than 0.001 are identified. Start and end indices of non-zero intervals are found (startIdx_S1 and endIdx_S1).

Removing consecutive intervals:

- Intervals with a difference between start and end indices less than 425 ms are identified (indices_with_difference_less_than_500). If such intervals exist, the one with the higher corresponding omegaS1 value is marked for deletion. The start and end indices of the marked interval are removed.

Search S2 energy

This code segment processes the x_s2 signal to detect S2 intervals. It involves peak detection, mean difference computation, zeroing values away from peaks, and the calculation of energy vectors for each S2 interval. The final omegaS2 vector represents the energy distribution associated with S2 intervals.

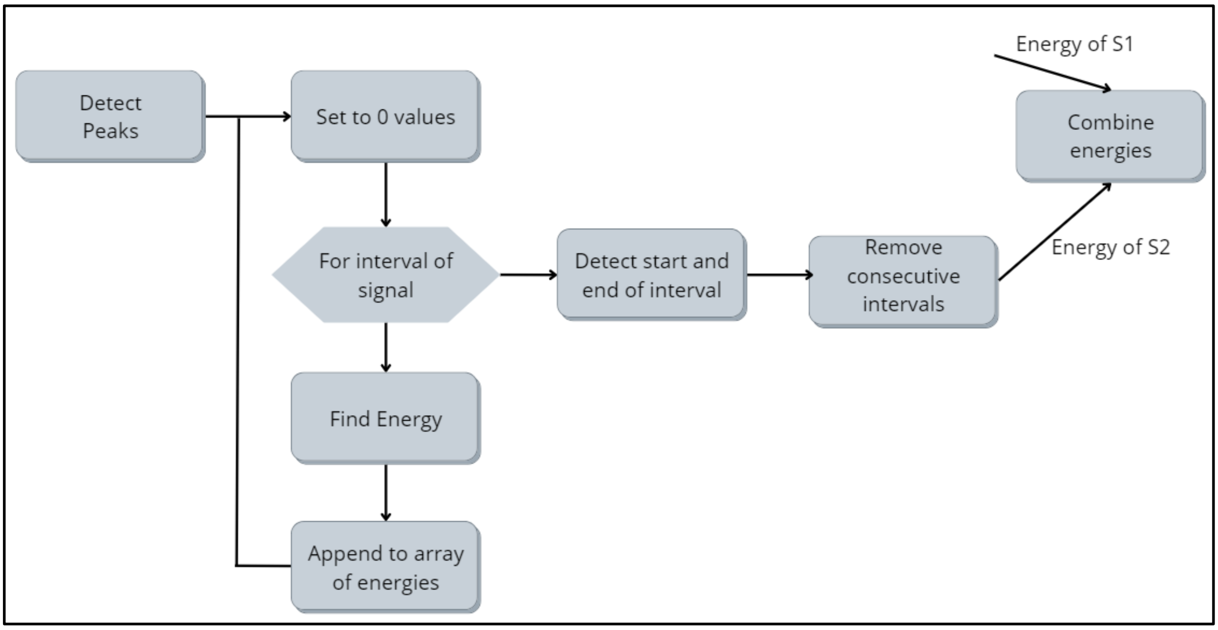

Figure 39 – Search S2 energy

Peak Detection:

- Peaks in absolute value of x_s2 are detected using the "findpeaks" function with specified parameters ("MinPeakDistance" and "MinPeakHeight").

Set to zero values:

- The differences between consecutive peak locations are computed. The mean difference is calculated. Values in x_s2 that are away from each peak location by mean difference/2 are set to 0.

Energy Vector Calculation:

- The code calculates energy vectors for S2 intervals using the energy_finding function. A concatenated vector is initialized as an empty vector. The first vector is computed for the first S2 interval, and subsequent vectors are computed for the middle intervals. A vector is also computed for the last S2 interval. These vectors are appended to concatenated vector.

Detection of indexes:

- Indices of omegaS2 with absolute values less than 0.001 are identified. Start and end indices of non-zero intervals are found (startIdx_S2 and endIdx_S2).

Removing consecutive intervals:

- If intervals have a difference between start and end indices less than 450 ms, they are marked for deletion. The start and end indices of the marked intervals are removed.

Energy Combination:

- The energy vectors of S1 (omegaS1) and S2 (omegaS2) are added to get the combined energy vector (energy).

Save features

This code segment is responsible for generating and saving visual results, as well as saving specific data related to S1 and S2 intervals.

Plotting:

- A figure is created with two subplots (subplot (211) and subplot (212)):

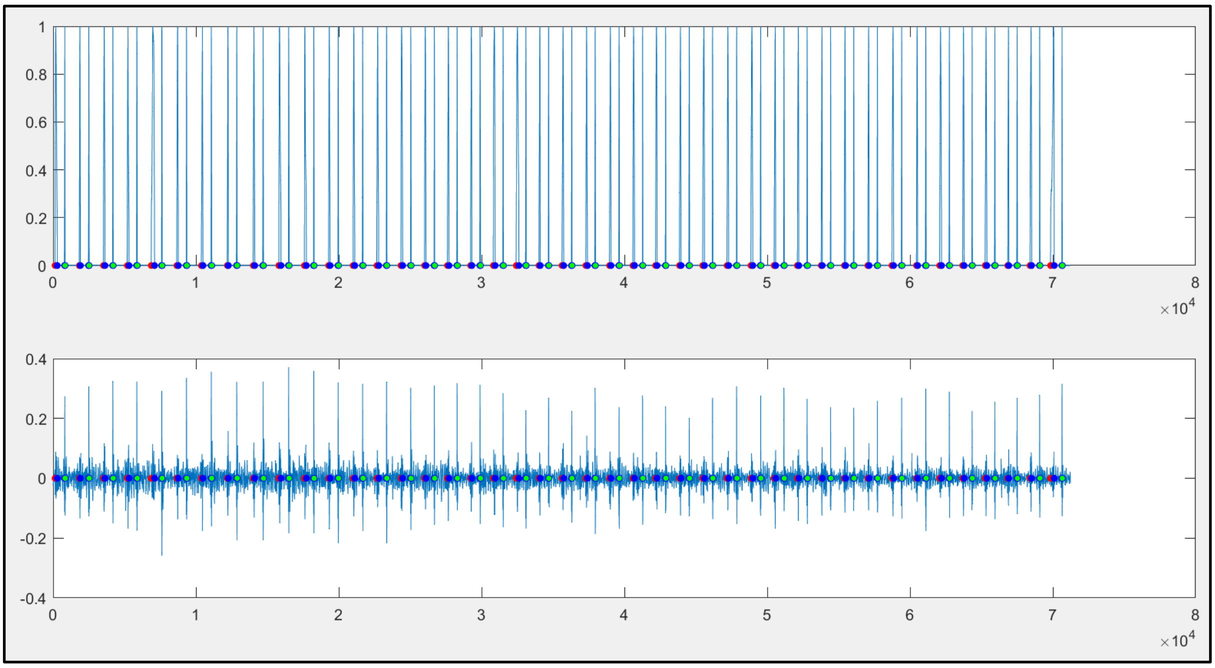

Figure 40 – Resulting plot from the algorithm

- The first subplot displays the energy signal (energy) with markers indicating the start and end indices of detected S1 and S2 intervals. Different marker colors are used for differentiation.

- The red point indicates the start of the S1 interval. The blue point indicates the end of S1 intervals. The yellow point indicates the start of the S2 interval and the green point indicates the end of the S2 interval.

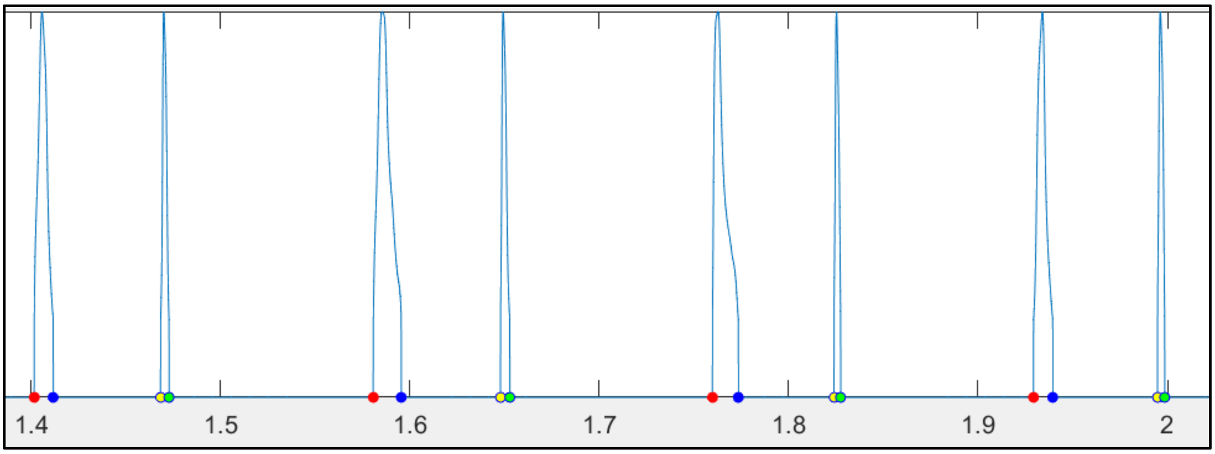

Figure 41 – Energy with points of start and end

- The second subplot displays the original audio signal with similar markers for S1 and S2 intervals:

