

POLITECNICO DI TORINO

Master's Degree
in Mechatronic Engineering

Master's Degree Thesis

**A Deep Learning Method for Automatic Detection
of Most Relevant ECG Features**



Supervisors

Prof. Eros Gian Alessandro Pasero
Eng. Vincenzo Randazzo

Candidate

Valentina De Vitis

Academic Year 2021-2022

*To all my family,
with love*

Summary

The main cause of mortality in the world are cardiovascular diseases (CVDs). Each year, around 17.9 million people die from CVDs and this number represents the 32% of all global deaths. Early diagnosis and treatment are very important for people which present high cardiovascular risk.

Irregularities in the heartbeat rhythm are defined "Arrhythmias" and they can rarely occur during human's life. This kind of problems can lead to some complications that may constitute an immediate risk for life and may cause potentially fatal events. The premature classification and detection of arrhythmias is a good starting point for cardiac disease diagnosis.

The most important element to detect these events is the electrocardiogram (ECG), a non-expensive, non-invasive method, which gives a record of heart's electrical functionality. The interpretation process of ECG requires a high degree of training and is time-consuming. So, the first attempts to automate the interpretation of ECG are dated at the end of 1950s.

The ECG is used for diagnosis of cardiovascular disease, and it mainly consists of three principal waves: P-wave, QRS complex and T-wave. The most relevant feature is the QRS complex because it represents the heartbeat, i.e., the ventricular contraction. It can also be the starting point for obtaining further useful medical information. Thus, for ECG-based evaluation, the QRS detection is critical. ECG signal, hence, contains a huge amount of information and it is difficult to analyse them just with a visual evaluation. For this reason, the design of an automated efficient system to detect the relevant features in an ECG signal is a critical task.

A new interesting approach, developed in the last decade, is Deep Learning. The huge impact of Deep Learning has motivated the implementation of this methodology for automatic classification of ECG features. In particular, the application of Deep Neural Networks in electrocardiographic signals is gaining importance to explore the enormous quantity of information that these signals contain. Today, deep learning enlarges the vision, introducing new methods to achieve better accuracy and increase time management in ECG features detection.

This thesis project focuses on a deep learning method to automatically detect the most relevant features in ECG signals. The learning-based approach is hybrid because it combines two different learning models. After signal pre-processing, using local regression, data are downsampled, then 1D ECG signals are converted into 2D Scalogram images to make easier the feature extraction. Finally, two methods are combined to construct the model: Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory Networks (BiLSTMs), so that a hybrid model, called CNN-BiLSTM, is constructed.

For training purposes, a k-fold cross validation (with $k = 10$) was used to test the model's ability to predict data that was never seen before. At the beginning, for training and testing of the proposed method and for evaluating the performances of the CNN-BiLSTM approach, a publicly available dataset, named "QT Database", was used. With this large amount of data, the proposed method provides an accuracy of 97.5% in QRS complexes detection. The relevant features extracted are finally measured and analysed to acquire the complete information of ECG signal.

Contents

List of Tables	VI
List of Figures	VII
1 Principles of Heart Anatomy	1
1.1 Heart Anatomy	1
1.1.1 Cardiovascular System	3
1.1.2 Electrical Conduction System	5
1.2 Electrocardiography	7
1.2.1 ECG Signal	10
1.2.2 ECG Grid	12
1.3 Cardiovascular Diseases (CVDs)	13
2 Principles of Deep Learning	15
2.1 General Introduction	15
2.2 Types of Deep Neural Networks	16
2.2.1 Multi-Layer Perceptrons (MLPs)	18
2.2.2 Convolutional Neural Networks (CNNs)	19
2.2.3 Recurrent Neural Networks (RNNs)	20
2.3 Parameters Setup	21
2.3.1 Parameters	22
2.3.2 Hyperparameters	22
2.3.3 Optimization Strategies	23
2.4 Metrics and Performances	24
3 State of Art	29
3.1 Deep Learning in Electrocardiography	29
3.1.1 Deep Learning for ECG Analysis	31
3.1.2 Deep Learning for CVD Diagnosis	32

4	Methodology	35
4.1	Google Colab	35
4.2	Dataset Description	35
4.3	Signal Pre-processing	37
4.3.1	Linear Regression Method for Baseline Wander Removal	38
4.3.2	Scalogram Derivation	40
4.4	Data Segmentation	42
4.5	Used Deep Learning structures	43
4.5.1	Convolutional Neural Networks' Layers	43
4.5.2	Bidirectional Long Short-Term Memory Neural Networks' Layers	44
4.5.3	Training of the Model	45
4.5.4	Optimizer and Activation Functions	46
4.6	k-fold Cross Validation	48
4.7	Other Features Extraction	49
5	Implementation and Results	52
5.1	Deep Neural Network Implementation	52
5.2	Results for QRS Complexes Detection	55
5.3	Results for P Waves Detection	59
5.4	Results for T Waves Detection	62
5.5	Features Measurements	66
5.5.1	Features Segmentation with Developed Models	66
5.5.2	Widths and Intervals Measurements	68
6	Validation and Discussion	76
6.1	Results' Validation	76
6.2	Discussion	78
6.3	Conclusions	81
	Acronyms	84
	Bibliography	94

List of Tables

4.1	QT Database Annotations	36
4.2	QT Database Recordings	37
5.1	Deep Neural Network Implementation	53
5.2	Results of k-fold CV for QRS Detection	55
5.3	Average Scores for all folds of CV	56
5.4	Various Metrics for QRS Detection Results	58
5.5	Results of k-fold CV for P Detection	59
5.6	Average Scores for all folds of CV	61
5.7	Various Metrics for P Detection Results	62
5.8	Results of k-fold CV for T Detection	62
5.9	Average Scores for all folds of CV	63
5.10	Various Metrics for T Detection Results	65
5.11	Obtained Results for each of the Features' Predictions	66
6.1	Results' Validation	77
6.2	Mean and Standard Deviation for Each of the Analysed Feature	78
6.3	Minimum and Maximum Deviations for Each of the Analysed Feature	78
6.4	Results for each of the Features' Predictions with Different Methods	80

List of Figures

1.1	Layers of Heart Wall.[1]	2
1.2	Heart Anatomy.[2]	3
1.3	The Heart and Circulatory System.[3]	4
1.4	The Heart Electrical Conduction System.[4]	6
1.5	Cardiac Action Potential Propagation.[5]	7
1.6	ECG leads on vertical (in blue) and horizontal (in red) planes.[6]	8
1.7	Einthoven's Triangle: ECG Limb and Augmented Limb Leads.[6]	9
1.8	ECG Signal Waves.[7]	10
1.9	Sequence of ECG Signal Generation.[8]	11
1.10	Grid Scale for ECG Wave Interpretation.[9]	12
2.1	Artificial Neural Network Structure.[10]	17
2.2	Hidden Node Structure.[10]	17
2.3	Convolutional Neural Network Structure.[11]	19
2.4	RNN simple cell versus LSTM cell.[12]	20
2.5	Main Activation Functions.[13]	22
2.6	Confusion Matrix.[14]	24
2.7	Receiver Operating Characteristic Curve.[15]	26
2.8	Precision-Recall Curve.[16]	27
3.1	Simple deep learning pipeline for ECG analysis.[17]	32
4.1	Example of signal extracted from 'QT Database' with annotations of QRS complexes.	38
4.2	Example of signal before and after removal of baseline variation.	40
4.3	Wavelet Morlet.[18]	41
4.4	Example of ECG signal with its associated scalogram.	41
4.5	Example of Scalogram in presence of Premature Ventricular Contraction Event.	42
4.6	Example of ECG data segmentation: each window contains one QRS complex or part of it with its associated label.	43
4.7	Gradient Optimization Process.[19]	46
4.8	ReLU Activation Function.	47
4.9	k-fold Cross Validation.[20]	48

