UNIVERSITAT POLITÈCNICA DE VALÈNCIA

POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

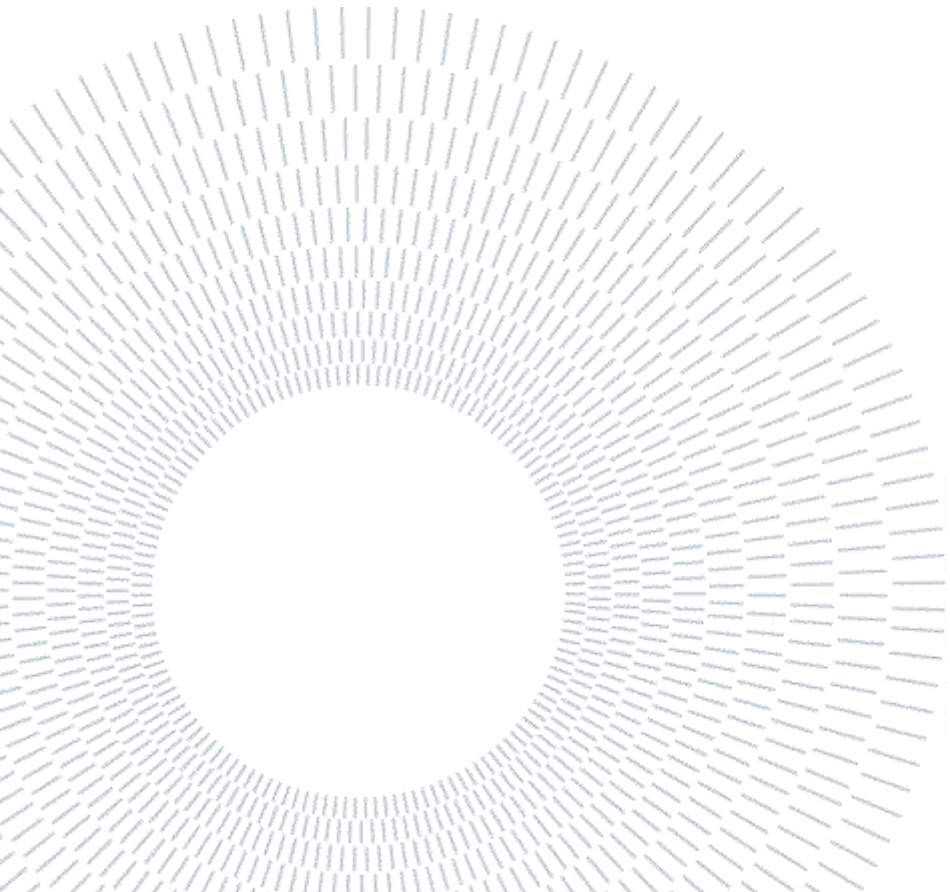
Development of an Artificial Intelligence model for COVID-19 detection based on electrocardiogram features.

BACHELOR'S THESIS IN BIOMEDICAL ENGINEERING

Author: **Juan Moltó Miró**

Student ID: 231641
Advisor: Luca Mainardi
Co-advisor: Guadalupe Garcia Isla
UPV Advisor: Carlos Sáez Silvestre

Academic Year: 2022-2023



Abstract

The objectives of this project are twofold. Firstly, to identify key morphological features in 12-lead electrocardiogram (ECG) signals that can help identify patients infected with the SARS-CoV-2 disease at higher risk of mortality. Secondly, to develop a machine learning model that utilizes these for Coronavirus Disease 2019 (COVID-19) detection.

The development of this work involve extracting features that represent the main structures of the ECG, such as the P-wave, QRS-complex, and T-wave. Open-access toolboxes in MATLAB are used for this purpose. Furthermore, feature cleaning, preprocessing, selection, and model training are carried out using Python. We use short 12-lead ECG signals obtained from COVID-19 patients at a hospital in the Italian city of Cremona (Ospedale di Cremona). The developed model will allow early detection of the COVID-19 virus using the non-invasive clinical procedure of ECG.

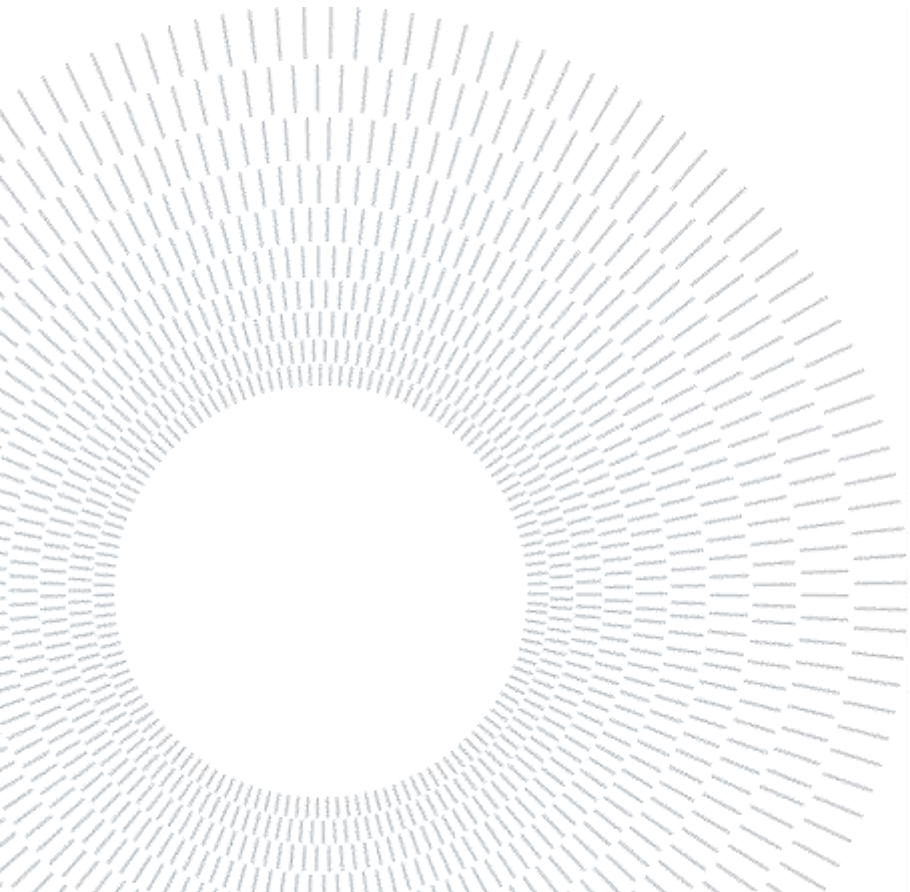
Key-words: Electrocardiogram, Features, COVID-19, detection, model, Machine Learning

Abstract in lingua italiana

Gli obiettivi di questo progetto sono due. In primo luogo, identificare le caratteristiche morfologiche chiave nei segnali dell'elettrocardiogramma a 12 derivazioni (ECG) che possono aiutare a identificare i pazienti infettati dalla malattia SARS-CoV-2 a più alto rischio di mortalità. In secondo luogo, sviluppare un modello di apprendimento automatico che le utilizzi per il rilevamento della malattia da Coronavirus 2019 (COVID-19).

Lo sviluppo di questo lavoro prevede l'estrazione di caratteristiche che rappresentino le strutture principali dell'ECG, come l'onda P, il complesso QRS e l'onda T. A tale scopo sono stati utilizzati toolbox ad accesso libero in MATLAB. Inoltre, la pulizia delle caratteristiche, la preelaborazione, la selezione e l'addestramento del modello vengono eseguiti utilizzando Python. Sono stati utilizzati brevi segnali ECG a 12 derivazioni ottenuti da pazienti COVID-19 presso un ospedale della città italiana di Cremona (Ospedale di Cremona). Il modello sviluppato consentirà di rilevare precocemente il virus COVID-19 utilizzando la procedura clinica non invasiva dell'ECG.

Parole chiave: Elettrocardiogramma, caratteristiche, COVID-19, rilevamento, modello, Machine Learning.



Contents

1. Introduction

1.1 Aim of the work and motivation

The COVID-19 pandemic that began in 2020 has had an unprecedented global impact, affecting millions of people worldwide. The rapid spread of this highly contagious virus has generated an urgent demand for efficient and accurate detection methods to identify and diagnose cases of COVID-19 infection. In this context, early and accurate detection has become a top priority to control the spread of the virus and provide adequate medical care to affected patients.

In recent years, artificial intelligence (AI) has demonstrated enormous potential in various areas of medicine, including disease diagnosis. In particular, the analysis of electrocardiogram (ECG) features has revealed a valuable source of information for the detection of heart disease and other medical conditions. In addition, it has been observed that COVID-19 can have significant impacts on the cardiovascular system of patients, which has led to research seeking to identify characteristic patterns in the ECGs of those affected by the virus.[1]

The motivation behind this thesis work lies in the need to harness the potential of artificial intelligence and ECG characteristics to develop an effective COVID-19 detection model. By combining these two fields, we can open new possibilities in the fight against the pandemic, providing a valuable tool for healthcare professionals in the early and accurate identification of COVID-19 cases.

The main objectives of this work are multiple. First, we seek to identify the specific features in ECGs that are most relevant for COVID-19 detection, which may involve extracting specific features and selecting those that provide the most discriminatory information. This will provide a better understanding of how the virus affects the cardiovascular system and contribute to the existing scientific knowledge on the cardiovascular manifestations of COVID-19.

In addition, we seek to develop an artificial intelligence model based on machine learning and signal processing techniques that can analyze ECG features and accurately and reliably classify patterns associated with COVID-19 infection. This involves the collection of a large amount of ECG data from patients with both confirmed COVID-19 and healthy subjects, which will allow training and validation of the proposed model. In our case, the ECG database comes from the Hospital of Cremona (Ospedale di Cremona).

Another key objective is to evaluate the effectiveness and accuracy of the model, and this involves performing comprehensive and comparative tests using independent datasets and validating the ability of the model to accurately generalize and detect COVID-19 cases in different clinical settings.

In summary, this thesis work mainly aims to develop an artificial intelligence model based on ECG features for COVID-19 detection. It seeks to harness the potential of artificial intelligence and signal analysis to provide an accurate and reliable tool to aid in the early and effective detection of COVID-19 infection. By achieving these goals, it is expected to contribute to the advancement of medicine and improve the response to the pandemic, providing a valuable tool for healthcare professionals and enabling faster and more accurate medical care for patients affected by the virus.

1.2 COVID – 19

COVID-19 disease, also known as coronavirus disease, is an infection caused by the SARS-CoV-2 virus, which belongs to the coronavirus family. It was first identified in December 2019 in the city of Wuhan, China, and has since spread rapidly globally, becoming a pandemic.

It spreads primarily through respiratory droplets generated when an infected person coughs, sneezes, speaks, or breathes. It can also be transmitted by touching contaminated surfaces and direct hand-to-mouth or hand-to-nose contact, although this is not the primary mode of spread. The disease has a wide range of symptoms, from asymptomatic cases to mild, moderate or severe symptoms. [2]

According to several scientific studies, the most common symptoms of COVID-19 include fever, dry cough, shortness of breath, fatigue, sore throat, nasal congestion, loss of taste and smell, muscle aches and headaches. However, it has been observed that some patients may experience less frequent symptoms such as diarrhea, conjunctivitis, skin rashes, dizziness, and even neurological problems. [3]

The severity of the disease varies widely, and it has been observed that older people and those with underlying chronic diseases, such as diabetes, heart disease and lung disease, are at a higher risk of developing more severe symptoms and complications. Pneumonia is one of the most common complications associated with COVID-19 and can require hospitalization and intensive care.