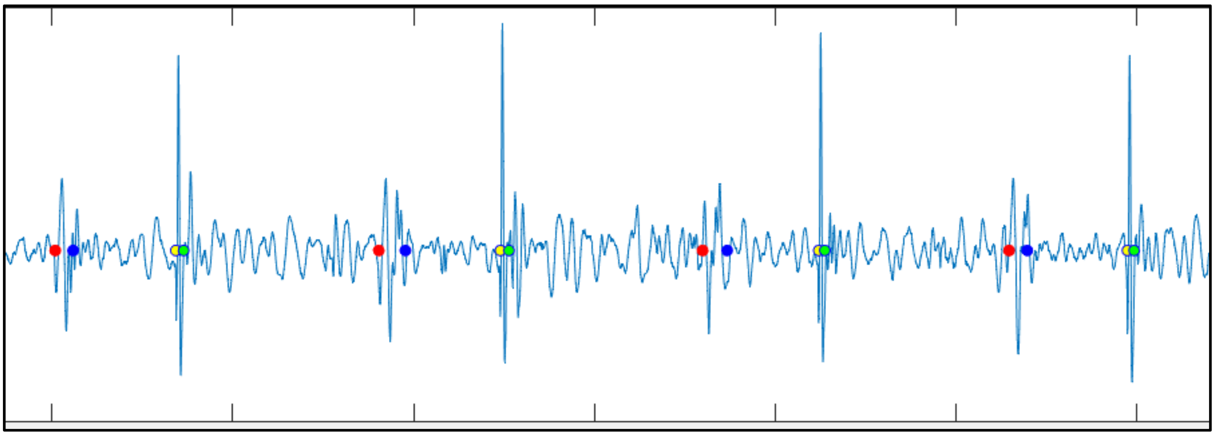

Figure 42 – Original signal with points of start and end

File Saving:

- The start and end indices of S1 and S2 intervals are saved in separate MAT files (*.mat) using the “save” function.

- The energy vectors (omegaS1 and omegaS2) corresponding to S1 and S2 intervals are also saved in separate MAT files.

In summary, this code segment provides visual representations of the detected S1 and S2 intervals on both the energy and audio signals. Additionally, it saves detailed information, such as start and end indices and energy vectors, for further analysis or future reference.

5.3 Results of Data Acquisition

The database contained 409 signals. Some of them were noisy or didn't exist simultaneously for ECG and PCG.

First, the ECG processing was implemented and tested. This processing has led to the next results:

Result	Quantity
Total of signals	409
The ECG signal was processed correctly leading to a correct detection of Q ant T waves	293
The signal was processed correctly, but contains some missed intervals or deviations	19
The signal wasn't processed correctly, because it didn't exist or it was too noisy to get a correct marking of Q and T	97
Performance	**72%**

Table 4 – ECG results

After finishing the processing of ECG signals, the PCG signals were processed that correspond to correctly processed ECG signals as for a prediction model it is required to have the results for both signals for the same patient.

Result	Quantity
Total	293
The PCG signal was processed correctly leading to a correct detection of S1 ant S2 intervals	188
The signal wasn’t processed correctly, because it didn’t exist or it was too noisy to get a correct marking of S1 and S2	105
Performance	**64%**

Table 5 – PCG results

The overall result is:

Result	Quantity
Total	409
The PCG signal was processed correctly leading to a correct detection of S1 ant S2 intervals	188
The signal wasn’t processed correctly, because it didn’t exist or it was too noisy to get a correct marking of Q and T or S1 and S2	221
Performance	**46%**

Table 6 – Overall result

This way from the dataset 188 signals were obtained that can be prepared in the next step and be the input for the prediction algorithms.

5.4 Data Preparation

Visualization

In the first place the processed information can be visualized in the same plot to observe how PCG and ECG features correlate with each other. There will be presented some plots, in which is plotted the ECG with the marked points of Q, R and T in blue and on the other hand the red plot will represent the S1 energy and the green plot will present the S2 energy:

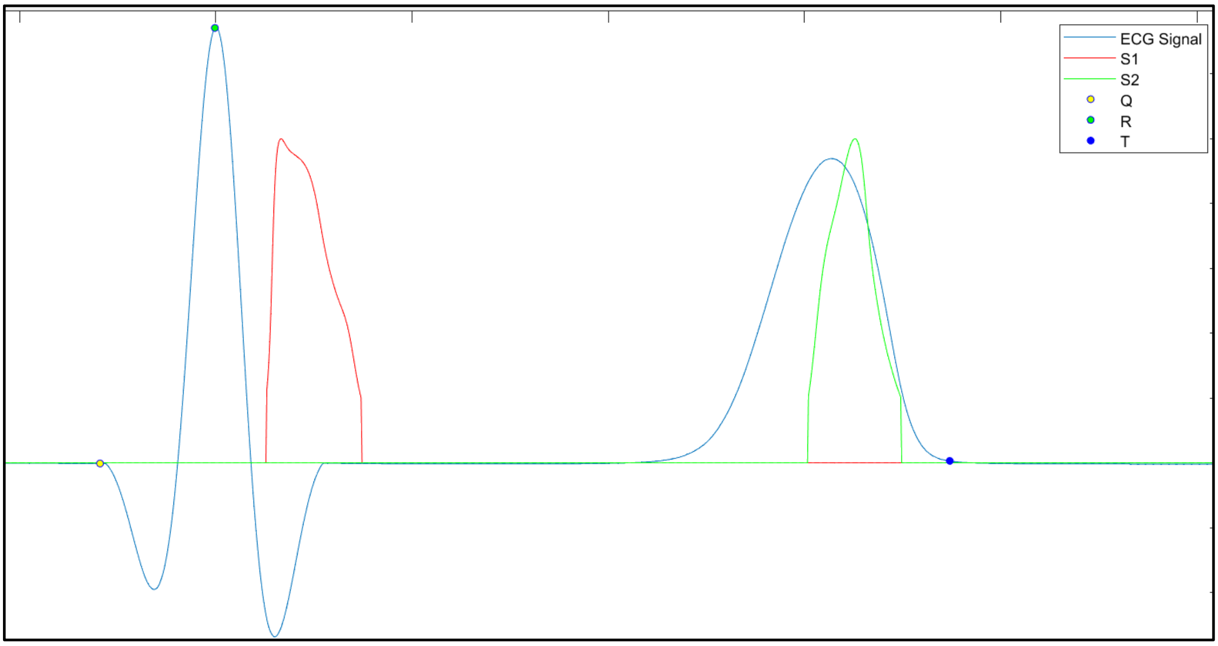

Figure 43 – Plot 1. ECG and PCG energy

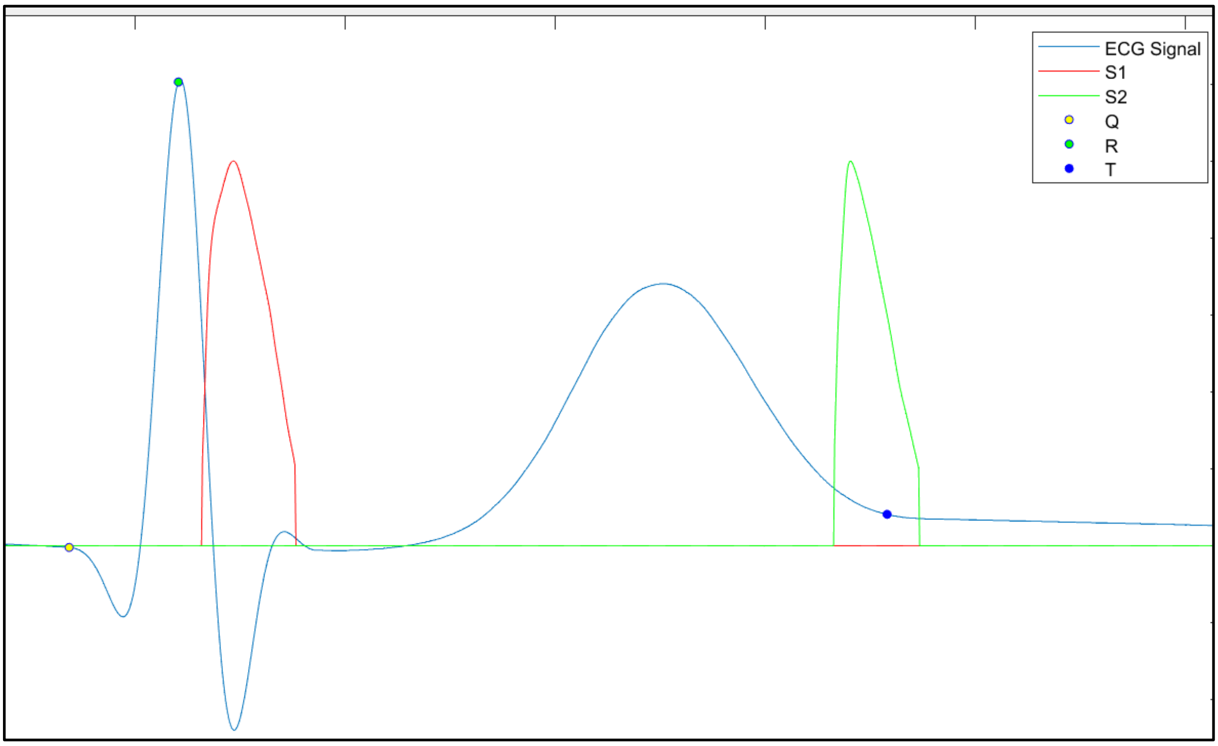

Figure 44 – Plot 2. ECG and PCG energy

After the visual information is obtained and it is possible to observe that the energy of S1 and S2 does not always have the identical location with respect to the ECG, the data will be prepared to be the input of prediction models.