4.10	Example of signal with annotation of P Waves.	49
4.11	Example of ECG data segmentation for P waves: each window contains one P wave or part of it with its associated label.	49
4.12	Example of signal with annotation of T Waves.	50
4.13	Example of ECG data segmentation for T waves: each window contains one T wave or part of it with its associated label.	50
5.1	Deep Neural Network Model Architecture.	54
5.2	Accuracy for QRS Detection during each epoch.	56
5.3	Loss for QRS Detection during each epoch.	56
5.4	ROC for QRS Detection.	57
5.5	Model's predictions on new signals.	57
5.6	Confusion Matrix for QRS Detection.	58
5.7	Accuracy for P Detection during each epoch.	59
5.8	Loss for P Detection during each epoch.	60
5.9	ROC for P Detection.	60
5.10	Model's predictions on new signals.	61
5.11	Confusion Matrix for P Detection.	61
5.12	Accuracy for T Detection during each epoch.	63
5.13	Loss for T Detection during each epoch.	63
5.14	ROC for T Detection.	64
5.15	Model's predictions on new signals.	64
5.16	Confusion Matrix for T Detection.	65
5.17	Predictions of the three networks on a random signal from Database.	67
5.18	Predictions of the three networks on a new signal never seen before.	67
5.19	Normal PR Interval on ECG grid.	68
5.20	Algorithm for Calculation of PR Interval in Each Window.	69
5.21	Normal QRS Complex on ECG grid.	70
5.22	Different QRS Shapes for Various Anomalies.[21]	71
5.23	Algorithm for Calculation of QRS Width in Each Window.	72
5.24	Normal QT Interval on ECG grid.	73
5.25	Algorithm for Calculation of QT Interval in Each Window.	73
5.26	Results of Intervals Measurements on a Prediction.	74
6.1	Performances Comparisons between Different Methods.	80

Chapter 1

Principles of Heart Anatomy

1.1 Heart Anatomy

The main muscular organ of the body is the heart, which pumps blood along all the body and makes it circulating through the circulatory/vascular system, transporting nutrients and oxygen to every tissue and organ. Its position is in the middle mediastinum, and it is wrapped in a serous sac composed of two layers, called pericardium, which provides mechanical protection. The heart has a shape of a quadrangular pyramid, whose base faces in the posterior thoracic wall and apex points towards the anterior thoracic wall. It is composed by striated involuntary muscle tissue and it can generate autonomously the electrical signal that allows its correct movement. The heart wall consists of several layers enclosed in the pericardium and clearly shown in Figure 1.1. In particular, the main layers are:

- **Epicardium:** formed by the visceral layer of the serous pericardium, it completely covers the external surface and gives it a translucent and smooth aspect.
- **Myocardium:** formed by striated excitable tissue, it is the muscular middle layer of the wall and constitutes the conducting system; in other words, this layer is responsible for contractions.
- **Endocardium:** which is the inner wall. The heart's cavities and valves are coated by this layer. It is formed by loose connective and simple squamous epithelial tissue. This layer also regulates heart's contractions.

The subepicardial and subendocardial layers mainly compose the rest of the heart.

In particular, the subepicardial layer joins the myocardium and the epicardium, while the subendocardial one joins the endocardium with the myocardium. The subendocardial layer contains the vessels and nerves of the heart's conducting system so, a damage to this layer can lead to arrhythmias.

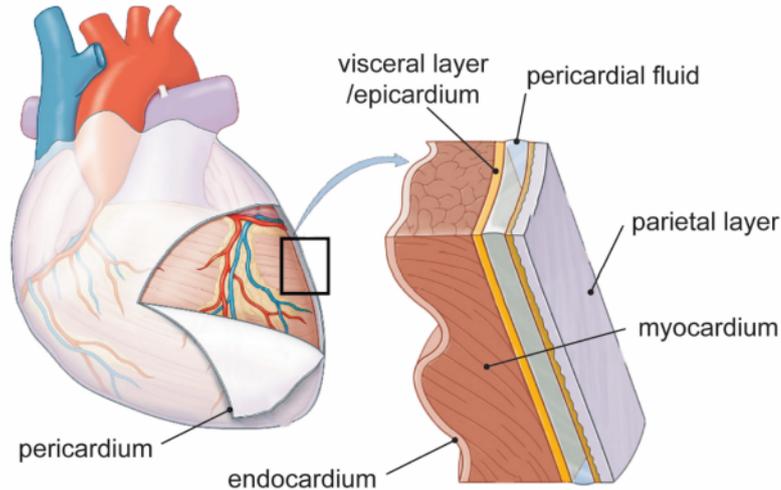


Figure 1.1. Layers of Heart Wall.[1]

Aorta, pulmonary vein, pulmonary artery, superior vena cava (SVC) and inferior vena cava (IVC) are the great vessels of the heart. All these vessels carry blood to and from heart. Aorta, in particular, has branches which supply the oxygenated blood to the whole body. Some of the most important branches of the aorta are the left subclavian and common carotid arteries and the brachiocephalic trunk. The SVC supply blood to the upper half of the body through left and right brachiocephalic veins, while the IVC supply the lower half of the body through the common iliac veins.

In the heart, there are also two pairs of valves: two atrioventricular and two semilunar valves. They maintain the unidirectionality of the blood's flow and prevent backward flow of the blood in the opposite direction. Atrioventricular valves (tricuspid and mitral valves) are located between atria and ventricles, semilunar ones (aortic and pulmonary valves) are in the ventricles' outflow tracts.

The heart (Figure 1.2) is divided into right and left halves by the septa and each half is subdivided into two cavities by a constriction. The upper cavity is called atrium, while the lower is the ventricle. Therefore, the heart consists of four chambers: right and left atria and ventricles. The atria differ in position and size, and they are not symmetrical. They communicate with ventricles through two valves. The left ventricle has muscle walls thicker than the right one.

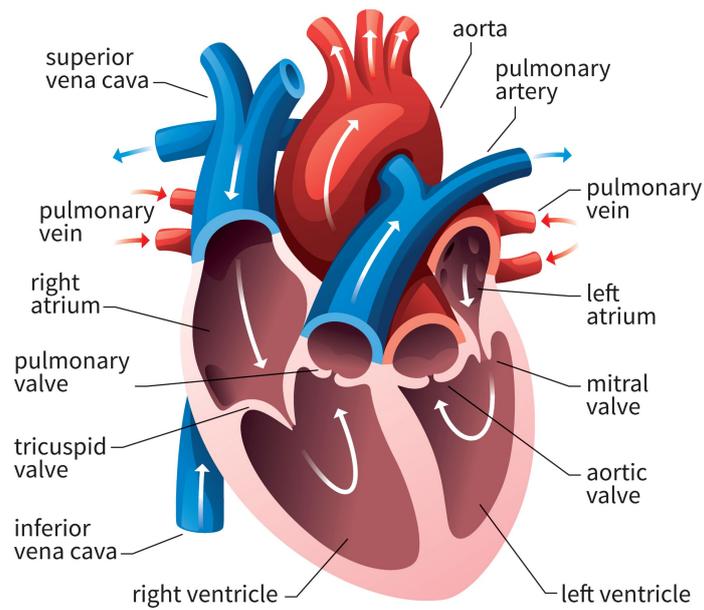


Figure 1.2. Heart Anatomy.[2]

1.1.1 Cardiovascular System

The heart has two atria and two ventricles. Furthermore, as said before, it has four valves: two semilunar valves, pulmonary and aortic, which regulate the flow between heart and blood vessels, and two atrioventricular valves (tricuspid and mitral) which allow unidirectionality of the blood flowing between atria and ventricles. The blood flows through the heart following this path (clearly visible in 1.3):

- The SVC, IVC and coronary sinus drain venous blood coming from the body into the right atrium. This blood comes in the atrium full of carbon dioxide deriving from cells and tissues.
- The right atrium pumps the blood into the right ventricle through the tricuspid valve.
- The right ventricle pumps blood into the pulmonary trunks by the pulmonary semilunar valve in order to oxygenate it into the lungs.
- Blood, rich of oxygen after returning from the lungs, drains into the left atrium through the four pulmonary veins.
- The left atrium pumps blood into the left ventricle via the bicuspid (mitral) valve.
- The left ventricle, through the aortic semilunar valve, pumps blood into the ascending aorta in order to return to cells and tissues and to supply the body.