According to a study published in Nature Medicine in April 2020, about 14% of infected patients had severe disease, and approximately 5% required intensive care. In addition, the estimated overall case fatality rate of COVID-19 at that time was 2.3%. [4]

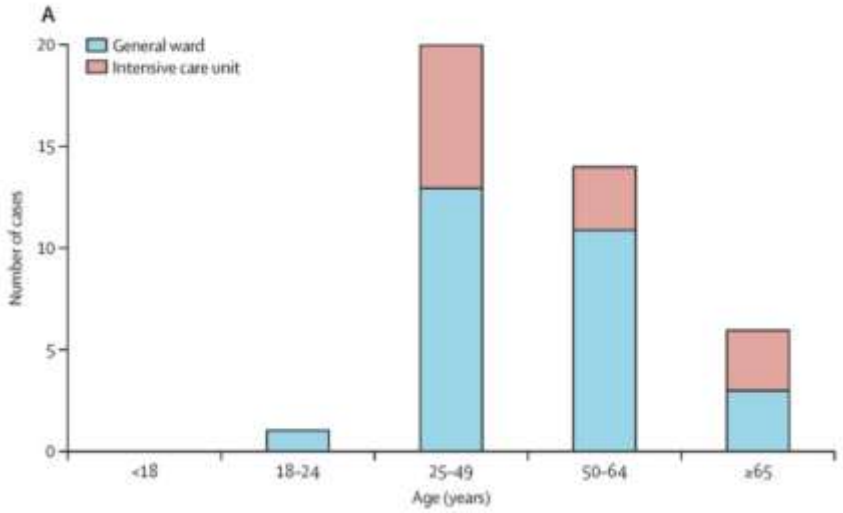


Figure 1: Age distribution of patients with laboratory-confirmed 2019-nCoV infection[5]

In terms of treatment, a number of drugs and therapies have been used to alleviate symptoms and improve patient outcomes. These include antivirals, corticosteroids and monoclonal antibodies. In addition, a variety of effective vaccines against the virus have been developed and distributed worldwide, which has helped to reduce the severity of cases and reduce the spread of the virus.

Finally, it is important to know that scientific information on COVID-19 is constantly evolving, and new studies and research are continually being conducted to better understand the disease, its spread and its effects.

1.3 ECG

An ECG (Electrocardiogram) is a diagnostic test that records the electrical activity of the heart over time. It is an essential tool for the diagnosis and monitoring of various cardiac conditions. The procedure is performed by placing electrodes on the patient's skin that capture the electrical signals generated by the heart.[6]

A complete ECG is composed of several leads, which are different views or perspectives of the heart's electrical activity. The leads are grouped into three main groups:

- Bipolar leads: these leads record the difference in electrical potential between two points. There are three leads: I, II and III.
    - Lead I: It records the potential difference between the left arm and the right arm.
    - Lead II: It records the potential difference between the left arm and the left leg.
    - Lead III: Records the potential difference between the right arm and the left leg.
- Unipolar leads: These leads use a central positive electrode and a combined negative electrode from all other extremities. There are three leads: AVR, AVL and AVF.
    - AVR lead: Records the potential difference between the right arm and the central point formed by the junction of the left and right arms.
    - AVL lead: It records the potential difference between the left arm and the central point.
    - AVF lead: records the potential difference between the left leg and the central point.
- Precordial leads: These leads are placed in the chest, just above the heart, and record electrical activity from different angles. There are six leads: V1, V2, V3, V4, V5 and V6.[7]

Each of these leads records the electrical waves of the heart. The most important waves in an ECG are:

- P wave: represents the depolarization of the atria (atrial contraction).
- QRS complex: represents depolarization of the ventricles (ventricular contraction).
- T wave: represents repolarization of the ventricles (ventricular relaxation).

In addition to these, other minor waves and complexes can be observed, such as the U wave, but those mentioned above are the most significant for diagnosis. Each of these

waves and complexes in the ECG provides valuable information about the electrical function and overall health of the heart.[6]

Finally, a concept related to the ECG that will appear in this work is the vectorcardiogram.

It is a graphical recording that represents the electrical activity of the heart in relation to time, magnitude and direction. It is very valuable in specific clinical cases where a more detailed evaluation of cardiac electrical activity is required.[8]

It has 3 main components: QRS complex, T wave and U wave.

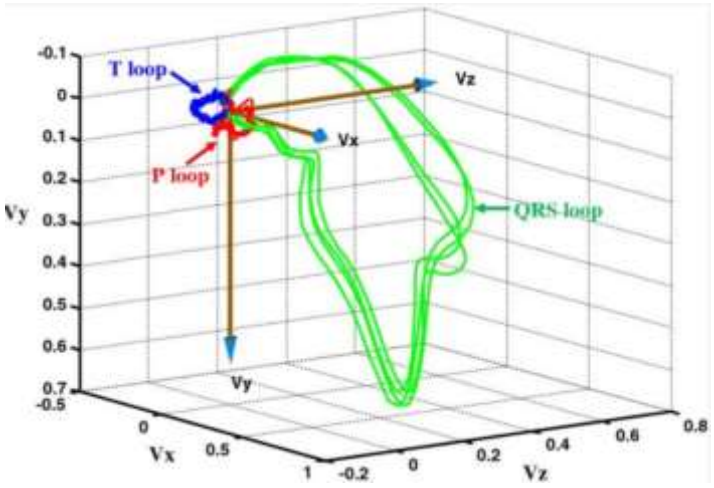


Figure 2: Example of vectorcardiogram[9]

1.4 Chapter structure

After having made the theoretical introduction and the objectives and motivations of the work, the structure that will be followed now will be the following:

First, the works similar to this one carried out recently in the 'State of the art' section will be analyzed.

Subsequently, in the 'Materials and Methods' section, the process followed to carry out the project will be explained, as well as the tools used and the necessary calculations.

This is followed by the 'Results and discussion' section, where we will be able to observe the results obtained and these values will be interpreted and compared.

Finally, in the 'Conclusions' section, different conclusions will be drawn from the results obtained and the future perspectives and limitations of the work.

2. State of the art and context

This chapter will analyze the state of the art, drawing results and conclusions from studies carried out on the same subject.

2.1 General description of the project context

As mentioned in the Introduction section, SARS-CoV-2 virus can directly affect the cardiovascular system and cause cardiovascular complications in infected patients.

Several studies have shown that COVID-19 can cause damage to cardiac tissue and trigger inflammatory and systemic responses that affect cardiovascular function. The virus can infect cells lining blood vessels and heart tissues, leading to inflammation and dysregulated immune response. This can result in damage to blood vessels, the formation of blood clots and cardiac dysfunction.

There are cases where serious cardiovascular complications have been reported, such as myocarditis (inflammation of the heart muscle), arterial and venous thrombosis, and acute damage to the heart muscle. These complications can lead to an increased risk of heart failure, stroke, and other adverse cardiovascular events.[10]

This involvement of the SARS-CoV-2 virus in the cardiovascular system is reflected in alterations in the electrocardiogram (ECG). Therefore, in this work we will seek to detect ECG features, and through Machine Learning, create a model capable of performing a COVID detection based on these features obtained.[1]

2.2 Projects related to the subject of this thesis

In recent years, numerous projects have been carried out in which ECG characteristics have been analyzed in order to draw conclusions and detect COVID-19. What has been observed in these studies is that the most common ECG alterations in patients with COVID-19 are: sinus tachycardia, prolonged QT interval, right bundle branch block, left bundle branch block, ST segment alterations, T wave changes[1], [10]–[12].

All this reflects the importance of monitoring and analyzing ECG changes during hospitalization of patients with COVID-19.

In addition, many of these studies have been performed using Machine Learning and Deep Learning techniques. In these studies, as in our case, a database of patients diagnosed with COVID-19 and another database with healthy patients (control) have been used, ECG feature extraction has been performed and models have been created capable of processing ECGs completely, from input to output, without requiring intermediate manual processing steps.[13]–[15]

In all of them good results have been achieved in terms of accuracy in ECG-based detection of COVID-19, however not so high as to replace diagnostic tests.

2.3 Conclusions about the state of the art

What can be concluded from all the studies observed is that Artificial Intelligence models are a very useful tool to support the diagnostic tests performed by the medical staff, and can also help to predict how the patient's evolution will be. However, these models still have limitations, so they are a support tool, but not the only tool that should be used.

3. Materials and Methods

This section will explain the methods followed, the tools used and the process carried out from the beginning until the results are obtained.

3.1 General pipeline

First of all, in this section we will be able to observe in a general way the steps to be followed to carry out the project, which will be explained in more detail later on.

The first step to follow is data acquisition, how the data have been obtained and how they have been stored and prepared.

Secondly, feature extraction is performed, where the algorithm used, lead synchronization and the functions of interest that have been used in the code will be explained.

Then these data will be processed, where splitting, feature preprocessing and undersampling will be performed.

Finally, the model will be trained, validated, tested and selected according to the parameters obtained.

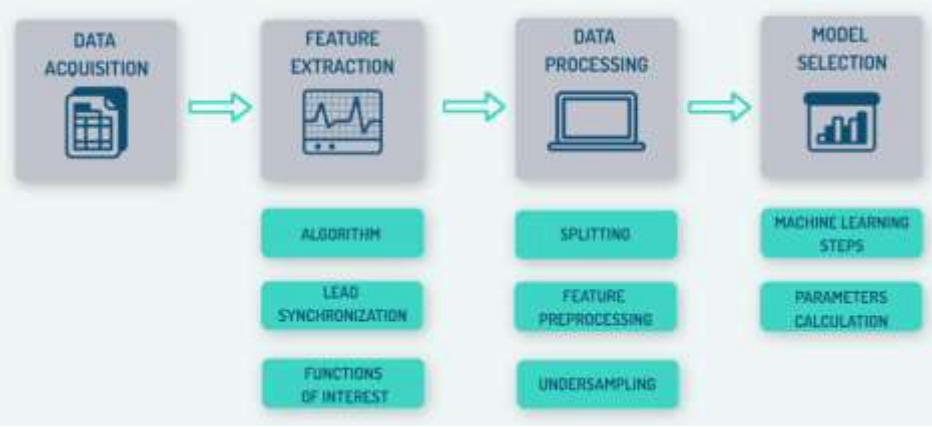


Figure 3: General pipeline followed in the project

3.2 Data acquisition

The first step, as mentioned above, is data acquisition.

I was provided with 2 databases: a database of patients diagnosed with COVID-19 and another of healthy patients, as a control group.

At first, the number of patients with COVID-19 was 295 and the number of healthy patients was 5615. However, due to missing data errors there are finally 286 COVID-19 patients and 5439 healthy patients, unbalanced data in terms of proportion.