Data frame Generation

A set of signal numbers is defined, and the code iterates through these signals to process each one.

- Processing and Filtering: Several arrays (e.g., Q, T, S1_on, S1_off, S2_on, S2_off, omegaS1, and omegaS2) are loaded from the results of the correctly processed signals.
- Signal Alignment and Filtering: The code aligns and filters the signals based on specific conditions, such as the distance between Q, T, S1, and S2 points. In this project have been considered the next values for the signal to be acceptable:
    - Maximum distance between the end of S1 and the start of T = 350 ms;
    - Maximum distance between Q and T = 700 ms;
    - Maximum distance between Q and the start of S1 = 200 ms;
    - Maximum distance between Q and the start of S2 = 750 ms;
    - Maximum distance between the start of S2 and T = 200 ms;

- Various features are extracted from the signals, including intervals, absolute maximum values, and indices of interest.
- Some features are normalized, and additional features like mean intervals and QT intervals are calculated.
- The code checks if the lengths of extracted arrays are equal and handles cases where lengths differ.
- Data Aggregation: Extracted features are aggregated into a matrix named Table, where each row corresponds to the segment of the signal, and each column corresponds to a specific feature.

The values are written in function of S1_on. So, the other variables are the actual values in graph minus the value of S1_on. All the values written to a data frame to a future analysis are:

Variable	Description
Index	The index indicates to which signal the segment corresponds
PCG Features	
S1_on	Indicates the start of S1
S1_off	Indicates the end of S1
S2_on	Indicates the start of S2
S2_off	Indicates the end of S2
S1_interval	Indicates the length between previous S1 location and the considered.
max_index_S1	Indicates where the energy of S1 reaches the maximum
abs_max_value_S1	Indicates the maximum value of S1 in the segment
max_index_S2	Indicates where the energy of S2 reaches the maximum
abs_max_value_S2	Indicates the maximum value of S2 in the segment
mean_S1_interval	The mean value of all the S1 intervals for the considered signal
mean_S1_off	The mean value of all the S1_off values for the considered signal
mean_S2_on	The mean value of all the S2_on values for the considered signal
mean_S2_off	The mean value of all the S2_off values for the considered signal
ECG Features	
Q	Indicates the location of Q
T	Indicates the location of T
QT	Indicates the value of QT
QTC	Indicates the value of QTC

Table 7 – Data frame Variables

Final Output: The Table matrix and associated files can be used for further analysis or machine learning applications:

	0	1	2	3	4	5	...	13	14	15	16	17	18
0	2	2298	303	598	671	1479.6	...	734.52	-144	578	722	717.64	0.40107
1	2	3793	244	444	720	1495.0	...	734.52	-118	603	721	717.64	0.39938
2	2	5329	117	631	766	1536.0	...	734.52	-144	574	718	717.64	0.39473
3	2	6819	337	645	711	1490.0	...	734.52	-117	602	719	717.64	0.39877
4	2	8366	132	602	699	1547.0	...	734.52	-144	573	717	717.64	0.39338
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6592	409	49070	175	744	804	2308.0	...	782.92	85	718	633	687.77	0.29278
6593	409	51461	197	702	787	2391.0	...	782.92	65	670	605	687.77	0.27239
6594	409	53734	147	693	786	2273.0	...	782.92	88	662	574	687.77	0.26598
6595	409	56017	191	707	797	2283.0	...	782.92	-39	671	710	687.77	0.33321
6596	409	58304	166	705	796	2287.0	...	782.92	68	677	609	687.77	0.28240

Figure 45 – Data frame

6. PREDICTION MODELS

As the data available to make the predictions doesn't contain a significant number of signals to analyze and it is necessary to train and predict the results in a reliable way, the leave-one-out cross validation will be used for each regression model.

Leave-one-out cross-validation, or LOOCV, is a configuration of k-fold cross-validation where k is set to the number of examples in the dataset.

LOOCV is an extreme version of k-fold cross-validation that has the maximum computational cost. It requires one model to be created and evaluated for each example in the training dataset.

The benefit of so many fit and evaluated models is a more robust estimate of model performance as each row of data is given an opportunity to represent the entirety of the test dataset. [22]

In the case of the generated data frame in the previous chapter it is not possible to directly use the leave-one-out validation to predict each segment, as other segments of the same signals will be almost identical in some cases and will make the regression model fit too well.

However, the goal is to be able to make a model, able to predict the QT interval, without knowing any ECG feature of the patient. In this case, the data frame will be splatted into training and testing datasets, such that in the training dataset, there will be all the segments not associated with the patient, and in the test dataset there will be allocated the values for the segments of each patient.

6.1 Linear Regression QT estimation

Linear regression estimates the coefficients of the linear equation, involving one or more independent variables, that will be all the PCG features extracted, and predict in a best way the value of the dependent variable, QT interval or T location. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a "least squares" method to discover the best-fit line for a set of paired data. [23]

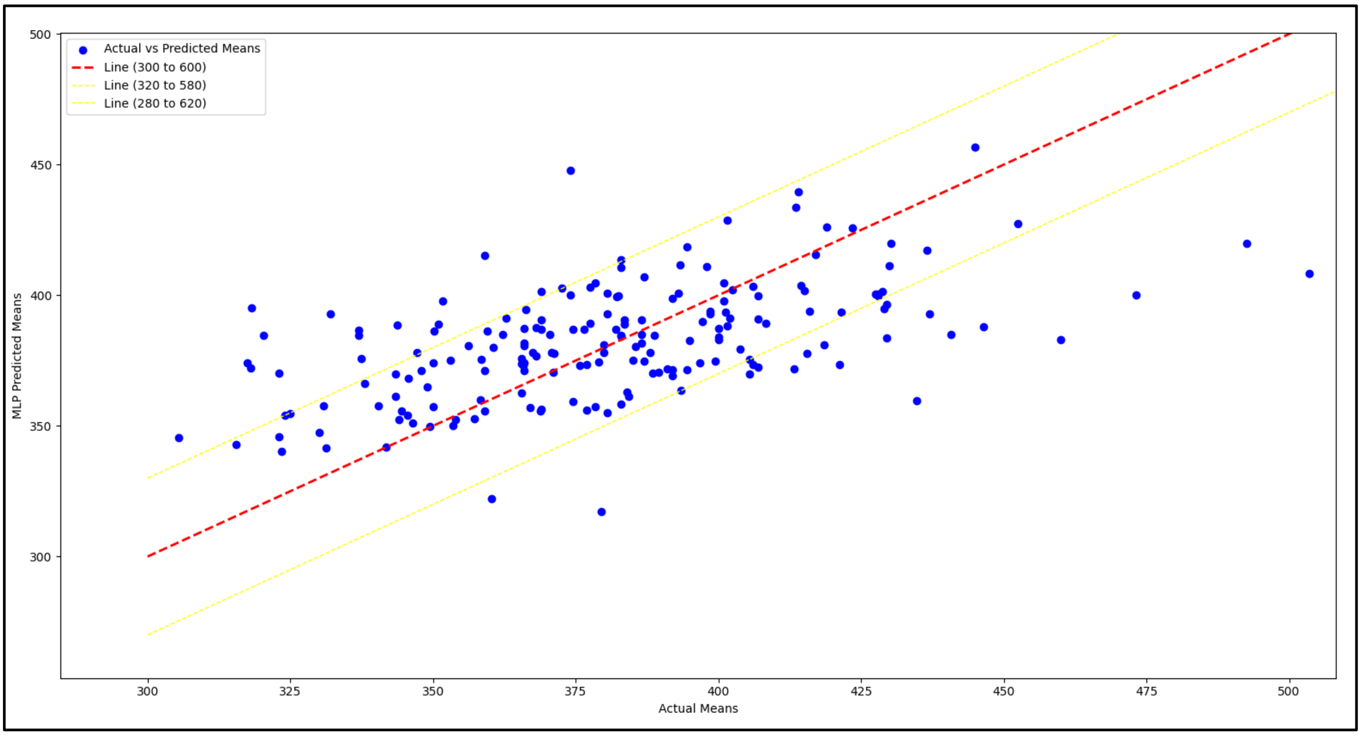

Figure 46 – Linear Regression Results 1

As it can be seen in the graphic for larger actual values of QT the linear regression starts to fit worse. It is associated with the fact that in the dataset there is a small quantity of the values larger than 450 ms, this way the linear regression doesn't fit well for these values.

If the dataset was larger, with more actual QT intervals higher than 450 ms, the model could predict better the testing values.