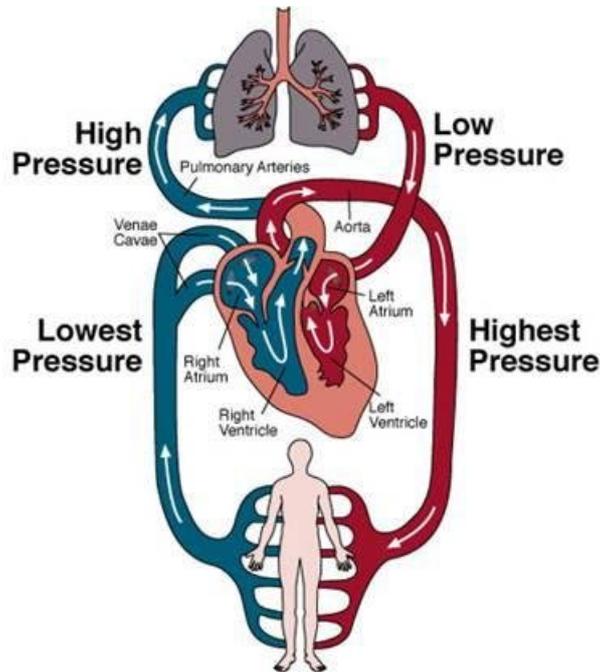


Figure 1.3. The Heart and Circulatory System.[3]

Summarizing, the deoxygenated blood is received by the right atrium and ventricle from systemic veins and pumped to the lungs, while the oxygenated blood is received by the left atrium and ventricle and pumped to the systemic vessels that can distribute it to the whole body. This heart cycle is completely regulated subconsciously by the cardiac plexus, an autonomous nerve plexus.

The blood circulation can be subdivided into systemic and pulmonary circulation:

- The **systemic circulation**, also called large circulation, transports blood oxygenated and rich of nutrients to the body's cells and tissues, and, when the exchanges have happened, transports it to the right side of the heart.
- The **pulmonary circulation**, also called small circulation, transports the blood rich of carbon dioxide to the lungs, and, when it becomes rich of oxygen, returns it to the heart's left side in order to reintroduce it into the systemic circulation.

In addition, each cardiac cycle has a phase in which the heart chamber is relaxed and fills with blood, called **diastole**, and a phase in which it is contracted and pumps blood, called **systole**.

1.1.2 Electrical Conduction System

The time of contraction of the heart's chambers and the pumping of the heart are regulated by the electrical conduction system (shown in Figure 1.4). In response to the electrical stimulus received, the heart muscle contracts. Electrical impulses are generated and conducted through the heart, inducing the muscle to contract and pump blood. The main elements in the conduction cardiac system are the atrioventricular and sinus nodes and the autonomic nervous system.

- The electrical impulses are generated in the sinus node, which is the cluster of cells located in the right atrium's wall (in its upper part). The sinus node, also called sinoatrial (SA) node, represents the heart's natural pacemaker.
- This generated electrical signal moves between cells down through the heart to reach the atrioventricular (AV) node. The AV node is a cells' cluster located between atria and ventricles in the centre of the heart.
- The AV node slows the electrical current and acts as a gate before letting the signal to pass down through the ventricles. Thanks to this delay, the atria have the possibility to fully contract before the stimulation of the ventricles.
- After passing through the AV node, the electrical current arrives to the ventricles flowing along special fibers, called bundle of His. In particular, the His bundle divides into two paths, one going to left and the other to right (bundle branches), which conduct the signal up to the respective Purkinje fibers (embedded in the lower part of the heart's walls), in order to give electrical stimulus to both the ventricles.
- The autonomic nervous system triggers the cardiac cycle's start by firing the sinus node. It can send a quick message to the sinus node so that it can, in turn, increase the heart rate very quickly (e.g., it can increase the rate to twice normal in 3 to 5 seconds). This is a fundamental response during exercise when the body increases its demand for oxygen so, the heart has to increase its beating speed.

In particular, the autonomic nervous system includes:

- The **sympathetic nervous system**, which makes the SA node's impulses faster, in order to increase the heart rate (fight or flight response).
- The **parasympathetic nervous system**, which makes the SA node's impulses slower, in order to decrease the heart rate (rest and digest response).

The discharge of an electrical stimulus is called "depolarization" while the recharge is called "repolarization". So, three stages form a single heartbeat: atrial depolarization, ventricular depolarization and finally atrial and ventricular repolarization.

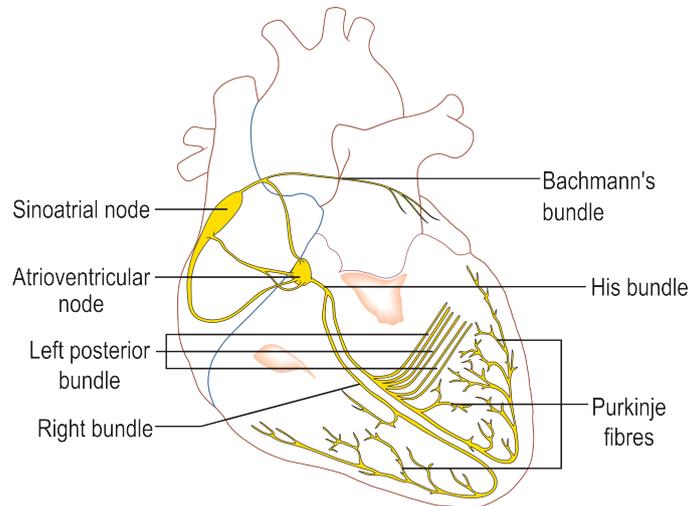


Figure 1.4. The Heart Electrical Conduction System.[4]

The electrical stimulus generated by heart is called "action potential" and it represents a voltage change through the heart cells' membrane. This change in voltage is caused by the ions' movements in and out the heart cells across proteins (called ion channels). As said before, the action potential arises from a specialized group of cells which is able to automatically generate potential, without help from nervous activity. Figure 1.5 shows the cardiac action potential propagation during the heart cycle.

The cardiac membrane potential in rest conditions is around -90mV and it means that cell is more negative inside than outside. In the same conditions, the main ions outside the cell are Na^+ and Cl^- while inside the cell there is a greater K^+ ion concentration. When the voltage becomes more positive, the action potential is triggered and the depolarization happens, due to the opening of sodium channels. Thanks to these channels, Na^+ moves inside the cell. On the contrary, the repolarization happens when the voltage becomes more negative, due to the opening of potassium channels allowing to K^+ to move outside. Inside and outside the cell, there are calcium ions (Ca^{2+}), responsible of the connection between the electrical signal and the muscle contraction. The action potential opens the calcium channels, inducing the release of a huge amount of Ca^{2+} ions. The Ca^{2+} release is essential for the plateau phase of action potential [22].

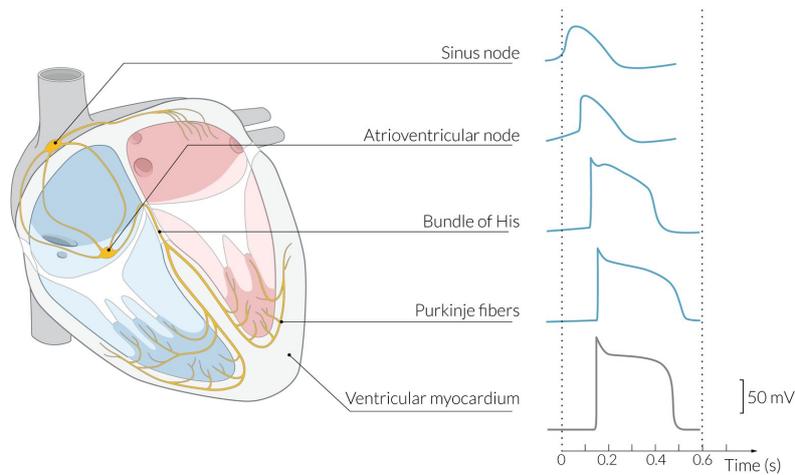


Figure 1.5. Cardiac Action Potential Propagation.[5]

1.2 Electrocardiography

The main method to monitor the heart's electrical activity is the electrocardiogram (ECG), a non-expensive and non-invasive method. Using electrodes placed on the skin, in fact, it is possible to obtain a graph with voltage versus time, which describes the electrical functionality of the heart. The electrodes detect the electrical changes that happen during each cardiac cycle and make possible to find numerous cardiac anomalies.

Conventionally it is used the 12-lead ECG, whose configuration has ten electrodes placed on the patient's surface of the chest and on his limbs. Then, the magnitude of the heart's electrical potential is recorded in a period of around ten seconds and is measured from twelve different angles, called "leads". With this technique, it is possible to capture the overall magnitude and direction of heart's electrical depolarization in the whole cardiac cycle. The ECG can be used to measure the size and position of the heart chambers, the rate and rhythm of heartbeats and the presence of some problems to the heart's cells or to the conduction system [23].

A lead is the dipole vector associated to each pair of electrodes or to an electrode and a zero reference, which is a virtual electrode with a potential equal to the average potential of the limb leads. A lead follows the electrical field generated by the heart. The 12 ECG leads are subdivided into two types:

- Six peripheral (or limb) leads, three unipolar and three bipolar.
- Six precordial (or chest) leads, all unipolar.