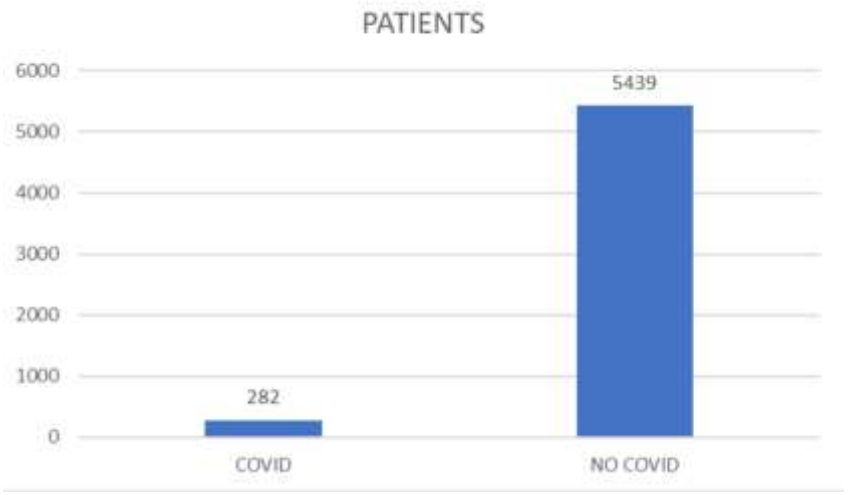


Figure 4: Graphic of number of healthy and COVID-19 patients

A graph in the Figure 4 shows that the number of healthy patients is notably higher than the number of COVID-19 patients.

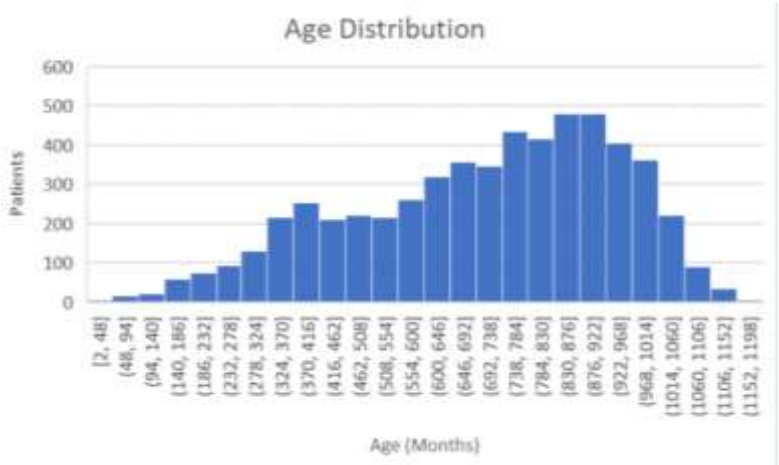


Figure 5: Graphic of age distribution of patients

Figure 5 shows a graph extracted from the data, where we can observe the age distribution. It can be seen that the age range with the highest number of patients is between 830 and 920 months, which is equivalent to between 69 and 76 years, elderly people.

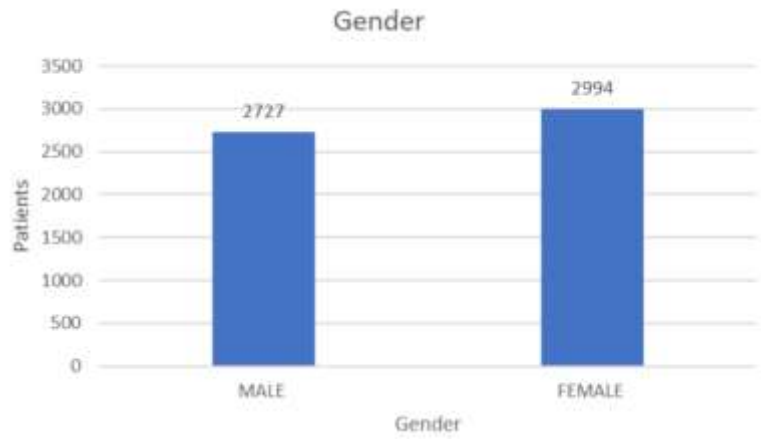


Figure 6: Graphic of number of male and female patients

On the other hand, in this graph in the Figure 6 we can observe a fairly balanced proportion between men and women.

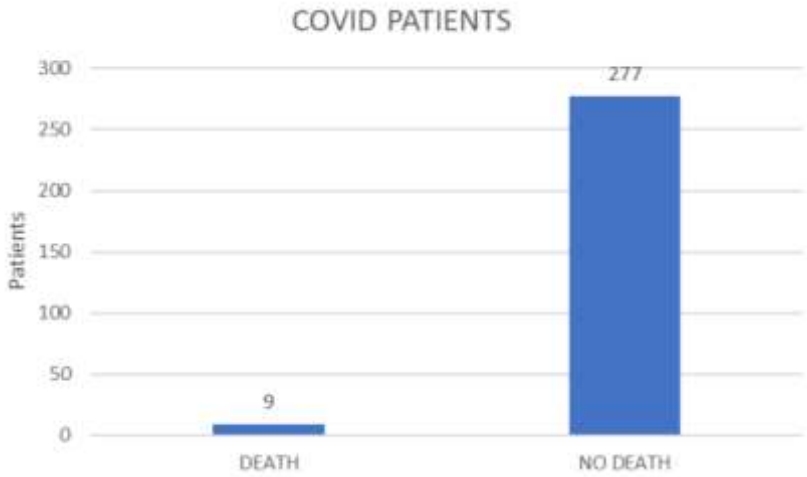


Figure 7: Graphic of death and alive patients

Finally, with respect to patients with COVID-19, the Figure 7 shows a graph showing the few data available on deceased patients.

3.3 Feature extraction

Once it has been explained how the data have been acquired, it will be now explained how the extraction of characteristics is done.

3.3.1 Algorithm and programming environment

The programming environment used for this project is MATLAB, with the help of two toolboxes: "ECG-kit Toolbox" and "PhysioNet Cardiovascular Signal Toolbox".

Our main input were the12 ECG leads unsynchronized and also clinical variables such as age and sex.

Regarding the steps that compose the algorithm, they are the following:

1. Peak R detection with Pan-Tompkins algorithm
2. Alignment of ECG leads and data matrix construction
3. Median filter applyment to remove baseline wander (ECG12filt)
4. Kors transformation to convert the 12 leads to XYZ leads
5. Fiducial points detection using ECG-kit toolbox
6. Features extraction using the XYZ leads and the detected fiducial points (GEH_analysis_git)

3.3.2 Leads synchronization

The problem I had to face was that the leads were not synchronized, since the standard leads (I, II, III, aVR, aVL and aVF) started at instant 0, while the precordial leads (V1, V2, V3, V4, V5 and V6) did not start at 0.

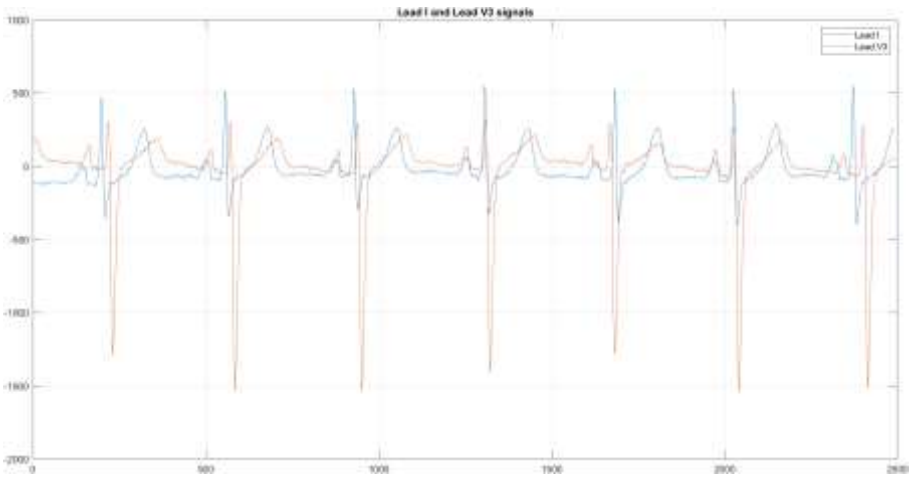


Figure 8: Plot of Lead I and Lead V3 of an ECG of the dataset

In Figure 8, we can observe a plot of Lead I and Lead V3, where we can see that the peaks are not synchronized.

To do the feature extraction we need them to be synchronized, therefore the following solution has been proposed:

1. Detect 1st Peak R with Pan-Tompkins algorythm

2. Set that first peak R of each lead as the first value
3. Homogenize data by eliminating by cutting off the beginning or the end of excess data

In this way, our result will be a set of 12 leads, now synchronized and ready to apply the algorithm and extract the features.

3.3.3 Functions of interest

The main functions of interest used in the code are:

- **Kors_git (Global-Electrical-Heterogeneity)**

  This function is in charge of the projection of 12 leads into XYZ space using Kors transformation coefficients. Besides, it is in charge of projection normalization, and finally, the vectorcardiogram obtention.

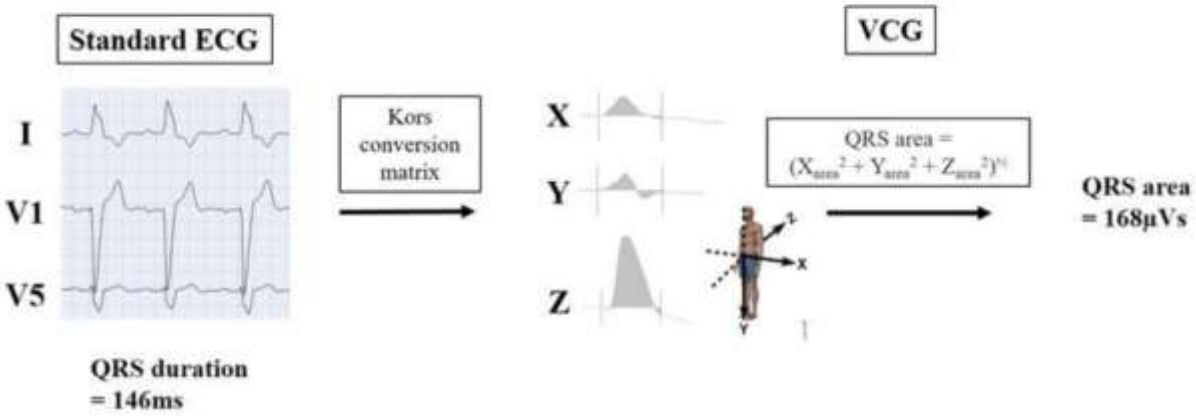


Figure 9: Kors transformation and vectorcardiogram obtention [16]

- **wavedet_3D_ECGKit (ECG-kit toolbox)**

In this other function of interest, belonging to the ECG-kit toolbox, time of characteristic points in ECG (Fiducial points) is returned. These points are used to measure intervals, amplitudes and other parameters in the ECG.

- **GEH_analysis_git (features extraction)**

Finally, this function uses the vectorcardiogram obtained in the "Kors_git" function and the fiducial points obtained in the "wavedet_3D_ECGKit" function to perform various calculations related to vector and angle analysis, including QRS and T vector integration, angle and magnitude calculations, and elevation and azimuth calculations.