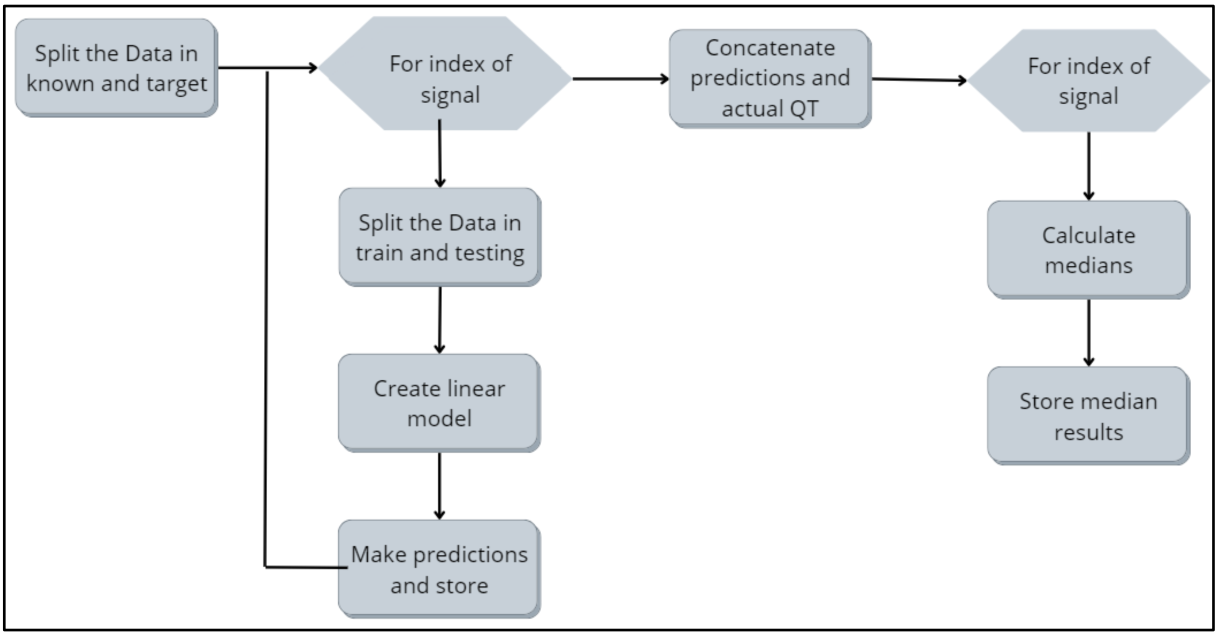

Figure 47 – Linear Regression

As it has been mentioned already, the way of predicting each value of QT will be made using leave-one-out cross validation in order to have more data to test with. Also as in our data there are segments in each row that could be associated with the same signal, so technically the testing will be done for each segment that was extracted from the same signal.

Finally the absolute and relative errors and other statistical parameters are computed.

The results of the error for the linear regression are:

Absolute error (ε)	22.138 ms
Relative percentage error (ε%)	5.806%

Table 8 – Errors for linear Regression 1

Study of the highest errors

In the first place the errors that are for the signals with QT higher than 450 will be removed from the dataset. For them it is possible to say that these high QT values correspond to the prolongation of QT. The predicted values for these high intervals are not accurate, but the predicted value is almost always close to or higher than 400 ms.

The objective of this project is to detect the prolonged QT interval. For this regression it is possible to say that the predicted values higher than 400 ms indicate that the QT could be prolonged, but it is not 100% reliable, as in the dataset there are not enough values to train correctly the machine learning algorithms.

	Group	Actual	Predicted	S1S1
43	86	473.25	400.110762	1848.0
108	218	452.50	427.499833	2283.4
121	240	503.50	408.195340	2268.4
151	310	492.50	420.020929	2396.5
181	400	460.00	383.117450	1902.4

Figure 48 – Errors for QT > 450 ms

This way, in order to compare the study made in this project with the previous study, that only tried to predict the QT for healthy subjects, these values will be removed.

Top 10 Groups with Highest Absolute Error (Actual <= 450):

	Group	Actual	Predicted	S1S1	Error
185	408	318.25	395.321788	1782.7	77.071788
27	50	434.75	359.546831	1653.8	75.203169
169	361	374.00	447.950168	2929.3	73.950168
74	153	320.25	384.544920	1851.7	64.294920
98	194	379.50	317.205471	1750.9	62.294529
19	35	332.00	392.804774	1750.3	60.804774
175	381	446.50	387.879728	1861.7	58.620272
147	298	317.50	374.082935	1837.2	56.582935
186	409	359.00	415.407470	2204.6	56.407470
34	63	440.75	384.934482	1847.4	55.815518

Figure 49 – Highest errors

Group 408:

The error is caused by the incorrect processing of the ECG and not detecting the t-wave correctly, as in the initial signal is complicated to distinguish it from the noise:

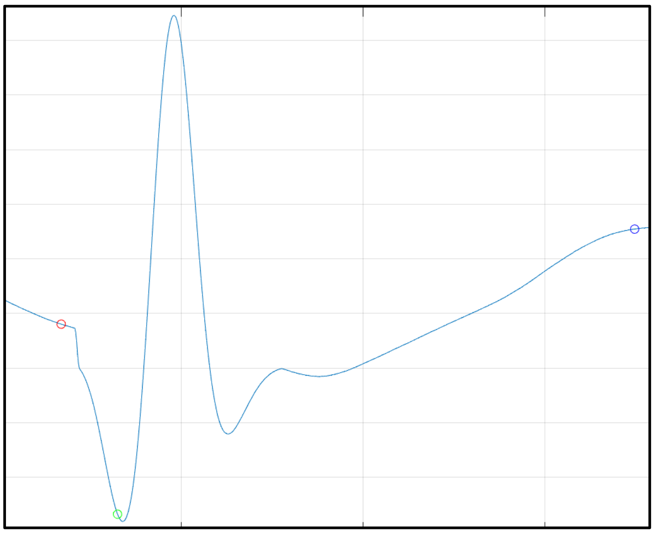

Figure 50 – Incorrect processing t-wave

The signal 50 caused an error because of the incorrectly stored values. It will be reprocessed and introduced to the regression again.

The signal 361 doesn't give a reliable detection of the T-wave as the signal 408, it won't be included in the training and testing, as the calculated QT interval is not reliable.

The error of the signal 153 is not due to processing, the wave S2 appears much after the end of the T-wave, this situation occurs with some other signals and it is a source of error for the regression. It should be studied why this effect happens.

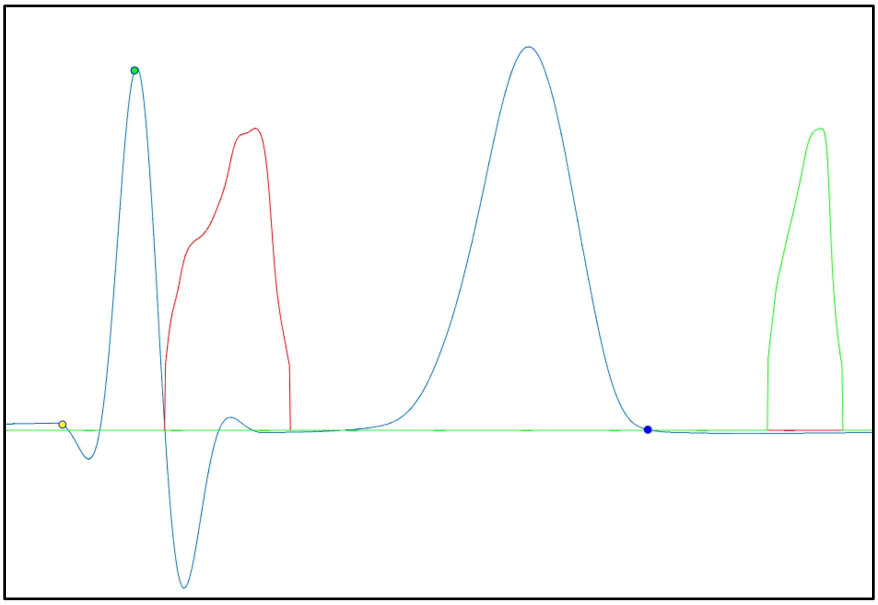

Figure 51 – Error T-wave before S2

The error of the signal 194 is due to not reliable processing of S2 wave, that doesn't have a clear start and end and must associated with some sort of heart disease:

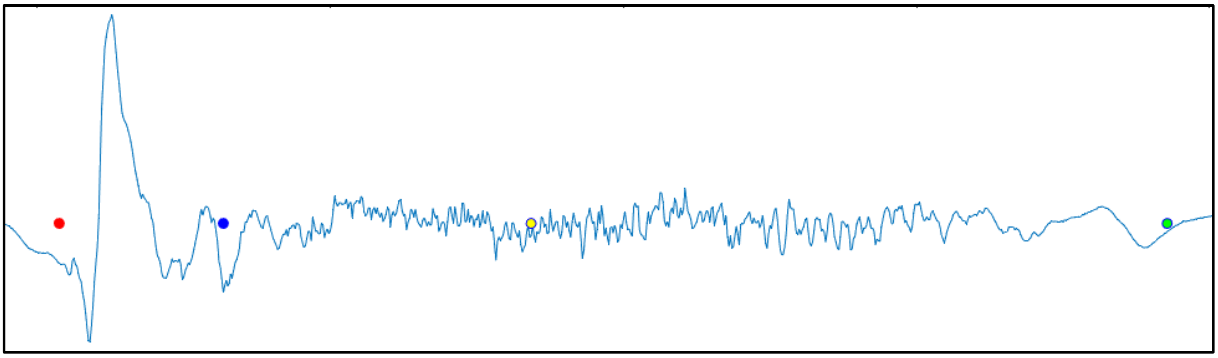

Figure 52 – Error not reliable S2

The signal 35 doesn't give a reliable detection of T-wave, it is hardly separated from the noise.

The signals 381 and 298 are processed correctly.