The limb leads are located in the coronal plane (vertical) while the chest leads lie on the perpendicular transverse plane (horizontal), as shown in Figure 1.6.

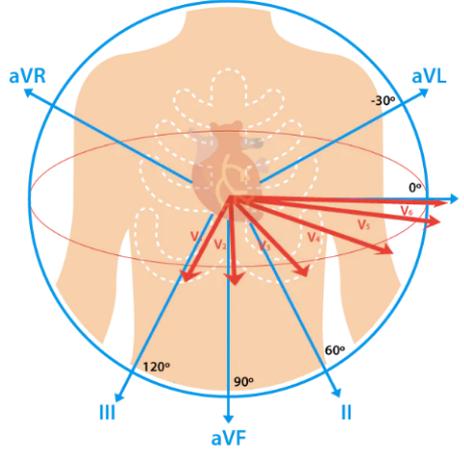


Figure 1.6. ECG leads on vertical (in blue) and horizontal (in red) planes.[6]

The difference between two electrodes positioned on the wrists or between a wrist and the ankle is represented by the bipolar, peripheral leads. In particular, leads I, II and III are the so-called limb leads and are located on the limbs (one on each wrist and one on the left leg). The limb leads constitute the Einthoven's triangle [24] and are derived from Equations 1.1, 1.2 and 1.3.

- **Lead I:** voltage between (positive) left arm (LA) and right arm (RA) electrodes. It gives 0 degree angle of orientation.

$$I = LA - RA. \quad (1.1)$$

- **Lead II:** voltage between (positive) left foot (LF) and right arm (RA) electrodes. It gives +60 degree angle of orientation.

$$II = LF - RA. \quad (1.2)$$

- **Lead III:** voltage between (positive) left foot (LF) and left arm (LA) electrodes. It gives +120 degree angle of orientation.

$$III = LF - LA. \quad (1.3)$$

Unipolar, limb leads are aVR, aVL and aVF and are also called augmented limb leads (described in Equations 1.4, 1.5 and 1.6). They derive from the same electrodes of the previous leads with the difference that they use Goldberger's central terminal as negative pole. This reference is obtained, with respect to a vertex, as the average of the potential of the other two vertices.

- **Lead aVR:** is the lead augmented vector right and has the positive electrode on the right arm. The negative pole is a combination of the other two electrodes. It gives -150 degree angle of orientation.

$$aVR = RA - \frac{1}{2}(LA + LF) \quad (1.4)$$

- **Lead aVL:** is the lead augmented vector left and has the positive electrode on the left arm. The negative pole is a combination of the other two electrodes. It gives -30 degree angle of orientation.

$$aVL = LA - \frac{1}{2}(RA + LF) \quad (1.5)$$

- **Lead aVF:** is the lead augmented vector foot and has the positive electrode on the left foot. The negative pole is a combination of the other two electrodes. It gives +90 degree angle of orientation.

$$aVF = LF - \frac{1}{2}(RA + LA) \quad (1.6)$$

These six limb leads are shown in Figure 1.7 and are used to calculate the heart's electrical axis in the frontal plane.

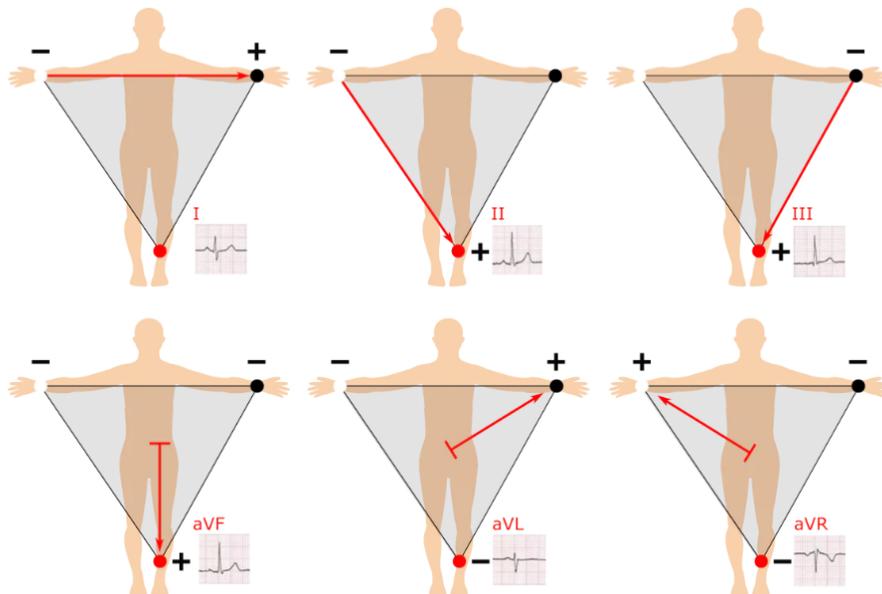


Figure 1.7. Einthoven's Triangle: ECG Limb and Augmented Limb Leads.[6]

The six precordial, unipolar leads are V_1 , V_2 , V_3 , V_4 , V_5 and V_6 . They are perpendicular to the limb leads and their electrodes are the positive poles. The potential is measured between an electrode and the Wilson's central terminal, which is obtained as the weighted sum of the potential of the other bipolar leads, virtually placed in the heart's centre.

1.2.1 ECG Signal

The ECG signal has typically an amplitude between 1 and 3 mV, a frequency range of 0.5-200 Hz and its interpretation is generally based on pattern recognition. The heart's depolarization and repolarization, in fact, produce some deflections that form what is called electrocardiogram. The main characteristics are:

- A positive deflection caused by the heart's depolarization toward the positive electrode.
- A negative deflection caused by the heart's depolarization away from positive electrode.
- A negative deflection caused by the heart's repolarization toward positive electrode.
- A positive deflection caused by the heart's repolarization away from positive electrode.

In particular, normal rhythm (1.8) produces: a P wave, which represents the atrial depolarization, a QRS complex, which represents the ventricular depolarization, a T wave, which represents the ventricular repolarization, and a U wave, which indicates the papillary muscle depolarization (it is typically not present). Changes in heart's structure can change this pattern.

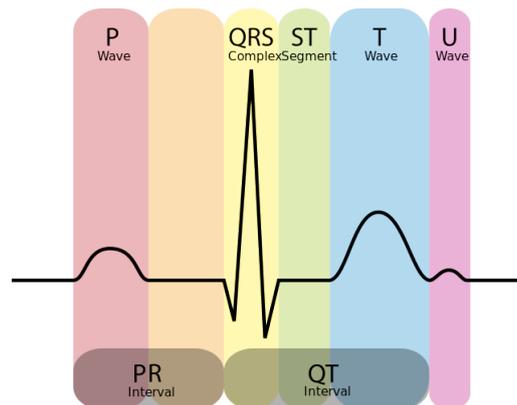


Figure 1.8. ECG Signal Waves.[7]

As said before, the ECG records the heart's electrical activity, so the sequence of ECG signal generation can be seen in Figure 1.9 and described as follows:

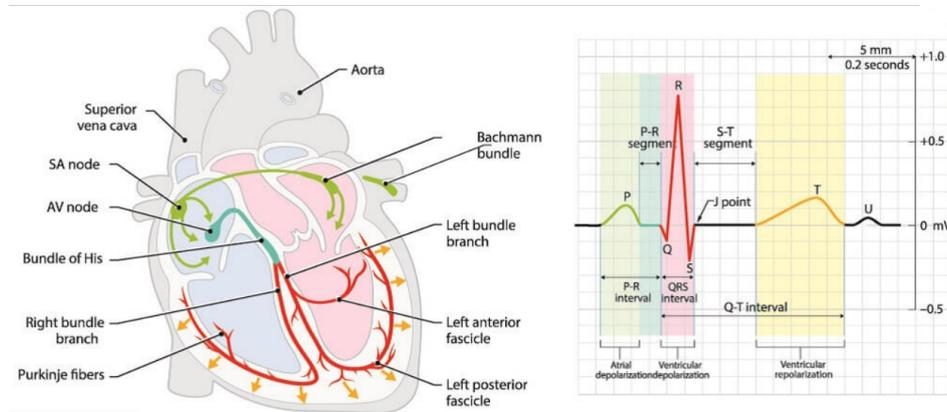


Figure 1.9. Sequence of ECG Signal Generation. [8]