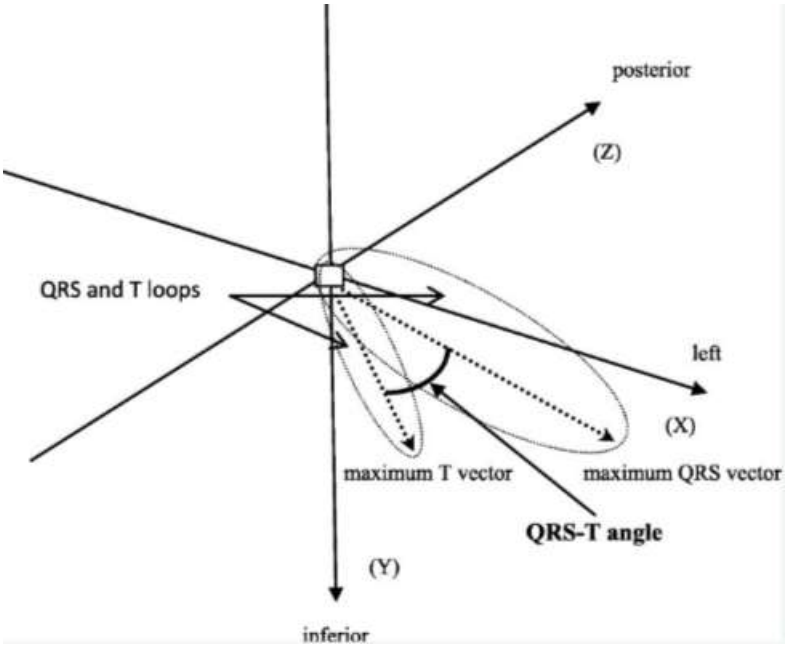


Figure 10: Example of vectorcardiogram used in the function [17]

Thanks to this function we obtain a total of 22 features, which added to the patient's age and sex, make a total of 24 features, which are as follows:

1. Age: Age of patient in months
2. Sex: M(0) , F (1)
3. QRST Peak Angle (QRSTang): Angle between the R and T vectors in degrees.
4. QRST Mean Angle (QRSTang_M): Angle between the QRS and T vectors in degrees.
5. QRS Peak Azimuth (AZ_OQ): Azimuth of the R vector in degrees.
6. QRS Mean Azimuth (AZ_OQM): Mean azimuth of the QRS vector in degrees.
7. T Peak Azimuth (AZ_OT): Azimuth of the T vector in degrees.

8. T Mean Azimuth (AZ_OTM): Mean azimuth of the T vector in degrees.
9. SVG Peak Azimuth (AZ_SVG): Azimuth of the SVG vector (sum of R and T vectors) in degrees.
10. SVG Mean Azimuth (AZ_SVG_M): Mean azimuth of the SVG vector in degrees.
11. QRS Peak Elevation (EL_OQ): Elevation of the R vector in degrees
12. QRS Mean Elevation (EL_OQM): Mean elevation of the QRS vector in degrees
13. T Peak Elevation (EL_OT): Elevation of the T vector in degrees.
14. T Mean Elevation (EL_OTM): Mean elevation of the T vector in degrees.
15. SVG Peak Elevation (EL_SVG): Elevation of the SVG vector in degrees
16. SVG Mean Elevation (EL_SVG_M): Mean elevation of the SVG vector in degrees.
17. QRS Peak Magnitude (QRS_Mag): Magnitude of the R vector.
18. QRS Mean Magnitude (QRS_Mag_M): Mean magnitude of the QRS vector.
19. T Peak Magnitude (T_Mag): Magnitude of the T vector.
20. T Mean Magnitude (T_Mag_M): Mean magnitude of the T vector.
21. SVG Peak Magnitude (SVG_Mag): Magnitude of the SVG vector.
22. QT Interval (QT_interval): Duration of the QT interval in milliseconds.
23. Area Under the Curve of Vector Magnitude (AUC_VM_QT): Area under the curve of the vector magnitude from the Q point to the T point.
24. Wilson Ventricular Gradient (WVG): Wilson ventricular gradient using the QRS and T vectors.

3.4 Data processing

Once the feature extraction has been done, we go into data processing.

3.4.1 Data splitting

The first step is to split the data into training set (80%) and testing set (20%). Then we will divide the training data into two other groups, train (80%) and evaluation (20%).

Figure 11: Scheme of the splitting carried out

3.4.2 Feature preprocessing

Once we have the data separated, we do the preprocessing of features. It is important to know that any transformation performed in the training must also be performed in the test.

The first step is to find missing data: Eliminate patients and features with more than 50% of missing values.

The second step is to do the outliers treatment: Substitute outliers for missing values and then impute a value for those missing values, using the mean. Values that were higher than the upper limit or lower than the lower limit were taken as outliers. The upper and lower limits were calculated as follows:

$$Upper\ Limit = Q75 + 1{,}5 \times IQR \quad (1)$$

$$Lower\ Limit = Q25 - 1{,}5 \times IQR \quad (2)$$

$$IQR = Q75 - Q25 \quad (3)$$

Finally, the last step of this part is the Z-Score normalization: to normalize every value in a dataset such that the mean of all of the values is 0 and the standard deviation is 1.

The way in which we apply this is with the following formula:

$$x_{scaled} = \frac{x - mean}{sd} \quad (4)$$

3.4.3 Undersampling

Finally, as mentioned before, we have a problem to perform Machine Learning, the number of healthy patients is much higher than the number of COVID-19 patients as we can see at the Figure 4, and two possible solutions have been proposed:

The first possible solution is to do an Undersampling 1-1: Same number of patients for each class, deleting extra data

The other possible solution is to do an Undersampling 2-1 and class weighting: Twice as many instances of the dominant class and weighting, adjusting the misclassification cost to penalize errors from the less predominant class twice as much.

3.5 Model selection

Finally, to select the model that gives us the results, we do the following steps:

1. Model training using all models available.
2. Model evaluation with cross-validation.
3. Selection of the best performing model taking into account different parameters: accuracy, sensitivity, specificity, Positive Predicted Value (PPV) and F1-Score.
4. Hyperparameter tunning of that model, using Bayesian Optimization
5. Model testing: where we will obtain the results and the performing of the model

The form of the confusion matrix obtained is as follows:

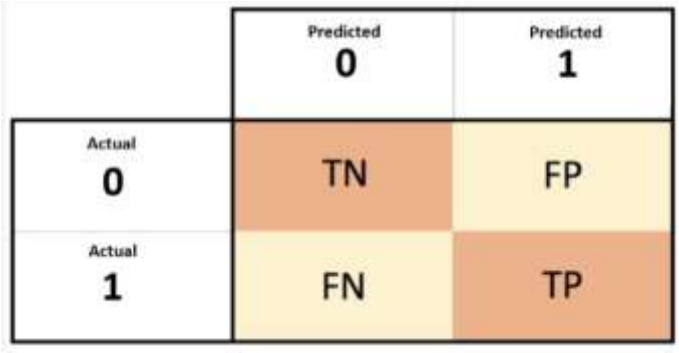


Figure 12: Confusion Matrix example

And the way that we obtain the parameters is with the following formulas:

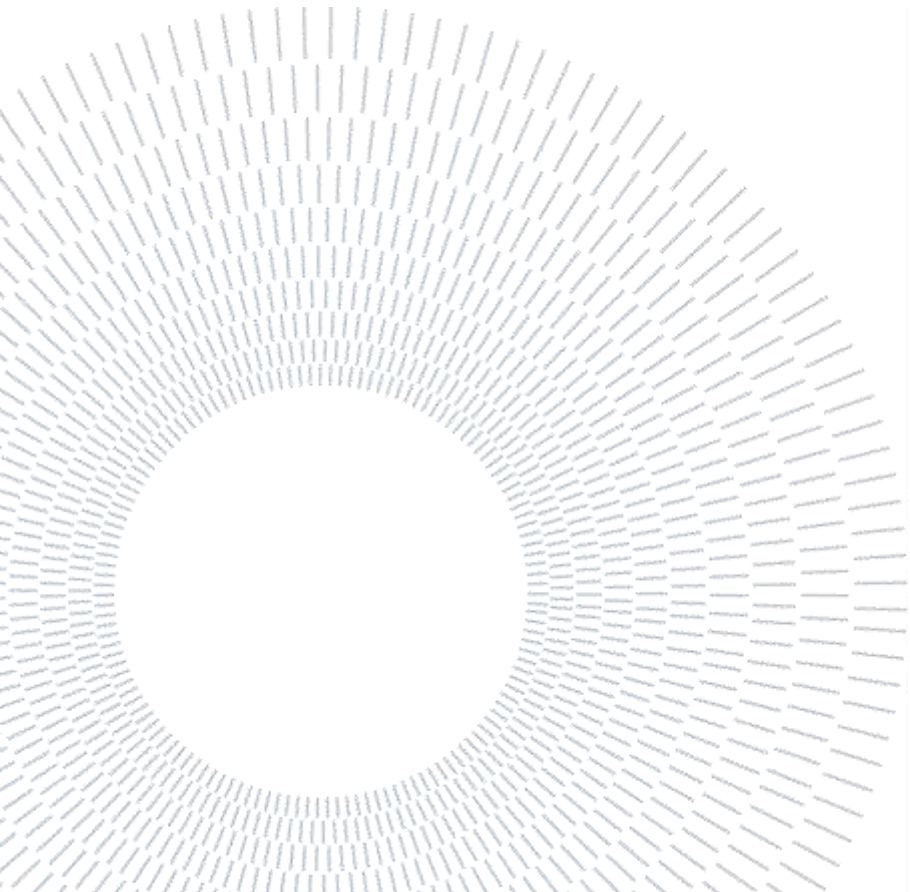
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \text{ (5)}$$

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \text{ (6)}$$

$$Specificity = \frac{TN}{TN+FP} \times 100 \text{ (7)}$$

$$Positive\ Predicted\ Value\ (PPV) = \frac{TP}{TP+FP} \times 100 \text{ (8)}$$

$$F1 - score = 2 \times \frac{Accuracy \times Sensitivity}{Accuracy + Sensitivity} \text{ (9)}$$



4. Results and discussion

In this section, the results obtained will be observed and discussed.

First, we will observe the performance in the train and validation of the models without undersampling, with undersampling 1-1 and with undersampling 2-1 and class weighting.

Subsequently, the performance of the selected models in the test will be observed.