The ECG processing of the signal 409 is not reliable. The original signal is noisy and contains peaks in the QRS complex, that is complicated to distinguish even with manual observation which one corresponds to Q:

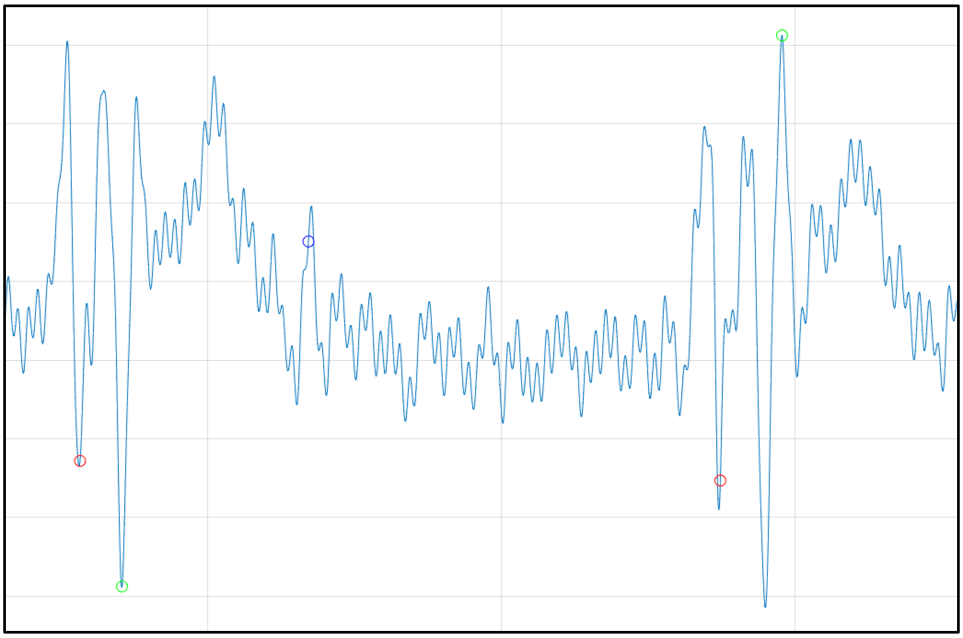

Figure 53 – Original noisy ECG

The signal 63 is processed correctly and has the same issue as the signal 153.

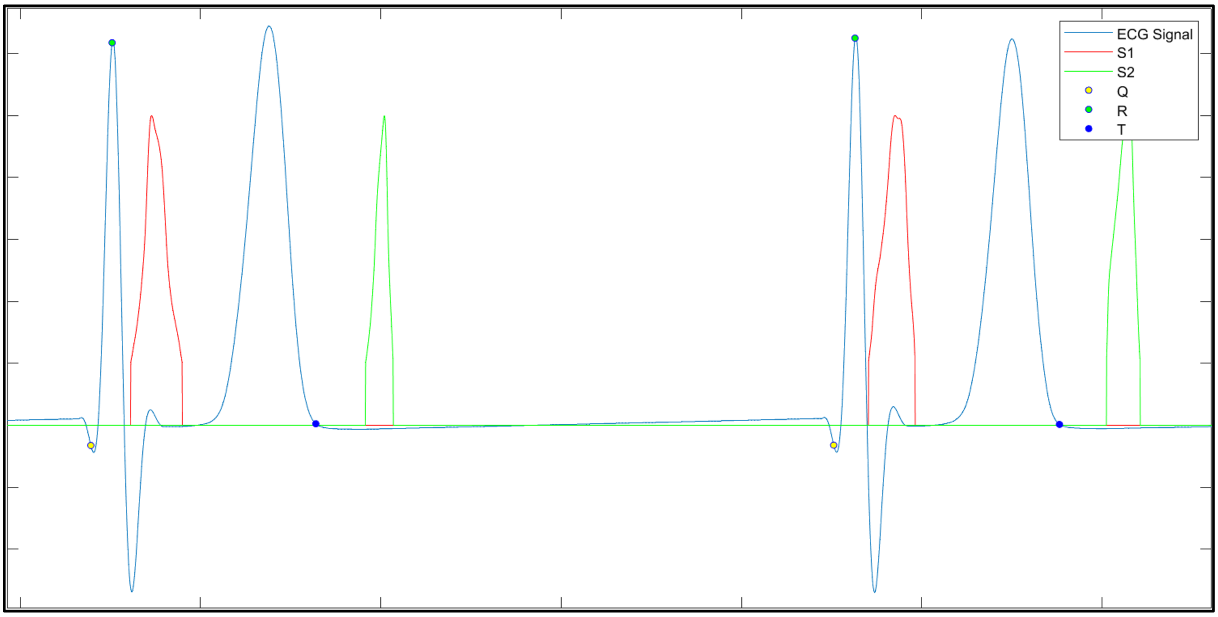

Figure 54 – Signal correctly processed

This way the linear regression will be run again removing from the dataset the signals: 86, 218, 240, 310, 400, 408, 361, 194, 35, 409.

After running the linear regression again with leave-one-out validation the next result is obtained:

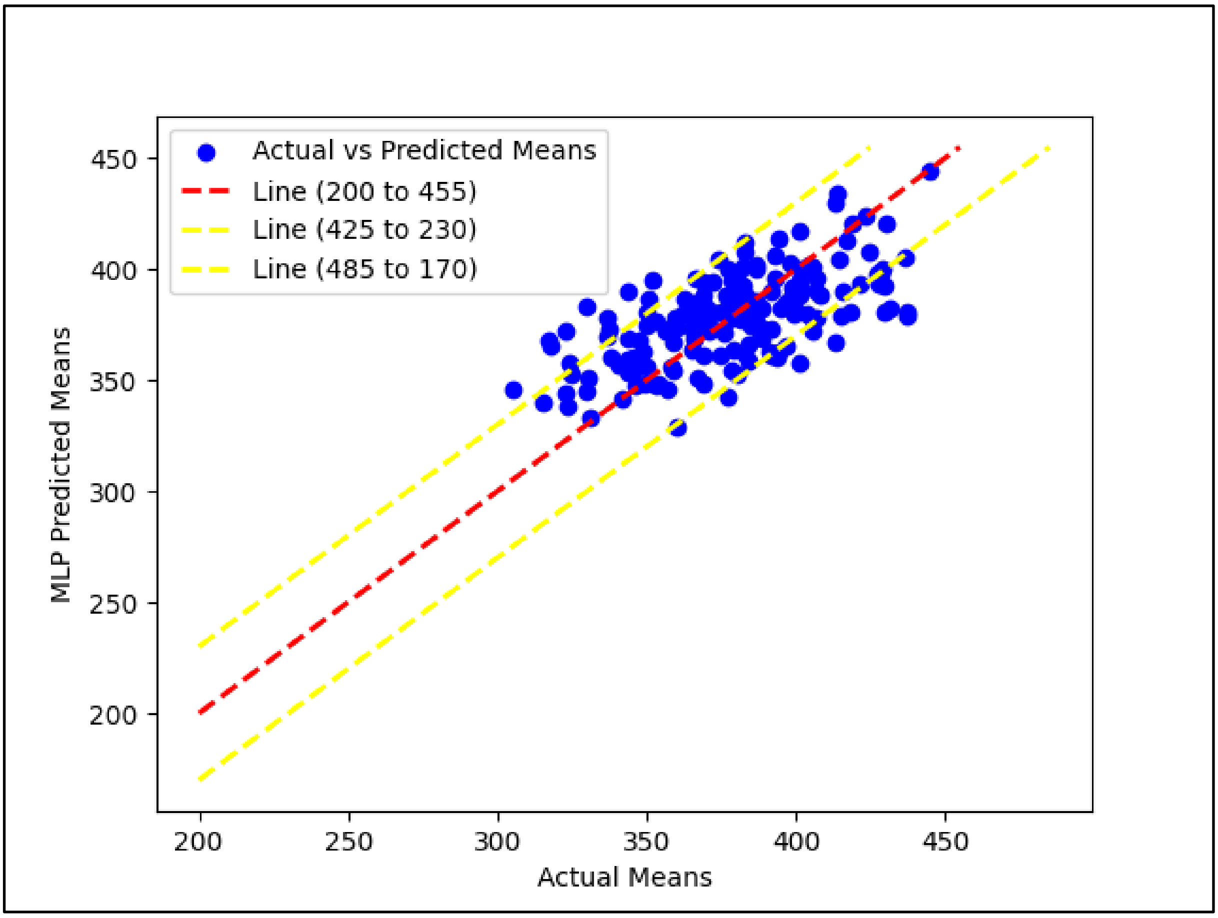

Figure 55 – Second Validation Graphic

The distribution of the absolute error is:

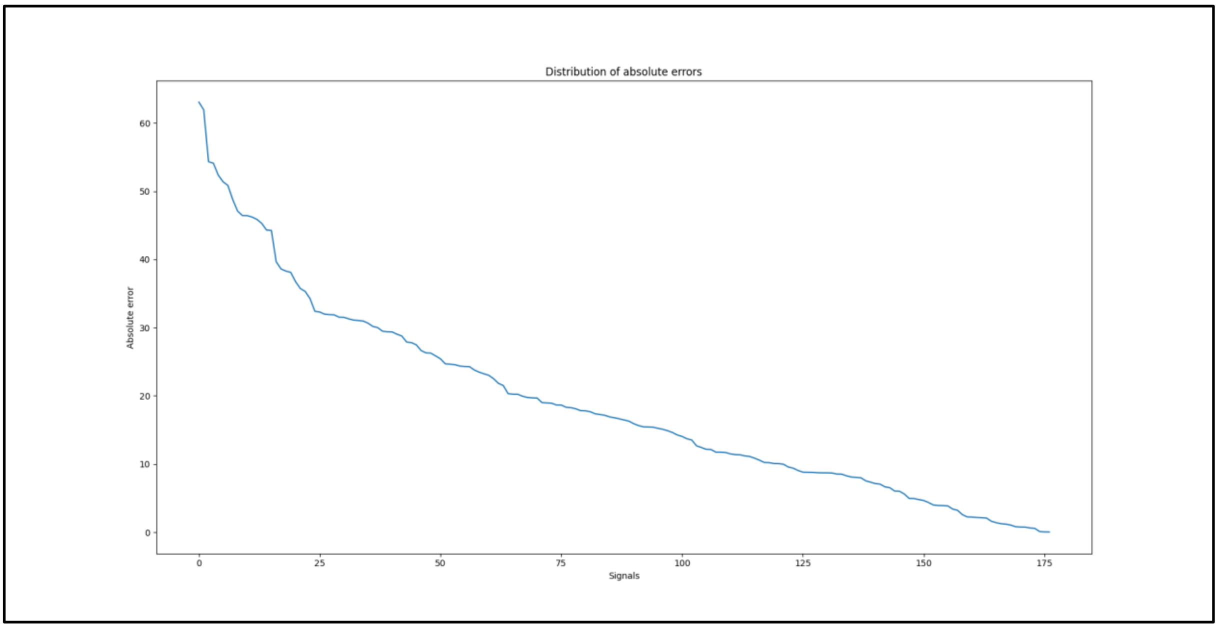

Figure 56 – Error distribution

The results for the corrected linear regression are:

Absolute error (ε)	**18.161 ms**
Relative percentage error (ε%)	**4.794%**
Pearson's correlation coefficient	**0.6425**
p-value Wilcoxon rank-sum test	**0.886**

Table 9 – Corrected errors for linear Regression

QTC calculation

Once the QT is predicted it is possible to apply the different formulas for QTC calculation, as was explained in the State of the Art.

The Framingham's formula can be applied to the predicted and actual data:

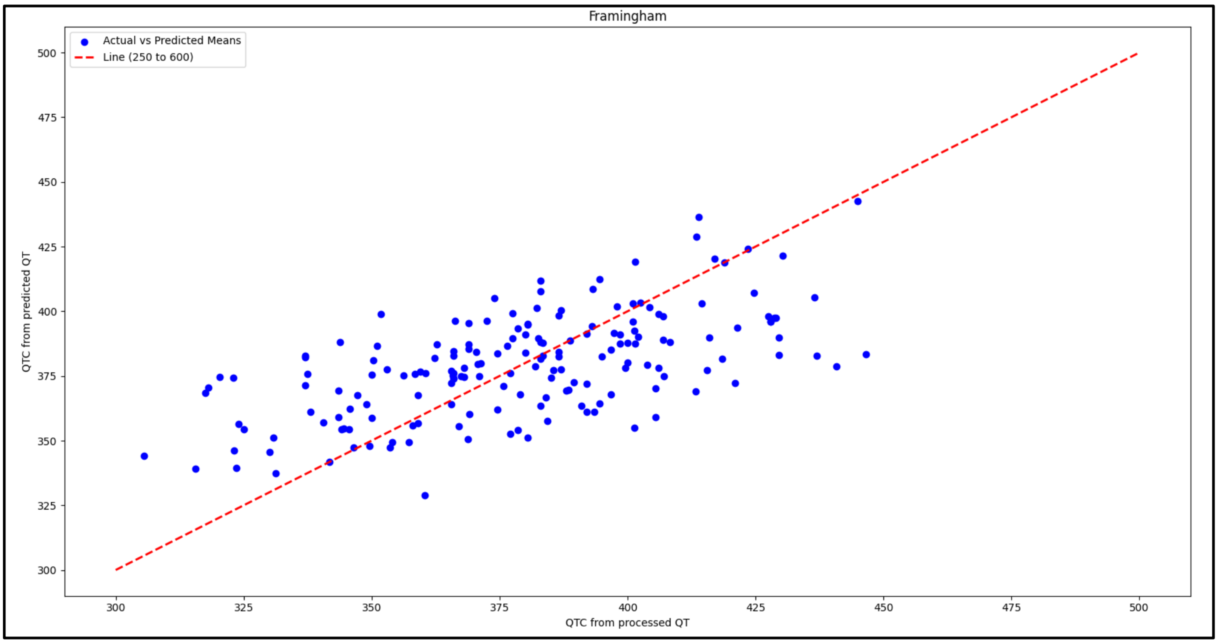

Figure 57 – Framingham's formula

The Bazett's formula can be applied to the predicted and actual data:

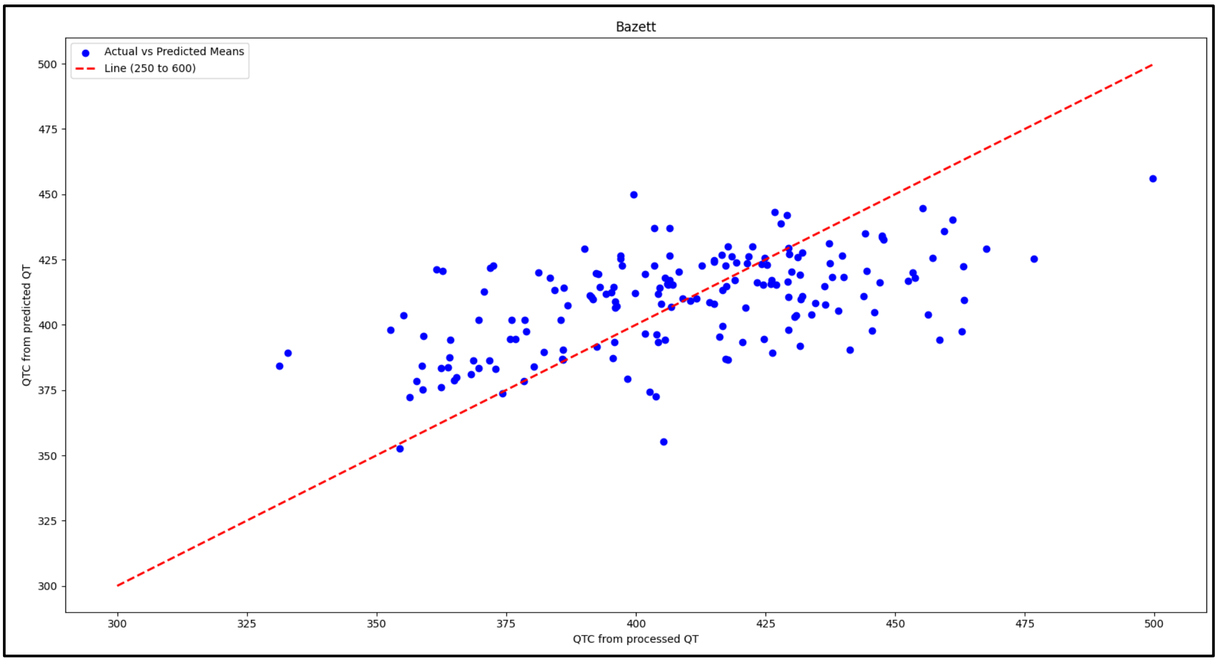

Figure 58 – Bazett's formula

The Fridericia's formula can be applied to the predicted and actual data:

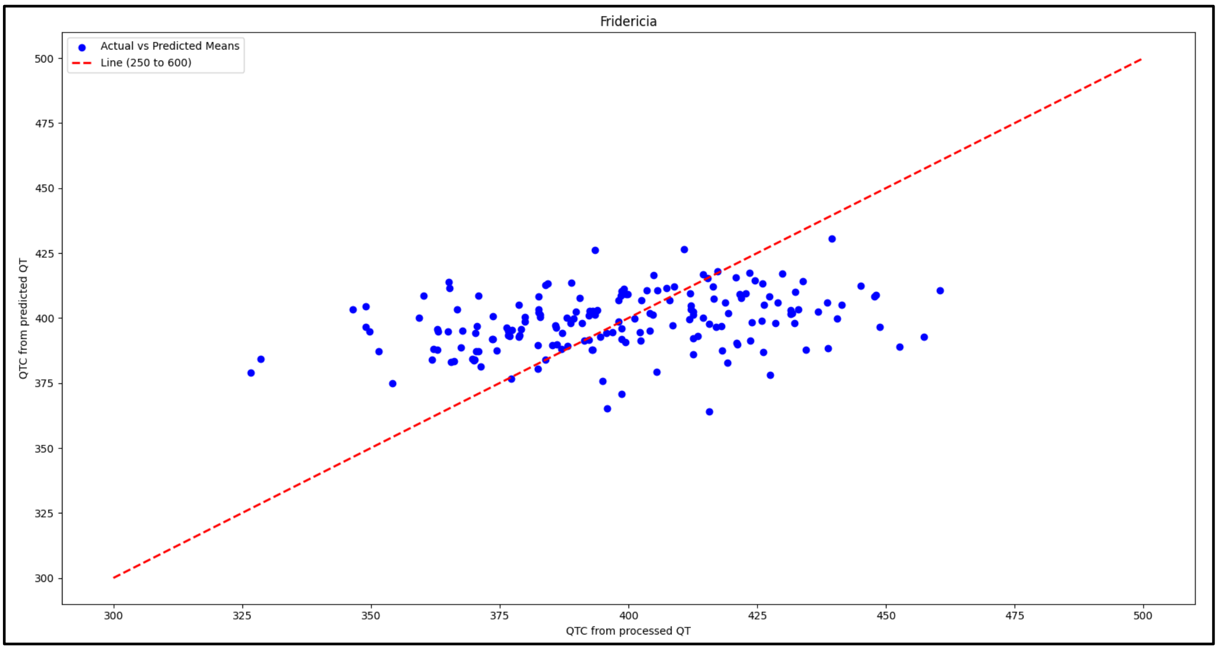

Figure 59 – Fridericia's formula

These results can be used as the measurement of QTc, that is a real application of QT, as ideally it doesn't depend on the HR and can indicate prolonged and shortened QT intervals for any HR.