- SA node produces P wave. Heart's electrical activity, in healthy conditions, is autonomously generated by the SA node. This impulse propagates in the right atrium and arrives to the left atrium thanks to the Bachmann's bundle. The impulse stimulates the atria's myocardium to contract and this conduction is seen on ECG signal as P wave. The normal duration is < 80 ms. Then, through the internodal tracts, the impulse travels from SA to AV node [25].
- AV node and bundles produce PR interval. The AV node delay is fundamental in the conduction system because it allows the right blood's flow between atria and ventricles. The PR interval represents this AV delay and part of atrial repolarization. The normal duration of this interval is from 120 to 200 ms.
- The PR segment represents the impulse's transmission from the AV node the the Purkinje fibers and it is located between the P wave's end and the begin of QRS complex.
- Purkinje fiber and ventricular myocardium produce the QRS complex. It is composed of three waves, Q, R and S (which may be absent in some leads) and it is the main complex since it represents the ventricular depolarization, i.e., the effective heartbeat. Its normal duration is from 80 to 100 ms.
- Ventricular repolarization produces J-point, ST segment and T and U waves [26]. In particular, J-point represents the end of QRS complex and the beginning of ST-segment. ST-segment is a neutral period between ventricles' depolarization and repolarization, and it is placed between the end of QRS

complex and the beginning of T wave. During this period the heart expels most of the oxygenated blood, remaining contracted. Finally, the T wave represents the ventricles' repolarization, and it has a normal duration of around 160 ms.

- All events of ventricular systole are represented by the QT interval, which has a normal duration < 440 ms and is located between the beginning of QRS complex and the T wave's end. Its duration can vary with the age, gender and heart rate.

1.2.2 ECG Grid

The ECG signal is generally printed on a special paper that contains a grid, shown in Figure 1.10. This paper is divided into grid-like boxes of 1 mm^2 size and the paper speed is around 25 mm/sec . At the end, the small horizontal boxes, which have a size of 1 mm , corresponds to 0.04 sec , so to 40 ms . Darker lines that form the bigger boxes are composed of five small boxes, so they represent intervals of 0.20 sec (200 ms).

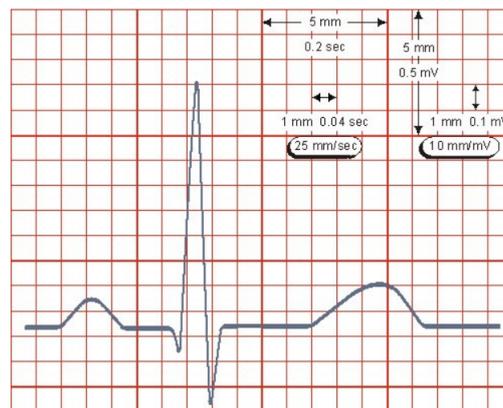


Figure 1.10. Grid Scale for ECG Wave Interpretation.[9]

Sometimes, to better define the waveforms, the paper speed can be increased to 50 mm/sec . In this situation, instead of 12-leads, there are only six leads, and the small boxes have a size of 0.02 sec while the large ones have a size of 0.10 sec .

Since in this case the speed is twice the previous speed, also the ECG intervals will be twice as long as normal, while the heart rate will be one-half of what is seen with 25 mm/sec speed. On occasion, also other paper speeds may be used.

The ECG graph, in vertical, measures the amplitude (height) of a wave or a deflection. Generally, it has a standard calibration of 10 small boxes, which

correspond to 10 mm, so to 1 mV. Sometimes, when the waveforms are small, a double standard is used, where 20 mm are equal to 1 mV [27].

Voltages and speeds are usually printed on the paper's bottom for reference.

1.3 Cardiovascular Diseases (CVDs)

Disorders that involve the heart and the cardiovascular system are called Cardiovascular Diseases (CVDs). These disorders are the first cause of death in the world, involving around 32% of all global deaths. The most of these deaths are due to heart attack and stroke (around 85%). Among 17 million of premature deaths (under the age of 70), in 2019, CVDs caused around 38% of them [28].

These kinds of diseases may be due to genetic risk factors. Sometimes, also a single gene's defect can involve all parts of the heart's structure. An interesting observation is that a genetic mutation in a household, is not carried but exhibits clinical variability (can range from symptom-free to premature death) [29].

In addition, it is known that men develop cardiac disorders 10-15 years earlier than women [30]. The hypothesis for understanding this difference is that the endogenous estrogen produced by women is cardioprotective and, in fact, the coronary heart disease rates increase in women after menopause or ovariectomy [31].

The risk factors of these diseases are principally: the use of tobacco and alcohol, an unhealthy diet or the absence of physical activity, which leads to obesity. These factors can lead to an increasing of blood pressure (hypertension), blood sugar (diabetes mellitus) or blood lipids (dyslipidaemia). Drug therapy (aspirin, beta-blocker or diuretic) can reduce myocardial infarction of 75%.

When the heart's functionality is not in normal conditions, there is a presence of a heart disease, which blocks blood vessels and can lead to heart attack, angina and stroke. The heart diseases found in new-born babies are called congenital, while those ones found at later ages, are called acquired. A lot of possibilities and challenges are designed for finding these diseases in early stages.

Heart disease's most common types are:

- Coronary Artery Disease (CAD): problems with blood vessels due to smoking, high blood pressure, diabetes and high cholesterol. A symptom of CAD is the angina, a chest pain.
- Congestive Heart Failure (CHF): it happens when heart cannot pump a normal level of blood.
- Abnormal heart rhythms: problems due to the heart's electrical activity. This kind of problem makes the heartbeat too slow or too fast.

Most of the CVDs can be prevented and, their early detection is an important task nowadays.

Chapter 2

Principles of Deep Learning

2.1 General Introduction

The computer system's ability to simulate human cognitive functions, such as learning and problem solving, is called Artificial Intelligence (AI). It can be seen as a system that, using mathematics and logic, simulate the human brain's functions of learning information and making decisions based on the processed data.

An application of AI is Machine Learning (ML), a process that allows the computer to learn and improve autonomously, without direct instructions, using mathematical models. This technique essentially represents the machine's ability to learn from a series of data. It automates the analytical model's construction through statistical modelling methods and operational research. In practice, ML consists in the use of algorithms to analyse data, learn from that data and then make a determination or prediction about new data.

One of the machine learning's approaches is the Deep Learning (DL), which takes inspiration from the brain's structure, in particular from interconnection of its neurons (neural networks). The Artificial Neural Networks (ANNs) are mathematical models based on directed acyclic graph models, which consist of a variables' set and conditional dependencies. Deep Neural Networks (DNNs) are a very innovative tool because they work very well with complex data and can be easily updated with other new data through batch propagation. Many kinds of problems can be solved with DNN architectures, which can be simply adapted by changing number and structure of layers.

The use of DL can lead to the following main advantages:

1. Automated feature generation: DL can produce new features, without human intervention, starting from limited number in the training dataset. This function permits to perform complex tasks without additional extensive feature engineering.

2. Better works with unstructured data: the ability of ML algorithms to analyse unstructured data is limited while the training of a deep learning neural network with unstructured labeled data, can help to optimize businesses.
3. Improved self-learning capabilities: models become more efficient to learn complex features and to improve computational tasks (i.e., capability to execute many complex operations at the same time) thanks to the multiple layers in DNN. This happens thanks to the DL ability to learn from its own errors, verifying the accuracy of the predictions and adjusting them when necessary. Furthermore, in DL, the larger are the datasets, the higher is the accuracy.
4. Advanced analysis: the more effective processing models of DL can be seen when it is applied to data science. The DL ability to learn unsupervised leads to improvements in accuracy and outcomes.
5. Scalability: the ability to process huge amounts of data and to perform a large number of computations with time and cost saving, makes DL highly scalable. This characteristic impacts on productivity, modularity, and portability [32].

As said before, deep learning using artificial intelligence is represented by neural networks. All deep learning algorithms, in fact, use different neural network's types to perform specific tasks, as shown in the next section.

2.2 Types of Deep Neural Networks

Artificial Neural Networks take inspiration from the brain's biological neurons of the human body, which, under specific circumstances, activate and perform an action. They consist of multiple layers of interconnected **neurons** which are activated by an **activation function** that switches them. Each connection between neurons can transmit a specific signal from one neuron to another. Certain values are learned from the network in the training phase.

Practically, what happens is that each neuron receives multiplied versions of inputs and random weights. These first elements are added with a static bias value, which is unique to each layer. At the end, an activation function decides the final value to be obtained at the outside of the neuron. When the output is generated, the loss function is calculated, as input versus output, and the weights are adjusted in order to minimize the loss, through back-propagation. The whole operation focuses on finding the weights' optimal values.