4.1 Training and validation

4.1.1 Without undersampling

First, training and validation were performed without undersampling. The following results were obtained:

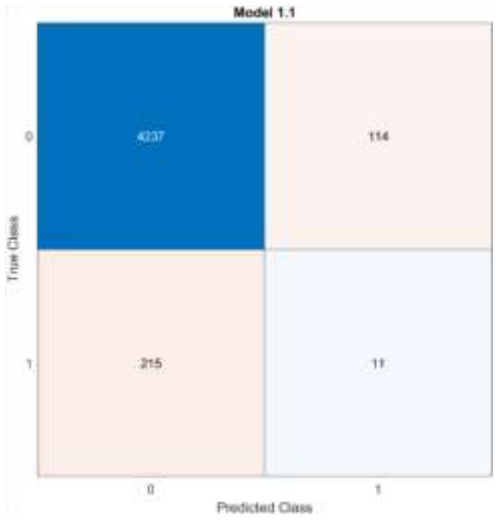


Figure 13: Fine Tree confusion matrix. Accuracy (Validation 92,8%)

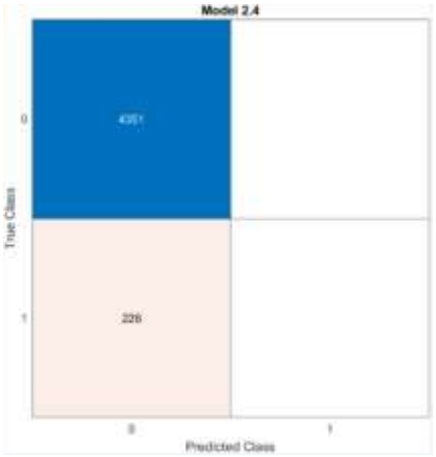


Figure 14: Logistic Regression confusion matrix. Accuracy (Validation): 95,2%

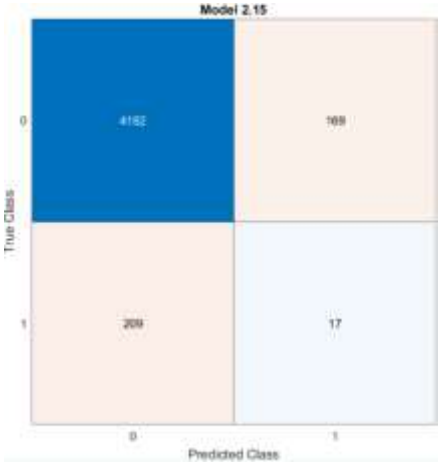


Figure 15: Fine KNN confusion matrix. Accuracy (Validation): 95,2%

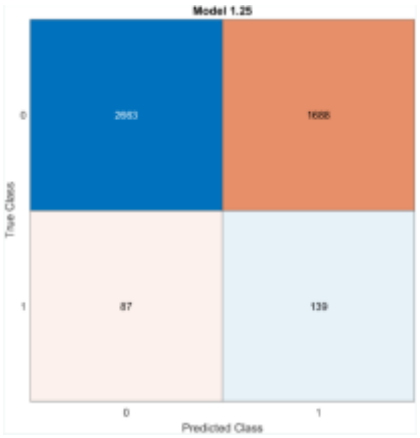


Figure 16: RUSBoosted Trees confusion matrix. Accuracy (Validation): 61,2%

As can be seen in the confusion matrices obtained, the results were not good because, despite having good precision results, this is due to the fact that the number of healthy patients is much higher. In fact, the sensitivity values are close to 0.

Therefore, it was decided to start with undersampling solutions.

4.1.2 Undersampling 1-1

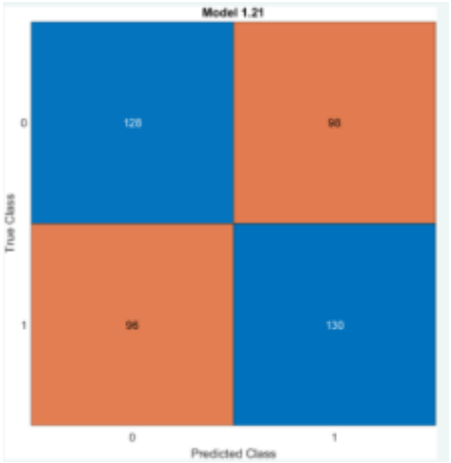


Figure 17: Boosted Tree confusion matrix. Accuracy (Validation): 57,1%

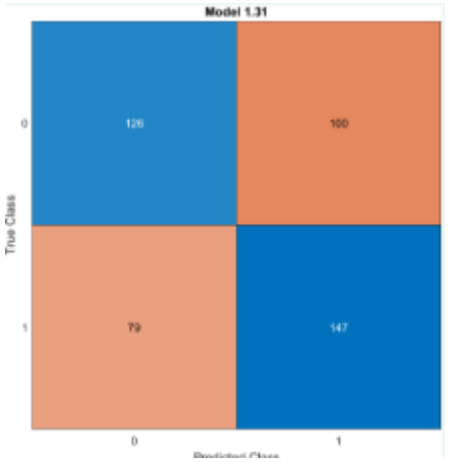


Figure 18: SVM Kernel confusion matrix. Accuracy (Validation): 60,4%

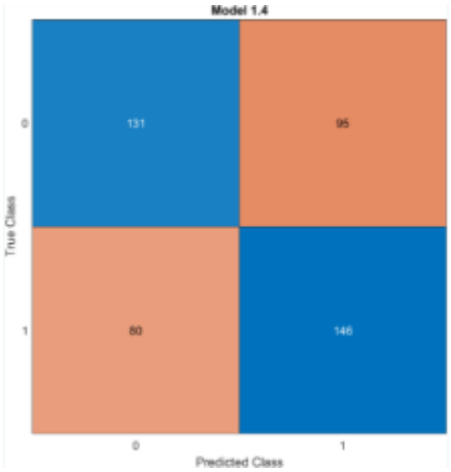


Figure 19: Logistic Regression confusion matrix. Accuracy (Validation): 61,7%

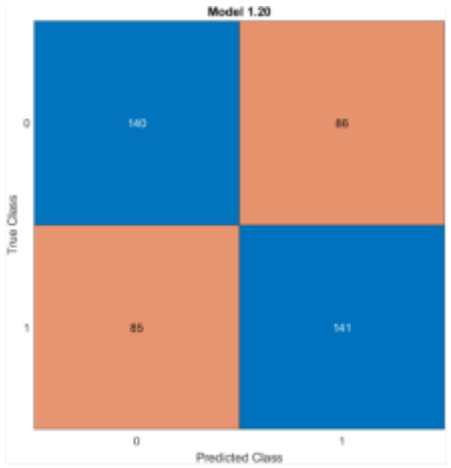


Figure 20: Weighted KNN confusion matrix. Accuracy (Validation): 62,2 %

As can be seen, the model with the best results is the Weighted KNN (Figure 21), with an accuracy of 62,2%. Therefore, it has been decided to apply the Hyperparameter Tunning to this model, applying the Bayesian Optimization.

The results obtained are as follows:

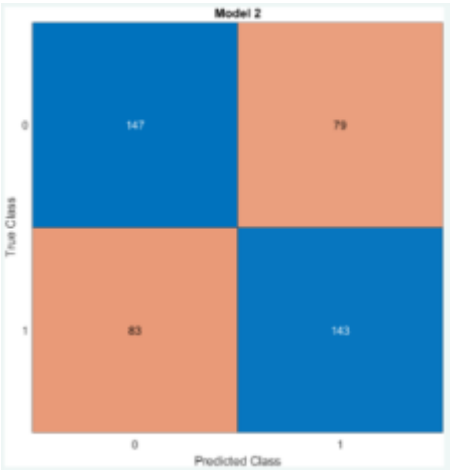


Figure 21: Weighted KNN Optimized confusion matrix. Accuracy (Validation): 64,2 %

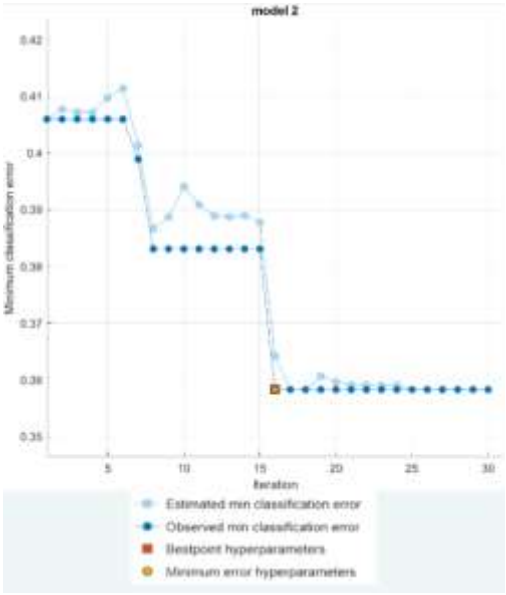


Figure 22: Minimum Classification Error of Weighted KNN Model Optimized

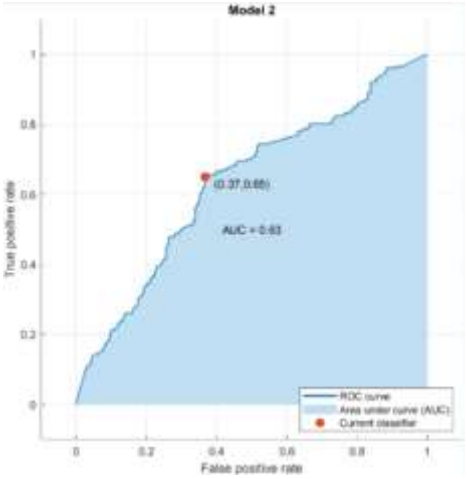


Figure 23: Weighted KNN Model Optimized ROC Curve

Figure 21 shows that the data obtained in the confusion matrix have improved, as expected, from an accuracy of 62.2% to 64.2%.

The Minimum Classification Error plot (Figure 22) shows that the observed error is lower than the estimated error, a positive indicator.

Besides, in the ROC curve (Figure 23) we observe that the Area Under the Curve, which is the ratio between the True Positive Rate (TPR) and the False Positive Rate (FPR) is 0.63. For optimal performance we should be close to 1.

4.1.3 Undersampling 2-1 and class weighting

On the other hand, we have also performed Undersampling training with twice the dominant class data, but adjusting the missclassification so that the infected class has twice the value.

The results obtained are as follows:

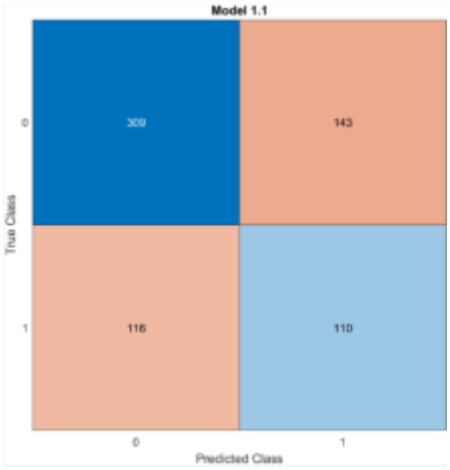


Figure 24: Fine Tree confusion matrix. Accuracy (Validation): 61,8 %

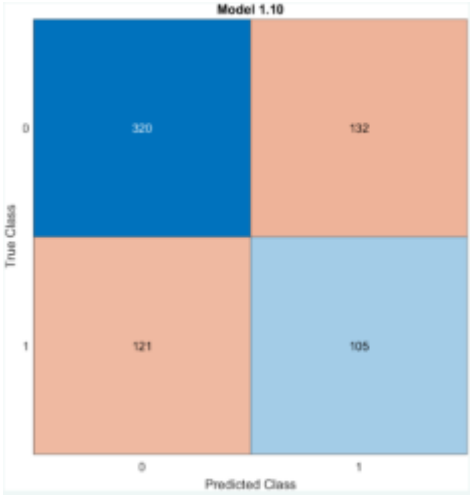


Figure 25: Cubic SVM confusion matrix. Accuracy (Validation): 62,7 %

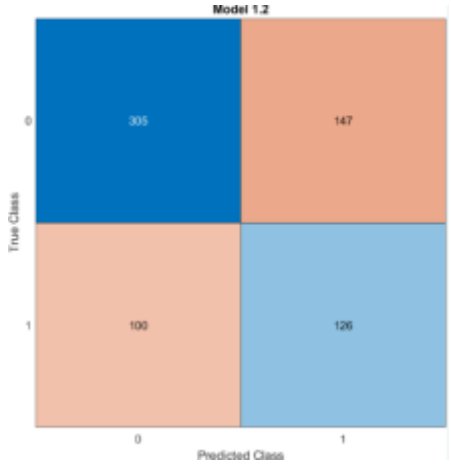


Figure 26: Medium Tree confusion matrix. Accuracy (Validation): 63,6 %

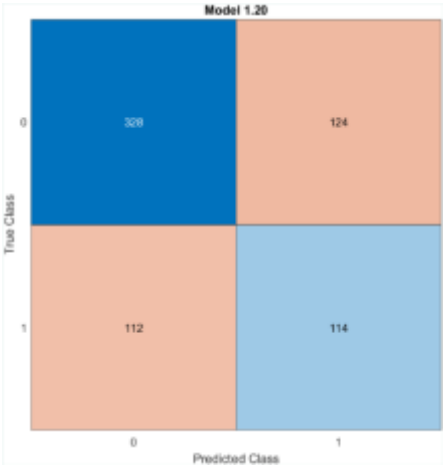


Figure 27: Boosted Trees confusion matrix. Accuracy (Validation): 65,2 %

As can be seen, the model with the best results is the Boosted Trees (Figure 27), with an accuracy of 65,2%. Therefore, it has been decided to apply the Hyperparameter Tunning to this model, applying the Bayesian Optimization.