6.2 Neural Network QT estimation

The processed data was passed to different neural network configurations. The Table contains the different tested configurations.

Characteristic	Tested values
Solvers	Adam, SGD
Neurons	50-15000
Activation Function	Linear, Tanh
Learning Rate	0.0001-0.1
Loss	mean_squared_error
Epochs	50-100

Table 10 – Neural Network Configurations

However, with the dataset used for the prediction a better result than using linear regression couldn't be obtained. It is logical, as for training the neural network there weren't enough different signals. The segments considered were more than 5000, nevertheless they corresponded to only 180 different signals.

The best results and the most efficient architecture are presented, but these results won't be used for now, as they don't bring a better solution than a simple linear regression.

- Architecture:

The next architecture has given the best results:

Characteristic	Tested values
Solver	Adam
Neurons	350
Activation Function	Linear
Learning Rate	0.001
Loss	mean_squared_error
Epochs	100

Table 11 – Architecture best results

➢ Results:

The next graphic was obtained for the neural network:

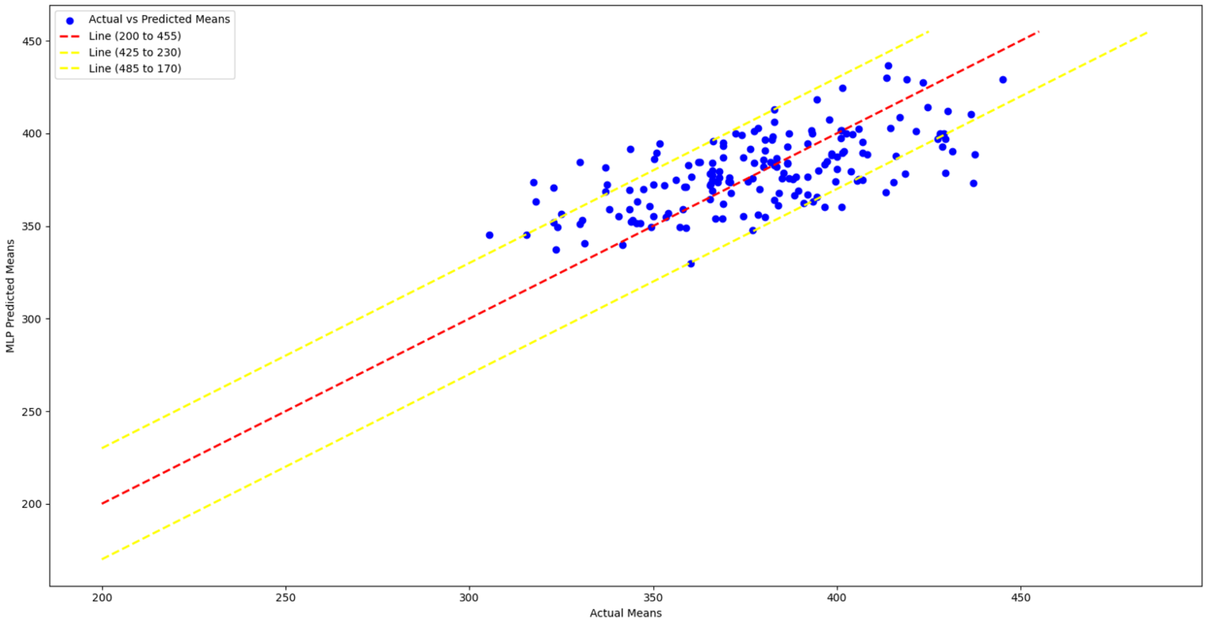

Figure 60 – Results Neural Network

And the next statistic results were obtained:

Absolute error (ε)	**18.406 ms**
Relative percentage error (ε%)	**4.859%**
Pearson's correlation coefficient	**0.6343**
p-value Wilcoxon rank-sum test	**0.9448**

Table 12 – Results for neural network

7. ECONOMICAL STUDY

The completion of this work entails various costs, associated to different tasks of the project, personal computer and MATLAB license. This article conducts a budget calculation to facilitate the execution of this project.

7.1 Calculation of unit prices

Below is the table of unit prices for materials, labor, and auxiliary elements:

Service	Rate	Total quantity (hours)	Total Cost
ECG Processing	40€/h	225h	9000€
PCG Processing	40€/h	225h	9000€
Prediction Model	40€/h	200h	8000€
Documentation	30€/h	100h	3000€
Personal Computer	500€/u	1	500€
Matlab License	900€/u	1	900€
		Subtotal	**34400€**

Table 13 – Unit Prices

7.2 Calculation of general prices

	Cost
Subtotal	**34400€**
General Costs 13%	4472€
Industrial Benefit 6%	2064€
Total without IVA	40936€
IVA	8.596,56€
Total with IVA	**49532.56€**

Table 14 – Total Budget

This budget of the Project of Retrieving heartbeat information from phonocardiogram amounts to the quantity of **forty-nine thousand five hundred thirty-two euros and fifty-six cents.**

8. RESULTS

In this chapter it will be discussed if the obtained results can be used in practice and compared with other previously obtained results.

In the first place the obtained errors will be compared with other errors that could be sour

In the previous study [14] as introduced in the State of the Art the model validation has lead to the next graphical representation of the predicted QT in reference to the actual QT value:

8.1 Graphical Comparison

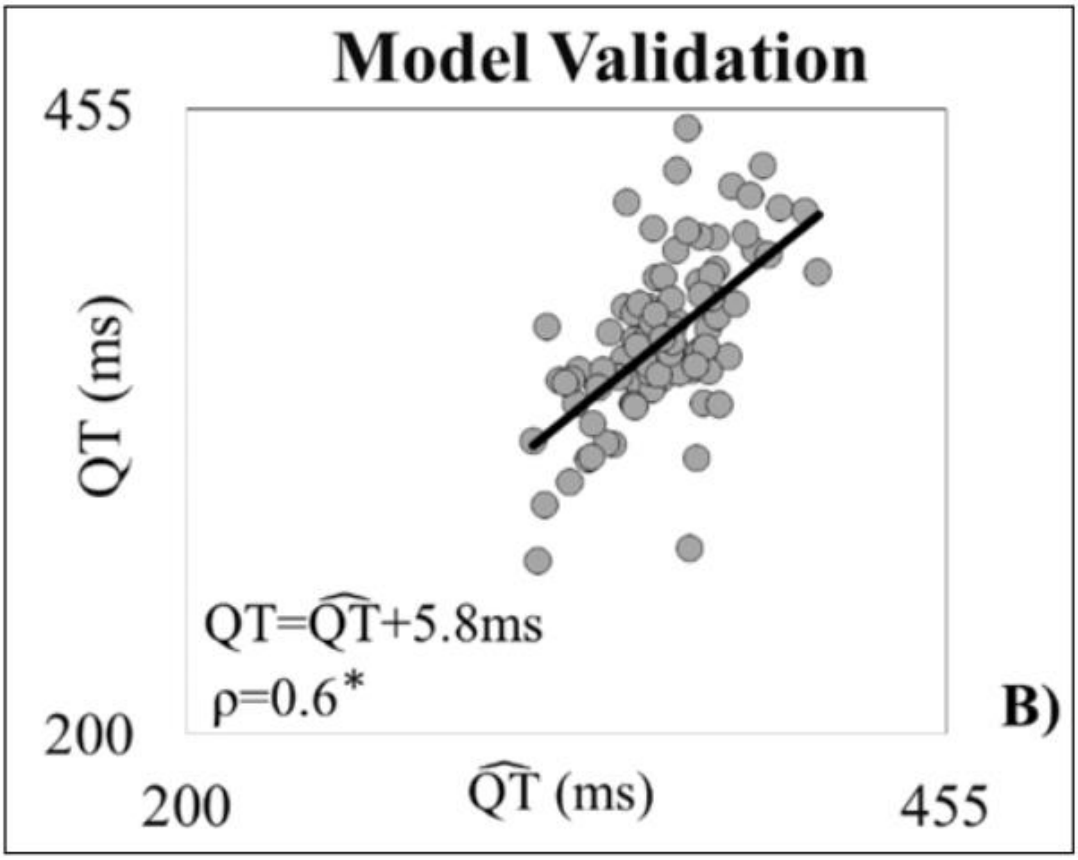

Figure 61 – Previous study graph

The linear regression from this project has led to the next graphic:

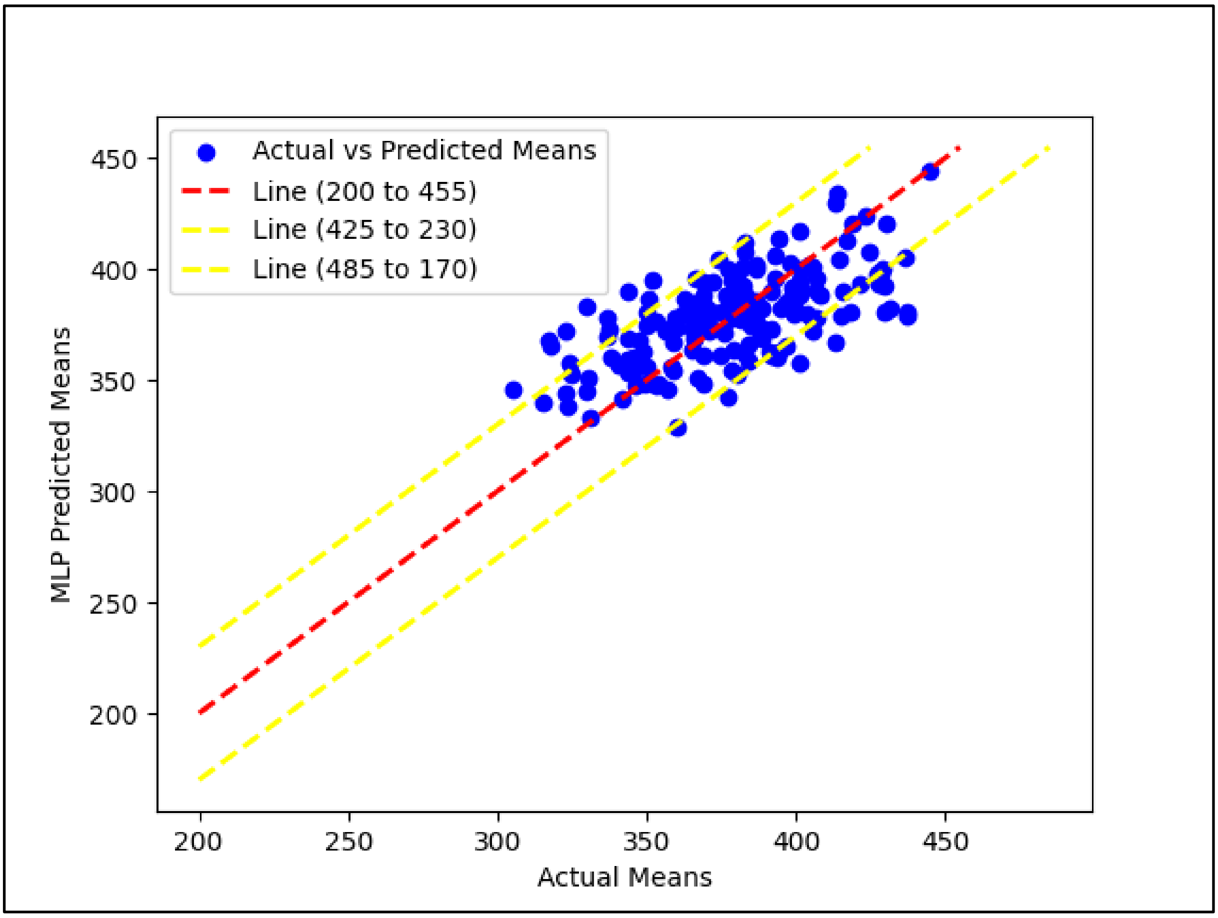

Figure 62 – Results linear regression 2

As it is possible to observe from the graphical representation, the proposed solution has less errors with high deviation from the

8.2 Statistical Comparison

The statistical comparison of the errors obtained couldn't provide reliable results, as in the previous study only the QT intervals between 345 and 377 ms were considered for healthy people. In this project the QT intervals varied from 300 to 450 ms, and in the first stage even the signals higher than 450 ms were considered. The statistical parameter of Pearson's correlation coefficient was a little bit higher for this project, reaching 0.6425 against 0.6 in the reference study.

8.3 Conclusion

The main result of this project is that a previous study made for the healthy patients can be extended to the whole variation of QT interval of healthy and not healthy patients and can provide a reliable estimation of the electrocardiographic QT interval from phonocardiographic heart sounds in healthy and not healthy subjects.

In order to make the prediction model more reliable, more synchronously obtained data from PCG and ECG of the same patient is needed. This could lead to an actually working algorithm based on neural networks. Also, the processing techniques must be improved. As the input to the prediction model more information could be considered, if the disease of the patient is classified and more features of the PCG are considered.

9. FUTURE LINES

This Master Thesis can be extended with future developments. Further, some possible improvements are introduced that could be implemented.

9.1 Testing of the Obtained Results

The obtained results have been tested on a small dataset of signals, reaching 180 different signals. However, this is not a representative dataset enough in statistical terms. As a future line it can be set to the testing of the obtained results in other data. This data needs to be recorded, processed and finally tested or used to improve the prediction models.

9.2 Use of the obtained results

The obtained results can be used as a real-world application, that will capture the PCG recording, that could be obtained through a wearable sensor, process it in real time and estimate the QT interval from the PCG recording.

This application wouldn't be 100% reliable, but can give a generic idea to the patient about his QT interval, without a need of the medical staff and complex medical equipment. In case, the risk of prolonged QT interval is detected, the application could suggest to the patient to visit a doctor and measure the QT interval to confirm the result.

9.3 Extension to other features

In this project only the QT interval was predicted, however, there may be other ECG features and diseases that could be predicted using the PCG information.

9.4 Prediction of heart diseases, using the combined ECG and PCG information

In this project sophisticated and reliable algorithms were developed to process ECG and PCG signals. This information and the features of both processed signals can be used not to predict features of one, using another recording, but to predict and classify the different heart diseases, using the combined information from ECG and PCG recordings. The diseases could be associated with the QT interval or with any other heart problem.

10. BIBLIOGRAPHY

In this chapter is exposed the full list of bibliographic references that have been used in this Master Thesis in the order of appearing in this document.

[1] National Library of Medicine. Conquering the ECG. 2004.

[2] Eko platform education. How to interpret a PCG and spot a murmur. November 30, 2022.

[3] Noemi Giordano and Marco Knaflitz. A Novel Method for Measuring the Timing of Heart Sound Components through Digital Phonocardiography. April, 2019.

[4]. Ismail S., Siddiqi I., Akram U. Localization and classification of heart beats in phonocardiography signals—A comprehensive review. EURASIP J. Adv. Signal Process. 2018.

[5]. Emmanuel B.S. A review of signal processing techniques for heart sound analysis in clinical diagnosis. J. Med. Eng. Technol. 2012.

[6]. Quiceno A.F., Delgado E., Vallverd M., Matijasevic A.M., Castellanos-Dominguez G. Effective phonocardiogram segmentation using nonlinear dynamic analysis and high-frequency decomposition; Proceedings of the Computers in Cardiology Conference. 14–17 September 2008.

[7]. Varghees V.N., Ramachandran K.I. A novel heart sound activity detection framework for automated heart sound analysis. Biomed. Signal Process. 2014.

[8]. Roy A.K., Misal A., Sinha G.R. Classification of PCG Signals: A Survey. Int. J. Comput..2014.

[9] Britannica. phonocardiography. Consulted: February, 2024.

[10] Ciprian Rezuş, Victor Dan Moga1 , Anca Ouatu, Mariana Floria. QT interval variations and mortality risk: Is there any relationship? 2015.

[11] Shen Luo PhD, Kurt Michler MSEE, Paul Johnston, Peter W. Macfarlane DSc (FRCP). A comparison of commonly used QT correction formulae: The effect of heart rate on the QTc of normal ECGs. October, 2004.

[12] R. E. Klabunde. Cardiovascular Physiology Concepts. Lippincott Williams & Wilkins, 2004.

[13] Agnese Sbrollini , Marta Beghella Bartoli , Angela Agostinelli , Micaela Morettini, Francesco Di Nardo, Sandro Fioretti , Laura Burattini. Second Heart Sound Onset to Identify T-Wave Offset. 2017.

[14] Agnese Sbrollini, Micaela Morettini, Ilaria Marcantoni, Laura Burattini. Model-Based Estimation of Electrocardiographic QT Interval From Phonocardiographic Heart Sounds in Healthy Subjects. 2020.

[15] Amazon. What is a neural network? Consulted: February, 2024.

[16] Chengyu Liu , David Springer , Benjamin Moody , Ikaro Silva , Alistair Johnson , Maryam Samieinasab , Reza Sameni , Roger Mark , Gari D. Clifford . Classification of Heart Sound Recordings: The PhysioNet/Computing in Cardiology Challenge 2016. March 4, 2016

[17] PhysioNet WFDB Toolbox for MATLAB and Octave. November 3, 2017.

[18] MathWorks. Matlab. Consulted: February, 2024.

[19] Python Software Foundation . Python. Consulted: February, 2024.

[20] Keras About Keras 3. Consulted: February, 2024.

[21] Nasser Mourad. ECG denoising algorithm based on group sparsity and singular spectrum analysis. April 2019.

[22] Noemi Giordano and Marco Knaflitz. A Novel Method for Measuring the Timing of Heart Sound Components through Digital Phonocardiography. April 19, 2019

[23] Jason Brownlee PhD. LOOCV for Evaluating Machine Learning Algorithms. August 26, 2020.

[24] IBM. What is linear regression? Consulted: February, 2024.