DNNs are multi-layer fully-connected neural networks structured as shown in Figure 2.1. They consist of:

- An **input** layer.
- Multiple **hidden** layers.
- An **output** layer.

Every node of each layer is interconnected with every node of the next layer. The network can be deeper just by the increase of layers' number.

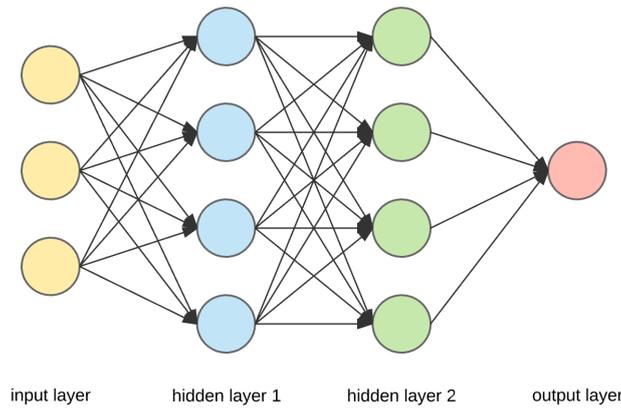


Figure 2.1. Artificial Neural Network Structure.[10]

Zooming into one of the hidden nodes, the following structure will be encountered. As it is possible to observe, in a neural network, each connection from one unit to another has its own assigned weight (a number between 0 and 1) which represents the strength of the connection between units.

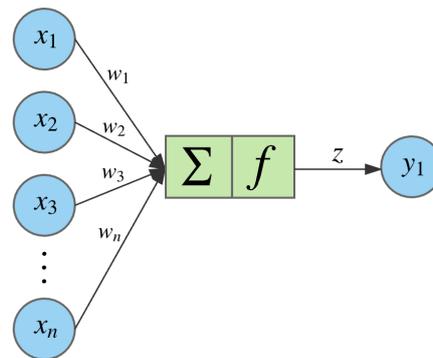


Figure 2.2. Hidden Node Structure.[10]

In particular, as shown in Figure 2.2, a specific given node takes the weighted sum of its inputs and passes it to the activation function (which is non-linear).

The output of the node becomes then the input of the node of the next layer and the signal flows from one node to the other until the final output is calculated. The **weights** (w_i) are numerical values which have to be multiplied with inputs, they are modified to reduce loss and they represent the values learned from neural network by the machine. So, training a deep neural network means to know the weights associated with all edges. The equation resulting for each node is written below. The weighted sum can be represented as a vector dot product with n equal to the number of inputs for the specific node.

$$z = f(b + x \cdot w) = f\left(b + \sum_{i=1}^n x_i w_i\right) \quad (2.1)$$

$$x \in d_{1 \times n}, w \in d_{n \times 1}, b \in d_{1 \times 1}, z \in d_{1 \times 1}$$

The **bias** term (b) has a value equal to 1 and it is an input to all the nodes. It has the purpose to shift the activation function's result and it helps the model training when all input features are 0. Finally, the working mode of a DNN is the following: each node receives information from data in the form of inputs, the inputs are multiplied by the node with random weights, which are then calculated; after this process, bias is added and finally the non-linear activation functions are applied to choose which neuron to fire.

DNNs are typically organized in layers. Different types of layers include:

- **Dense** (or fully connected) Layers: used frequently in models that link every input to every output within a layer.
- **Convolutional** Layers: used in a model that works with image data.
- **Recurrent** Layers: used in a model that works with time series data.
- Pooling Layers, Normalization Layers and many others.

In the following subsections, the main deep learning algorithms are explained and analysed in order to have an idea of what tasks they can perform.

2.2.1 Multi-Layer Perceptrons (MLPs)

Multi-Layer Perceptron models (MLPs) are the most basic neural networks (Figure 2.1) and they are a class of feedforward ANN. They consist of a series of fully connected layers and each of the new layers represents a set of non-linear functions of a weighted sum of all outputs coming from the previous one.

These kinds of models have the same number of input and output layers, but they may have multiple hidden layers. Their working mechanism is the following:

- MLPs feed the data to the network's input layer. The signal passes in one direction through the connected layers of neurons.
- MLPs compute the input with the weights between input and hidden layers.

- MPLs use activation function to fire the nodes.
- MPLs train the model to learn dependencies between variables of the training dataset [33].

2.2.2 Convolutional Neural Networks (CNNs)

Another class of DNNs is constituted by Convolutional Neural Networks (CNNs, or ConvNets), which are inspired to the human visual cortex. These kinds of networks are mainly employed in computer vision and image recognition. CNNs take images or videos from real world and automatically learn to extract the features of the inputs that they receive to complete a specific task, such as image classification, image segmentation or face authentication.

In CNNs, one or multiple convolutional layers extract the simple features from input by executing operations of convolution. The fundamental element in a CNN is the convolutional layer. Each neuron in this layer only processes a small part of the image; in fact, in this kind of networks, input features are taken in batch (like a filter) and then, operations are computed multiple times to fully complete the image processing. Figure 2.3 shows the CNN schematic architecture.

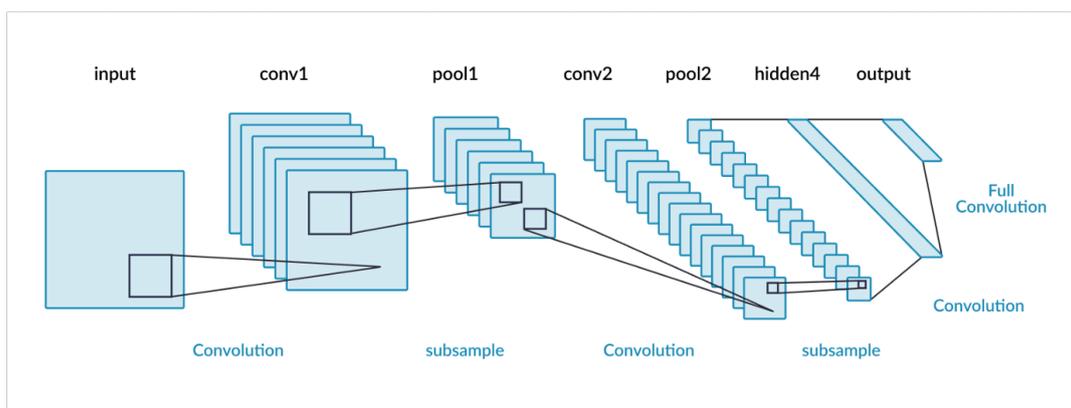


Figure 2.3. Convolutional Neural Network Structure.[11]

Since neurons in each layer are not connected to each single point but only to the points in their receptive field, this architecture allows the network to focus on small features in the first hidden level and then assemble them into larger items [34]. Convolutional kernels or filters are built at the bottom of the convolutional levels. During each passage, each filter is convoluted producing a 2-dimensional activation map, called feature map, of that filter. The neurons of the same feature map share the same weights and biases. When the network detects a specific feature in a spatial position in the input, it learns which filters are activated.

In order to reduce the computational load, the memory usage and the number of parameters, an important concept is the pooling layer, which is a non-linear downsampling. It has a receptive field but no weights, so it just aggregates the inputs. Max pooling is the most common pooling function: it partitions input image into rectangles and, for each region, outputs the maximum [35].

2.2.3 Recurrent Neural Networks (RNNs)

An ANN in which the neurons' connections make a graph directed along a sequence is called Recurrent Neural Network (RNN). With this neural network it is possible to reconstruct a phenomenon's dynamic temporal behaviour. For input sequences processing, RNNs use internal memory, which makes them applicable to activities that extend over a period of time such as handwriting and speech recognition or automatic translation. In RNN, the relationships between the elements are maintained during training so, all inputs are related to each other. These networks model a function that can provide output based on input.