The results obtained are as follows:

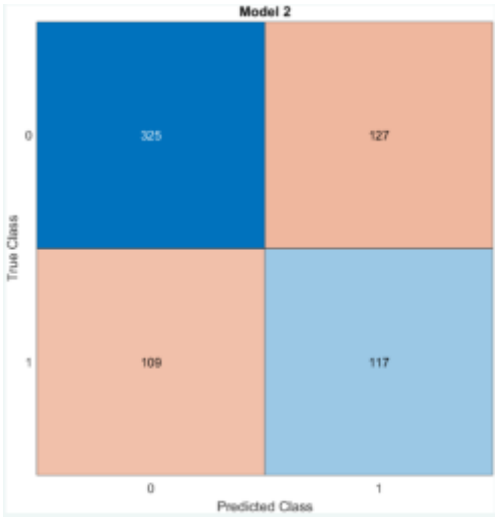


Figure 28: Ensemble Model Optimized confusion matrix. Accuracy (Validation): 65,2 %

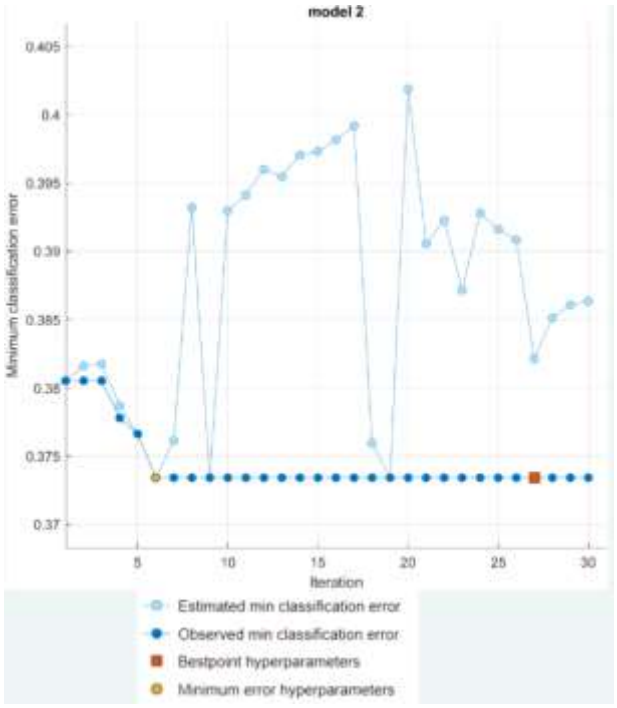


Figure 29: Minimum Classification Error of Ensemble Model Optimized

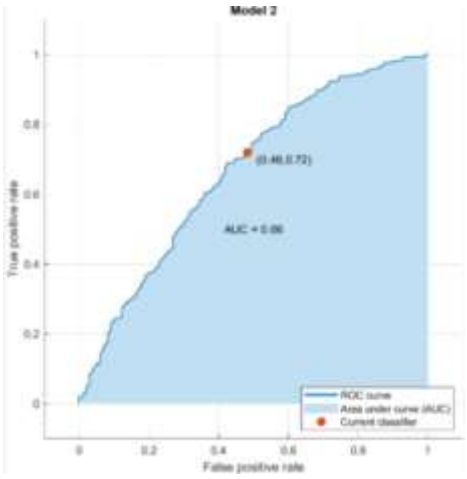


Figure 30: Ensemble Model Optimized ROC Curve

In this case, the plot of the Minimum Classification Error (Figure 29) shows that the observed error is much lower than the estimated error, a very positive indicator.

On the other hand, the ROC curve (Figure 30) we observe that the Area Under the Curve is 0.66, a little bit higher than the one of the Undersampling 1-1

4.2 Testing

Once the model is trained and validated, it is time to start with the test.

4.2.1 Undersampling 1-1

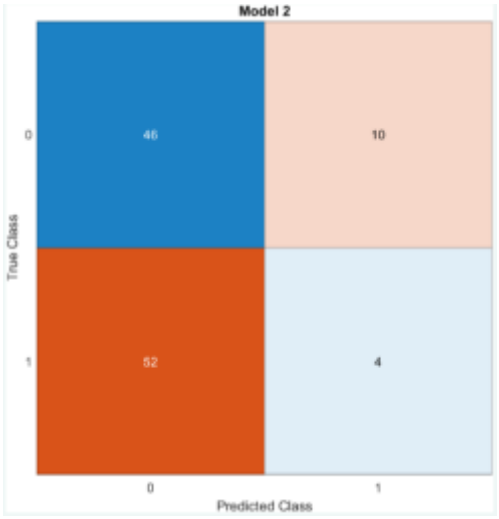


Figure 31: Test confusion matrix (Weighted KNN). Accuracy (Test): 44,6%

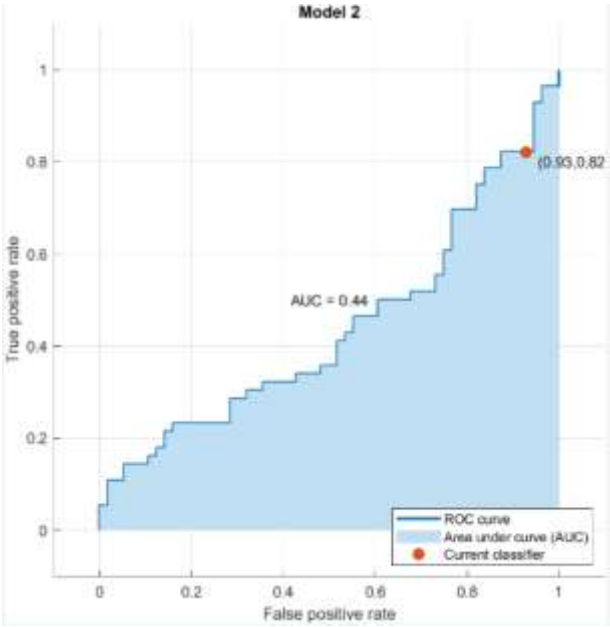


Figure 32: Test ROC Curve (Weighted KNN)

Once the test results are obtained, we calculate the parameters with which we can analyze the performance of the model. To do this we will use the formulas (5), (6), (7), (8) and (9) of section 3.5 and obtain the following parameters, comparing those obtained in the validation and in the test:

- Weighted KNN Validation (Figure 20):
    - Accuracy: 62,17 %
    - Sensitivity: 62,39 %
    - Specificity: 61,50 %
    - Positive predicted value: 61,95 %
    - F1-score: 0,62
- Weighted KNN Optimized Validation (Figure 21):
    - Accuracy: 64,16 %
    - Sensitivity: 63,27 %
    - Specificity: 65,04 %
    - Positive predicted value: 64,42 %
    - F1-score: 0,64
- Weighted KNN Optimized Test (Figure 31):
    - Accuracy: 44,64 %
    - Sensitivity: 7,14 %
    - Specificity: 82,14 %
    - Positive predicted value: 28,57 %
    - F1-score: 0,12

In this case of undersampling with the same number of classes, the results have worsened notably with respect to the test, with an accuracy of 44.6% and having now an F1-score of 0.12.

Normally, the test results worsen slightly, but in this case the change has been drastic. It has been assumed that this was due to a case of overfitting.

4.2.2 Undersampling 2-1 and class weighting

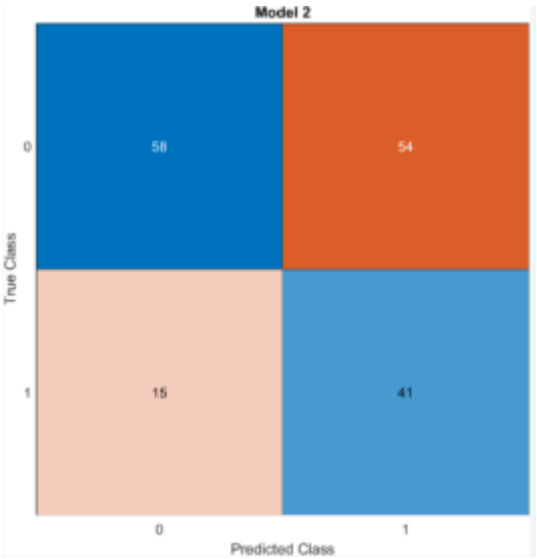


Figure 33: Test confusion matrix (Ensemble). Accuracy (Test): 58,9%

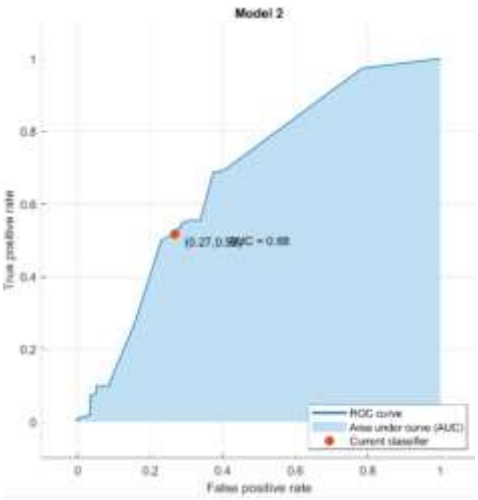


Figure 34: Test ROC Curve (Ensemble)

As in the previous case, the test results are obtained, we calculate the parameters with which we can analyze the performance of the model. To do this we will use the formulas (5), (6), (7), (8) and (9) of section 3.5 and obtain the following parameters, comparing those obtained in the validation and in the test:

- Boosted Trees Validation (Figure 27):
  - Accuracy: 65,19 %
  - Sensitivity: 50,44 %
  - Specificity: 72,57 %
  - Positive predicted value: 47,90 %
  - F1-score: 0,49
- Ensemble Optimized Validation (Figure 28):
  - Accuracy: 65,19 %
  - Sensitivity: 51,77 %
  - Specificity: 71,90 %
  - Positive predicted value: 47,95 %
  - F1-score: 0,50
- Ensemble Optimized Test (Figure 33):
  - Accuracy: 58,92 %
  - Sensitivity: 73,21 %
  - Specificity: 51,79 %
  - Positive predicted value: 43,16 %
  - F1-score: 0,54

On the other hand, in the case of Undersampling 2-1 and class weighting, the results are quite similar and have not decreased as much as in the case of Undersampling 1-1.

Are better results, with a sensitivity of 58.92% and an F1-score of 0.54. Therefore, we have obtained better results for the Ensemble Optimized model with Undersampling 2-1 and class weighting.

5. Conclusions

5.1 Summary of the findings

As noted above, in terms of testing, better results were obtained for the option Subsampling 2-1 + class weighting with Ensemble model than Subsampling 1-1 with weighted KNN model, probably due to overfitting in the latter case.