Long Short-Term Memory (LSTM)

The main type of recurrent network is the Long Short-Term Memory (LSTM), which manages information in memory for a period longer than RNN. So, LSTMs can maintain long-term temporal dependencies but remember just the most important information. These networks are based on special units: the memory blocks. The difference between a RNN simple cell and the LSTM cell is shown in figure below:

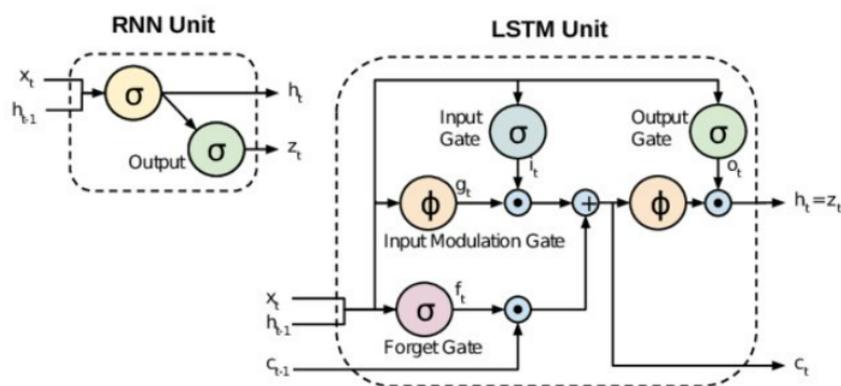


Figure 2.4. RNN simple cell versus LSTM cell.[12]

The hidden state h_t represents the working memory, while the gates regulate the cell's state. The forget gate drops some memories and the input gate replaces these memories with new ones. The rule of the forget gate is given by the following equation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_t) \quad (2.2)$$

where W is the vector of weights, b_t represents the bias, σ is the sigmoid function and t is the current instant.

One copy of the new state is passed to a tanh function and filtered by output gate, as shown in 2.3, the other one is sent to the next iteration.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (2.3)$$

New output is created by combining this result with the current inputs and the previous output. The hidden state is used for predictions, contains information on previous inputs and its output is equal to:

$$h_t = o_t * \tanh(c_t) \quad (2.4)$$

Important inputs are recognized by the input gate and stored into the long-term state. Inputs that are no longer needed are deleted by the forget gate, while the output gate decides when an input from long-term state has to be extracted [36]. The layer controlled by the tanh function calculates the new memories.

The short-term state is constituted by current input and previous output, and it is fed to four different fully connected layers. The layers controlled by sigmoid function control the gates, they have a range from 0 to 1 and are fed to element-wise multiplication operations. If the output is zero, the gate will be closed, if it is one, the gate will be opened.

2.3 Parameters Setup

The **parameters** are chosen by the model and they represent its coefficients. While learning, the algorithm optimizes these values and returns a parameters' array which minimizes the error.

The **hyperparameters** are element that have to be set. The model, in fact, does not update them according to the optimization strategy and they need manual intervention.

The **strategies** are the approaches towards the model. So, for example, normalizing data before managing them can help to improve the algorithm's performances.

2.3.1 Parameters

The only thing to do with parameters is to initialize them and the best way is to not set them equal to zero.

2.3.2 Hyperparameters

These elements need manually intervention. Some of them are:

- **Number of Hidden Layers:** the idea is to find a trade-off between keeping a model as simple as possible and having a model that classify well the input data. The best way to proceed is with manual attempts: after a certain number of layers, in fact, the accuracy stops to improve so, adding other neurons, is useful and just makes the algorithm heavy.
- **Learning Rate:** this rate refers to the backpropagation's step. In particular, the training process starts with arbitrarily set weights and then these weights are incrementally updated to minimize the loss. The size of these incremental steps represents the learning rate.
- **Activation function:** function through which the weighted sum is passed. The main activation functions are: ReLU, Tanh and Sigmoid. A variant of Sigmoid is the SoftMax, which is used for multiclass classification, and it returns as output a vector of probability with sum equal to one. Activation function can be placed in any point of the neural network and many times. These functions are showed below:

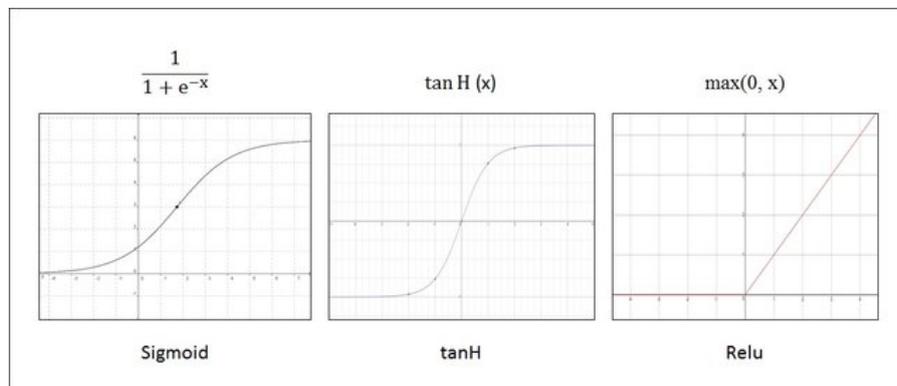


Figure 2.5. Main Activation Functions.[13]

Sigmoid is a complex function and requires more time, while ReLU is very fast. A good choice is to use ReLU for hidden layers and Sigmoid in the last one.

- **Epochs:** this hyperparameter represents the number of times for which the algorithm trains the whole dataset (it is different from the number of iterations, which is the number of batches necessary to complete one epoch). It is better to impose a condition such that epochs stop when the error is near to zero or, it is good to start with a little number of epochs and then increase it on the basis of some metrics, like accuracy.
- **Dropout:** this is useful to make the neural network not too heavy and it consists of removing some nodes during the training phase. It is better to keep each node with probability p , called 'keep probability' or to drop that one with probability $1-p$, called 'drop probability' [37].

2.3.3 Optimization Strategies

Optimization strategies are the best approaches to have a more performing algorithm. Some of these strategies are:

- **Parameter initialization:** as explained before.
- **Data Normalization:** which means converting all data to the same scale in the range 0-1. Another choice is to standardize the data making them normally distributed with a standard deviation equal to 1 and mean equal to 0. Another way to normalize data is to apply, after the weighted sum, 'Batch Normalization', and apply it also before the activation function.
- **Optimization Algorithm:** one of the main used optimization algorithm is Gradient Descent (GD) but there are many variants, such as: the Stochastic Gradient Descent (SGD), which minimizes the loss according to the GD optimization and, for each iteration, randomly selects a training sample; the RMSprop, in which each parameter has an adapted learning rate; the Adam Optimizer, which is often the best choice since it allows to customize the neural network, setting different hyperparameters.
- **Regularization:** optimal strategy to avoid overfitting and keep a simple model. Regularization adds a new term to the loss function, which tends to increase if the re-calibration increases weights. The main two types of regularization are: the Lasso (L1) and the Bridge (L2):

$$L1 : Loss = L(w) + \lambda \sum |w_i| \quad (2.5)$$

$$L2 : Loss = L(w) + \lambda \sum w_i^2 \quad (2.6)$$

L1 shrinks weights to zero (so, it risks getting rid of some inputs that will be multiplied with null values), while L2 shrinks weight to low values but different from zero (so, inputs are preserved) [37].

2.4 Metrics and Performances

The process of identification and aggregation of the elements of some data into predefined classes is called classification. This approach refers to a set of properties of the data elements, called features, and defined a-priori. In particular, at the beginning of this approach, the data are pre-classified, knowing the target values. By exploiting these values, the model's structures and patterns between input and output are detected and then the model can learn to classify correctly new data.

This thesis project is based on a classification problem and its aim is to correctly classify the ECG features, in order to detect the ECG waves and the relevant points of ECG signal.

The performances of a classification model are estimated by the analysis of **Confusion Matrices** (figure 2.6) and associated **AUC** (Area Under Curve). In particular, the performances are a measure of the number of times in which the network makes a correct classification, i.e., how many times a certain condition (event/non-event) is understood by the neural model.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Figure 2.6. Confusion Matrix.[14]

The values forming the confusion matrix are the following:

- **True Positive (TP)**: number of correct predictions that the result is positive, i.e., positive class correctly identified as positive.
- **False Negative (FN)**: number of incorrect predictions that the result is negative, i.e., positive class incorrectly identified as negative.

- **False Positive (FP)**: number of incorrect predictions that the result is positive, i.e., negative class incorrectly identified as positive.
- **True Negative (TN)**: number of correct predictions that the result is negative, i.e., negative class correctly identified as negative.