In addition, this study obtained much worse results than those obtained in the studies mentioned in the "State of the art" section. However, deep learning techniques were also used in these studies, achieving accuracy results of over 90%.

As shown by the F1-score value of 0.54, the model is getting more of a half of the positive and negative cases right. The model has difficulties in correctly classifying some cases, which may be due to several reasons: Firstly, lack of discriminatory features. Secondly, insufficient sample size (very low COVID-19 sample size).Future works and limitations.

5.2 Future works and limitations

As a final conclusion looking forward, it can be concluded that Machine Learning model is a low-cost tool that can be used in the future for quick, non-invasive testing of larger populations, but it has limitations.

Besides, as for future prospects, it would be desirable to test the model with a large sample and with a more equal proportion of healthy and COVID-19 patients. In this way, ECG features that are related to COVID-19 disease could be observed more effectively.

As a final conclusion, we can say that these models are useful as a complementary tool in the detection of COVID-19, since they can help to monitor the disease and prevent possible complications.

Moreover, being non-invasive techniques, they are very comfortable for the patient and also for the medical staff.

Therefore, these techniques are useful as mentioned as a complementary tool to diagnostic tests, but not as the only tool, as they still have certain limitations.

Bibliography

[1] G. Aghagoli, B. Gallo Marin, L. B. Soliman, and F. W. Sellke, "Cardiac involvement in COVID-19 patients: Risk factors, predictors, and complications: A review," *J Card Surg*, vol. 35, no. 6, pp. 1302–1305, Jun. 2020, doi: 10.1111/jocs.14538.

[2] "Coronavirus disease (COVID-19)." https://www.who.int/emergencies/diseases/novel-coronavirus-2019 (accessed Jul. 10, 2023).

[3] W. Guan *et al.*, "Clinical Characteristics of Coronavirus Disease 2019 in China," *New England Journal of Medicine*, vol. 382, no. 18, pp. 1708–1720, Apr. 2020, doi: 10.1056/NEJMoa2002032.

[4] F. Wu *et al.*, "Author Correction: A new coronavirus associated with human respiratory disease in China," *Nature*, vol. 580, no. 7803, pp. E7–E7, Apr. 2020, doi: 10.1038/s41586-020-2202-3.

[5] C. Huang *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," *The Lancet*, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140-6736(20)30183-5.

[6] C. J. Breen, G. P. Kelly, and W. G. Kernohan, "ECG interpretation skill acquisition: A review of learning, teaching and assessment," *J Electrocardiol*, vol. 73, pp. 125–128, Jul. 2022, doi: 10.1016/j.jelectrocard.2019.03.010.

[7] M. Nilsson, G. Bolinder, C. Held, B.-L. Johansson, U. Fors, and J. Östergren, "Evaluation of a web-based ECG-interpretation programme for undergraduate medical students," *BMC Med Educ*, vol. 8, no. 1, p. 25, Dec. 2008, doi: 10.1186/1472-6920-8-25.

[8] M. A. Ghossein, A. M. van Stipdonk, F. W. Prinzen, and K. Vernooy, "Vectorcardiographic QRS area as a predictor of response to cardiac resynchronization therapy.," *J Geriatr Cardiol*, vol. 19, no. 1, pp. 9–20, Jan. 2022, doi: 10.11909/j.issn.1671-5411.2022.01.003.

[9] H. Yang, S. T. Bukkapatnam, and R. Komanduri, "Spatiotemporal representation of cardiac vectorcardiogram (VCG) signals," *Biomed Eng Online*, vol. 11, no. 1, p. 16, 2012, doi: 10.1186/1475-925X-11-16.

[10] E. Mehraeen *et al.*, "A systematic review of ECG findings in patients with COVID-19," *Indian Heart Journal*, vol. 72, no. 6. Elsevier B.V., pp. 500–507, Nov. 01, 2020. doi: 10.1016/j.ihj.2020.11.007.

[11] A. Türkay Kunt, N. Kozaci, and E. Torun, "Mortality Predictors in Patients Diagnosed with COVID-19 in the Emergency Department: ECG, Laboratory and CT," *Medicina (B Aires)*, vol. 57, no. 6, p. 629, Jun. 2021, doi: 10.3390/medicina57060629.

[12] B. Long *et al.*, "Electrocardiographic manifestations of COVID-19," *Am J Emerg Med*, vol. 41, pp. 96–103, Mar. 2021, doi: 10.1016/j.ajem.2020.12.060.

[13] M. M. Bassiouni, I. Hegazy, N. Rizk, E.-S. A. El-Dahshan, and A. M. Salem, "Automated Detection of COVID-19 Using Deep Learning Approaches with Paper-Based ECG Reports.," *Circuits Syst Signal Process*, vol. 41, no. 10, pp. 5535–5577, 2022, doi: 10.1007/s00034-022-02035-1.

[14] K. Prashant, P. Choudhary, T. Agrawal, and E. Kaushik, "OWAE-Net: COVID-19 detection from ECG images using deep learning and optimized weighted average ensemble technique," *Intelligent Systems with Applications*, vol. 16, p. 200154, Nov. 2022, doi: 10.1016/j.iswa.2022.200154.

[15] A. S. Sakr, P. Pławiak, R. Tadeusiewicz, J. Pławiak, M. Sakr, and M. Hammad, "ECG-COVID: An end-to-end deep model based on electrocardiogram for COVID-19 detection," *Inf Sci (N Y)*, vol. 619, pp. 324–339, Jan. 2023, doi: 10.1016/j.ins.2022.11.069.

[16] M. A. Ghossein, A. M. van Stipdonk, F. W. Prinzen, and K. Vernooy, "Vectorcardiographic QRS area as a predictor of response to cardiac resynchronization therapy.," *J Geriatr Cardiol*, vol. 19, no. 1, pp. 9–20, Jan. 2022, doi: 10.11909/j.issn.1671-5411.2022.01.003.

[17] A. Jaroszyński, J. Furmaga, T. Zapolski, T. Zaborowski, S. Rudzki, and W. Dąbrowski, "The improvement of QRS-T angle as a manifestation of reverse electrical remodeling following renal transplantation in end-stage kidney disease patients on haemodialysis," *BMC Nephrol*, vol. 20, no. 1, p. 441, Dec. 2019, doi: 10.1186/s12882-019-1624-3.

A. Appendix: MATLAB Code

get_12ECG_features(): Function to extract the features

```
function features = get_12ECG_features(museECG)

addpath(genpath('C:\Users\usuario\OneDrive\Documentos\4º GIB\TFG\matlab-
classifier-2020-master\Tools'))
load('HRVparams_12ECG','HRVparams')

num_leads = length(museECG.leadData); % number of leads (12)

% Create a data matrix to store the amplitude data of the 12 leads
data = zeros(num_leads, 2488);

% Iterate through the leadData cell array and extract the amplitude column of each
cell
for i = 1:num_leads
    data(i,:) = museECG.leadData{i}(:,2)';
end


B = museECG.leadData{1,1};
Total_time = size(B,1)/museECG.samplingFrequency;   % total time of recording in
seconds

Fs = museECG.samplingFrequency;
age = museECG.ageMonths;

if museECG.sex == 'M'
    sex = 0;
else
    sex = 1;
end

HRVparams.Fs=Fs;
HRVparams.PeakDetect.windows = floor(Total_time-1);
HRVparams.windowlength = floor(Total_time);

ecg1 = data(1, :);
[~, jqrs_ann1, ~] = pan_tompkin(ecg1, Fs, 0);
P1_tompkins = jqrs_ann1;
```

```
% P2: pan tompkings de los leads 7-12 P2 = [40, 50,...]
ecg2 = data(7, :);
[~, jqrs_ann2, ~] = pan_tompkin(ecg2, Fs, 0);
P2_tompkins = jqrs_ann2;
da = abs(P1_tompkins(1)-P2_tompkins(1));
    if P1_tompkins(1) <= P2_tompkins(1)
    ECG_align = [data(1:6, 1:end-da); data(7:12, da+1:end)];
        else
    ECG_align = [data(1:6, da+1:end); data(7:12, 1:end-da)];
    end

data = ECG_align;

% Median filter to remove baseline wander
ECG12filt = zeros(num_leads, size(data,2));
for i = 1:num_leads
    ECG12filt(i,:) = medianfilter(data(i,:)', Fs);
end


% P1: pan tompkings de los leads 1-6 P1 = [30, 40,...]
% ecg = data(1, :);
% [~, jqrs_ann, ~] = pan_tompkin(ecg, Fs, 0);
% P2: pan tompkings de los leads 7-12 P2 = [40, 50,...]
% ecg = data(7, :);
% [~, jqrs_ann, ~] = pan_tompkin(ecg, Fs, 0);
% da = abs(P1(1)-P2(1));
% if P1(1)<= P2(1)
%  ECG_align = [ecg(1:6, da:end); ecg(7:12,end-da)]
% else
% ECG_align = [ecg(1:6, 1:end-da); ecg(7:12, da:end)]


% Convert 12 leads to XYZ leads using Kors transformation
XYZLeads = Kors_git(ECG12filt);
VecMag = vecnorm(XYZLeads');

feature_names = {'FileName', 'Age', 'Sex', 'QRSTang', 'QRSTang_M', 'AZ_OQ',
'AZ_OQM', 'AZ_OT', 'AZ_OTM', 'AZ_SVG', 'AZ_SVG_M', 'EL_OQ', 'EL_OQM', 'EL_OT',
'EL_OTM', 'EL_SVG', 'EL_SVG_M', 'QRS_Mag', 'QRS_Mag_M', 'T_Mag', 'T_Mag_M',
'SVG_Mag', 'QT_Interval', 'AUC_VM_QT', 'WVG'};
% leads_names = {'Lead I', 'Lead II', 'Lead III', 'Lead aVR', 'Lead aVL', 'Lead
aVF', 'Lead V1', 'Lead V2', 'Lead V3', 'Lead V4', 'Lead V5', 'Lead V6'};
features = cell(1, length(feature_names));
features(1, 1) = {museECG.fileName};
features(1, 2) = num2cell(age);
features(1, 3) = num2cell(sex);


% Convert ECG waveform to RR intervals using Pan-Tompkins algorithm
ecg = data(1, :);
[~, jqrs_ann, ~] = pan_tompkin(ecg, Fs, 0);

% Find fiducial points using ECGKit
```

```
ECG_header.nsig = 1; ECG_header.freq = Fs; ECG_header.nsamp = length(VecMag);
wavedet_config.setup.wavedet.QRS_detection_only = 0;
[Fid_pts, ~, ~] = wavedet_3D_ECGKit(VecMag', jqrs_ann', ECG_header,
wavedet_config);
[XYZ_Median, Fid_pts_Median] = Time_coherent_code_github(XYZLeads, Fid_pts, Fs);
GEH_features = GEH_analysis_git(XYZ_Median, Fid_pts_Median, Fs);

features(1, 4:end) = num2cell(GEH_features);


features = cell2table(features, 'VariableNames', [feature_names]);
end
```

main_ECG: Script to call the function

```
% Carpeta que contiene los archivos ECG
folder_path = 'C:\Users\usuario\OneDrive\Documentos\4º GIB\TFG\New_Covid_Signal';

% Obtener la lista de archivos en la carpeta
files = dir(fullfile(folder_path, '*.mat'));

% Inicializar la tabla para almacenar los resultados
result_table = table();

cont_error = 0;

% Iterar sobre cada archivo
for i = 1:numel(files)

    try
    %%
    fprintf('File %d/%d\n', i, numel(files))
    % Cargar el archivo ECG
    file_path = fullfile(folder_path, files(i).name);
    load(file_path, 'museECG');

    % Obtener las características utilizando la función get_12ECG_features
    features = get_12ECG_features(museECG);

    % Agregar una columna con el nombre del archivo
    features.FileName = repmat({files(i).name}, size(features, 1), 1);

    % Agregar los resultados a la tabla principal
    result_table = [result_table; features];

    catch ERROR
        fprintf('-------WARNING ERROR-------\n')
        cont_error = cont_error + 1;
        continue
    end
end
```

```
fprintf('\n Total errors: %d\n', cont_error)

% Mostrar la tabla con los resultados
disp(result_table);
```

main_ML: Sript to do Machine Learning preprocessing

```
% File para cargar y procesar datos para machine learning

% 1- load data
rutaArchivo = 'C:\Users\usuario\OneDrive\Documentos\4º
GIB\TFG\features_COMPLETO.xlsx' ;
data = readtable(rutaArchivo);

% I - FEATURE PREPROCESSING

% 2 - train-test splitting
% divide data into 80% training 20% test - stratified: maintaining the
% proportion of both classes (covid vs non-covid) in both train and test
% TODAS LAS TRANSFORMACIONES QUE SE HAGAN EN EL TRAINING TAMBIEN SE APLICAN
% AL TEST

idx_1 = find(data.COVID == 1);
idx_0 = find(data.COVID == 0);

t_i1 = round(0.8*length(idx_1));
t_i0 = round(0.8*length(idx_0));

idx_train = [idx_1(1:t_i1); idx_0(1:t_i0)];
idx_test = [idx_1(t_i1+1:end); idx_0(t_i0+1:end)];

train = data(idx_train, :);
test = data(idx_test, :);


% --------- ONLY TRAINING SET----------

% 3 - missing data
% 3.1. eliminate patients with more than 50% of missing values
missingFilas = sum(ismissing(train), 2);
filasConMissingAltos = find(missingFilas > 0.5*size(train, 2));
disp('Filas del training con más del 50% de valores faltantes:');
disp(filasConMissingAltos);

% 3.2. eliminiate columns (features) with more than 50% missing values
missingColumnas = sum(ismissing(train));
columnasConMissingAltos = find(missingColumnas > 0.5*size(train, 1));
disp('Columnas del training con más del 50% de valores faltantes:');
disp(columnasConMissingAltos);
```

```
% Localizacion del missing data: NO TENEMOS NINGÚN MISSING DATA
[filasIndice_train, columnaIndice_train] = find(ismissing(train));

% 4. Outliers
% find outliers in training data: for each column independenlty (feature)
% find upper and downer limit.
% then subtitute outliers for missing value

% Seleccionamos las columnas 3 y 7:28, ya que la 1,2,4,5,6 son
% ID,FileName, Sex, COVID, Death

selectedColumns = [3, 7:28];

train_removed = train;
cont = 0;
for col = selectedColumns
    cont = cont+1;
    % Obtener los valores de la columna actual
    columnData = train_removed{:, col};
    % Calcular los percentiles y los límites
    Q_25 = prctile(columnData, 25);  % Percentil 25
    Q_75 = prctile(columnData, 75);  % Percentil 75
    IQR = Q_75 - Q_25;
    lim_up(cont) = Q_75 + 1.5 * IQR;
    lim_down(cont) = Q_25 - 1.5 * IQR;
    % Encontrar los outliers
    outliers = (columnData < lim_down(cont)) | (columnData > lim_up(cont));
    % Sustituir los outliers por missing data
    train_removed{outliers, col} = NaN;
end

% 5- missing data imputation
% diferentes maneras:
%   - univariate: media, mediana - específico para columna
%   - multivariate: KNN ...
% Elegimos modo univariate(media) para imputar un valor a los missing data

train_filled = train_removed;
% Iterar sobre las columnas seleccionadas
cont = 0;
for col = selectedColumns
    cont = cont+1;
    % Obtener los valores de la columna actual
    columnData = train_removed{:, col};
    % Calcular la media de la columna
    columnMean(cont) = nanmean(columnData);
    % Sustituir los missing data por la media de la columna
    columnData(isnan(columnData)) = columnMean(cont);
    % Actualizar la columna en la tabla
    train_filled{:, col} = columnData;
end

train = train_filled;
```

```
% 6 - data normalization/standarization
% z-scoring
% Teoricamente, antes de aplicar la z-scor standarization
% se deberia comprobar que la variable sigue una distribucion normal

% Columnas a estandarizar (índices basados en 1)
columnas_estandarizar = [3,7:28];
% Obtener las columnas seleccionadas como una matriz
columnas_seleccionadas = table2array(train(:, columnas_estandarizar));
% Calcular la media y la desviación estándar de las columnas seleccionadas
media = mean(columnas_seleccionadas);
desviacion_estandar = std(columnas_seleccionadas);
% Realizar la estandarización Z-score en las columnas seleccionadas
columnas_seleccionadas_estandarizadas = (columnas_seleccionadas - media) ./
desviacion_estandar;
% Asignar las columnas estandarizadas a la tabla de datos original
train(:, columnas_estandarizar) =
array2table(columnas_seleccionadas_estandarizadas);


% ---- train UNDERSAMPLING 1-1 -----

% train_under1 = train;
% idx_1 = find(train.COVID == 1);
% idx_0 = find(train.COVID == 0);
%
% total_1 = length(idx_1);
% delete_0 = idx_0(total_1+1:end);
%
% train_under1(delete_0, :) = [];

% ---- train UNDERSAMPLING 2-1 -----

train_under2 = train;
idx_1 = find(train.COVID == 1);
idx_0 = find(train.COVID == 0);

total_1 = length(idx_1);
total_0 = 2 * total_1; % Duplicar el número de instancias de la clase dominante

if total_0 < length(idx_0)
    delete_0 = idx_0(total_0+1:end);
    train_under2(delete_0, :) = [];
end


%--------------------- TEST PREPROCESSING--------------------------

% 1. Missing Data
% 1.1. eliminate patients with more than 50% of missing values
missingFilas = sum(ismissing(test), 2);
filasConMissingAltos = find(missingFilas > 0.5*size(test, 2));
disp('Filas del test con más del 50% de valores faltantes:');
```

```
disp(filasConMissingAltos);
% 1.2. eliminiate columns (features) with more than 50% missing values
missingColumnas = sum(ismissing(test));
columnasConMissingAltos = find(missingColumnas > 0.5*size(test, 1));
disp('Columnas del test con más del 50% de valores faltantes:');
disp(columnasConMissingAltos);
% Localizacion del missing data: NO TENEMOS NINGÚN MISSING DATA
[filasIndice_test, columnaIndice_test] = find(ismissing(test));

% 2. Outliers
test_removed = test;
selectedColumns = [3, 7:28];
cont = 0;
for col = selectedColumns
    cont = cont+1;
    % Obtener los valores de la columna actual
    columnData = test_removed{:, col};
    % Encontrar los outliers
    outliers = (columnData < lim_down(cont)) | (columnData > lim_up(cont));
    % Sustituir los outliers por missing data
    test_removed{outliers, col} = NaN;
end

% Missing data imputation
test_filled = test_removed;
% Iterar sobre las columnas seleccionadas

cont = 0;
for col = selectedColumns
    cont = cont+1
    % Obtener los valores de la columna actual
    columnData = test_removed{:, col};
    % Sustituir los missing data por la media de la columna
    columnData(isnan(columnData)) = columnMean(cont);
    % Actualizar la columna en la tabla
    test_filled{:, col} = columnData;
end
test = test_filled;

% ---- train UNDERSAMPLING 1-1 -----

% test_under1 = test;
% idx_1 = find(test.COVID == 1);
% idx_0 = find(test.COVID == 0);
%
% total_1 = length(idx_1);
% delete_0 = idx_0(total_1+1:end);
%
% test_under1(delete_0, :) = [];

% ---- train UNDERSAMPLING 2-1 -----

test_under2 = test;
idx_1 = find(test.COVID == 1);
```

```
idx_0 = find(test.COVID == 0);

total_1 = length(idx_1);
total_0 = 2 * total_1; % Duplicar el número de instancias de la clase dominante

if total_0 < length(idx_0)
    delete_0 = idx_0(total_0+1:end);
    test_under2(delete_0, :) = [];
end

% Z-Score
% Obtener las columnas seleccionadas como una matriz

columnas_estandarizar = [3,7:28];
% Obtener las columnas seleccionadas como una matriz
columnas_seleccionadas = table2array(test(:, columnas_estandarizar));
% Calcular la media y la desviación estándar de las columnas seleccionadas
% Realizar la estandarización Z-score en las columnas seleccionadas
columnas_seleccionadas_estandarizadas = (columnas_seleccionadas - media) ./
desviacion_estandar;
% Asignar las columnas estandarizadas a la tabla de datos original
test(:, columnas_estandarizar) =
array2table(columnas_seleccionadas_estandarizadas);
%% TARGET

% 1 - training
% 2 - validation
% 3 - test
% cada uno tiene que tener una columna de label (en nuestro caso COVID)
% predcited_labels = trainedModel_RUS.predictFcn(features_test)
% compute sensitivity, specificity, positive precitive value, f1-score


% % train - validation splitting
% tr_idx_1 = find(train.COVID == 1);
% tr_idx_0 = find(train.COVID == 0);
%
% tr_t_i1 = round(0.8*length(tr_idx_1));
% tr_t_i0 = round(0.8*length(tr_idx_0));
%
% tr_idx_train = [tr_idx_1(1:tr_t_i1); tr_idx_0(1:tr_t_i0)];
% tr_idx_val = [tr_idx_1(tr_t_i1+1:end); tr_idx_0(tr_t_i0+1:end)];
%
% train_train = data(tr_idx_train, :);
% train_val = data(tr_idx_val, :);
```

List of Figures

Acknowledgements

I would like to express my sincere thanks to all those who have contributed significantly to the completion of this thesis. First of all, I wish to acknowledge the valuable support provided by the staff of the Universitat Politècnica de València and Politecnico di Milano, who have been a constant source of knowledge, guidance and motivation throughout this arduous process.

I am deeply grateful to my professors and academic advisors for their dedication and commitment in guiding me throughout this research. Their experience and wisdom have been fundamental for the development of this thesis and have left an indelible mark on my academic formation.

I cannot overlook the unconditional support of my classmates. Their encouragement, collaboration and valuable discussions have been a constant source of inspiration and motivation. Together we have shared challenges, triumphs and moments of personal and professional growth. To each of you, thank you for being part of this journey and for your enduring friendship.

A special thanks goes out to my family, who have been always by my side. Your unconditional support, words of encouragement and understanding have been the driving force that has kept me going through difficult times. Their love and sacrifice have been fundamental in achieving this goal and I will always be grateful for their constant support.

Finally, I am also grateful to the institutions and organizations that have facilitated access to the resources and information necessary to carry out this research.

In short, this achievement would not have been possible without the unconditional support of all these people. Your presence, encouragement and confidence in me have been the fundamental pillars for overcoming the challenges and achieving success. My deepest gratitude to every one of you for being an essential part of this journey and for contributing to make this dream come true.