Other important advanced metrics, based on confusion matrix, are:

- **Sensitivity**: also known as **Recall** or as **True Positive Rate**. It measures the positive results correctly labeled as positive by the classifier and it has to be higher. In medical field, tests with high sensitivity are used for screening purposes. It is expressed by Equation 2.7

$$Sensitivity = \frac{TP}{TP + FN} \quad (2.7)$$

- **Specificity**: also called **True Negative Rate**. It measures the negative results correctly labeled as negative by the classifier and it has to be higher too. In medical field, tests with high specificity are used for confirmation purposes. It is expressed by Equation 2.8

$$Specificity = \frac{TN}{TN + FP} \quad (2.8)$$

- **Accuracy**: is the proportion of correct predictions with respect to the total number of predictions. It is expressed by Equation 2.9

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.9)$$

- **Positive Predictive Value (PPV)**: also called **Precision**, it is the ratio between the positive results classified correctly and the total number of predicted positive results. It is expressed by Equation 2.10 and gives the correctness of positive predictions.

$$PPV = \frac{TP}{TP + FP} \quad (2.10)$$

- **Negative Predictive Value (NPV)**: it is the ratio between the negative results classified correctly and the total number of predicted negative results. It is expressed by Equation 2.11 and gives the correctness of negative predictions.

$$NPV = \frac{TN}{TN + FN} \quad (2.11)$$

- **F-score**: it is calculated from the test's precision and recall. In particular, the harmonic mean of the precision and recall is represented by the F-score. It is expressed by Equation 2.12 and gives a measure of a test's accuracy.

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (2.12)$$

Receiver Operating Characteristic (ROC) curve and AUC

The Receiver Operating Characteristic (ROC) curve determines the classifier's capability to classify. It shows the TPR (Sensitivity) values versus the FPR (1-Specificity) at various cut-off values, as shown in Figure 2.7.

A perfect classifier presents no false negatives so, a value of 100% for sensitivity, and no false positives so, a value of 100% of specificity. A classifier without prediction capabilities produces a straight line, called 'line of no discrimination', passing through (0,0) and (1,1). The points on this line produce a random guess, the points above this line represent good results, while the points below this line represent bad results in classification.

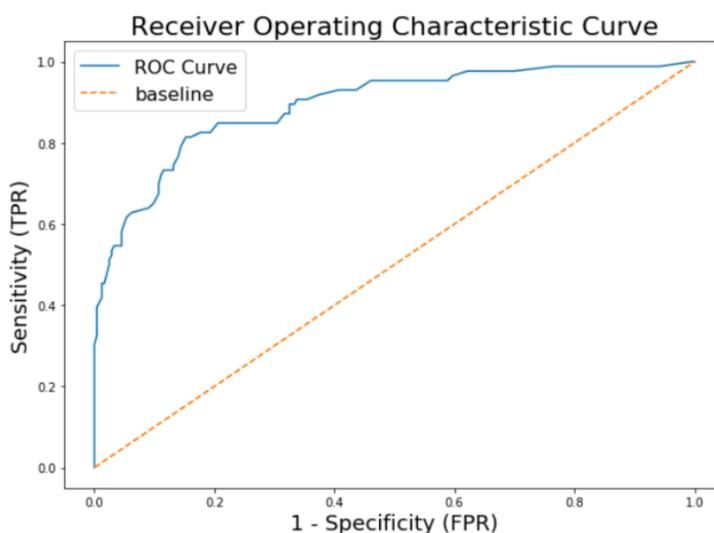


Figure 2.7. Receiver Operating Characteristic Curve.[15]

The AUC (Area Under Curve) represents the area under the ROC curve. It measures how good are the classifier's performances. For a random classifier, it has a value of 0.5, for the perfect classifier, it has a value of 1. Many classifiers assign to each class a number proportional to the probability that the point belongs to a certain target class. This classification depends on a threshold: the data above this threshold belong to the positive class, the data below belong to the negative one. The ROC represents points as function of this threshold and gives an idea of the model performances with different thresholds.

Precision-Recall Curve

The Precision-Recall (PR) curve (Figure 2.8) plots the Precision (PPV) versus the Recall (Sensitivity or TPR) as function of the threshold and it assesses the classifier's performances focusing only on the positive.

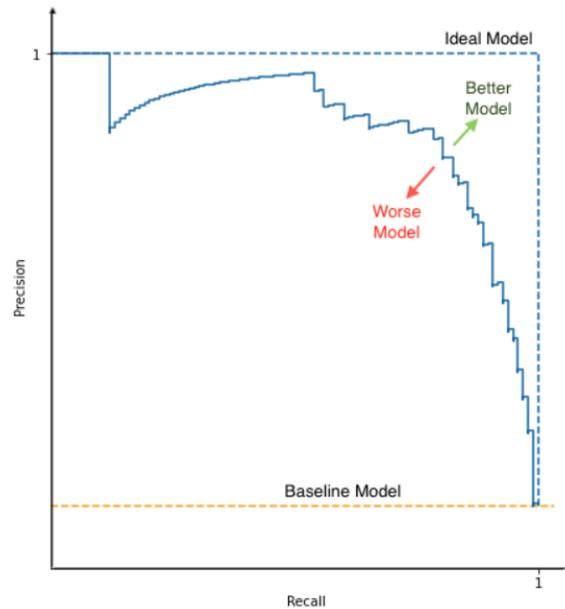


Figure 2.8. Precision-Recall Curve.[16]

So, both curves represent the TPR, one with the FPR (ROC) and one with the Precision (PR), indicating to which reference quantity compares the elements' total wrongly classified as positive (FP).

If the true negatives' number is more significant than true positives' numbers, the FPR can be small. This may happen when using easily classifiable data, in which case the precision is much more informative than FPR. If data are not classifiable, the precision is no more too informative.

In general, the advantage of the ROC curve is that it allows an intuitive interpretation. For example, an AUC of 0.7 means that the probability for classifier to correctly classify a data is of 70%. So, this graph tells the expected TPR and FPR for a specific threshold.

Chapter 3

State of Art

Algorithms in medicine have a lot of benefits for doctors and patients but to adjust these algorithms is a hard task. Since this is an innovative field, some of these algorithms have been approved by the United States Food and Drug Administration (FDA) but currently there are not universal guidelines for their approval. For clinical trials, in fact, FDA has very strict acceptance criteria and the algorithms, before being approved, have to show extreme transparency on scientific methods. Some algorithms are based on mathematics very difficult to decode, so they are considered as black boxes. Failures to clarify the algorithm's internal mechanisms impact the likelihood of the FDA approving trial [30].

The most relevant tool in diagnosis of diseases is medical imaging, which is user-dependent since it depends on human skills and interpretation in the acquisition and processing phases. For this reason, it may be potentially inaccurate. Automatic imaging analysis can overcome this problem thanks to adequately designed AI algorithms. The ML studies' relevance is related to the used model's design: an inefficient model design, in fact, does not produce generalizable outcomes. So, DL in medical imaging, with a correctly designed model, can produce a sensitivity and specificity similar to healthcare human professionals.

The AI application to the ECG can permit a diagnosis of not previously detectable conditions with an efficiency greater than before.

3.1 Deep Learning in Electrocardiography

Among the fields in which ML methods are applied, one important field is the electrocardiography. By the training of algorithms to learn classification functions from a set of labeled data, the researchers focused especially on identification of cardiac abnormalities. The main typical algorithms used are Support Vector Machine (SVM), Decision Trees (DT), Random Forests (RF), k-Nearest Neighbour

(KNN) and Artificial Neural Networks (ANNs), which include DL models such as Convolutional Neural Networks (CNNs) [38].

Most of ML algorithms are fed with handcrafted feature vectors, such as QRS duration, QT and RR intervals, morphological and frequency domain features, in order to accomplish the classification task. Thanks to the recent advent of DL, an innovative learning strategy has been developed. This strategy lets the classifier to directly learn relevant features from raw or slightly pre-processed data. ANN with multiple layers are the most common DL algorithms and perform specific operations such as convolutional filters in CNNs and subsampling, creating a model called Deep Neural Network (DNN). This kind of model already showed promising results for abnormal heart rhythms identification.

The electrical activity of heart can be seen with ECG, which comprises different waveforms representing the myocardium polarization or depolarization. The most prominent waveform is the QRS complex, which reflects the ventricular contraction. This complex, thanks to its specific shape, provides the most significant information in heartbeats' automatic detection.

The QRS complexes' detection is a difficult task, due to its physiological variability and to the presence of artefacts, such as baseline drift, power-line interference and activity of muscles. The traditional QRS detection methods can be seen in [39], while most recent QRS detection algorithms can be found in [40] and include Pan-Tompkins algorithm, algorithms based on derivatives and thresholds, or algorithm based on Wavelet Transform. Modern approaches, which include robust and well optimized linear methods, are ECG morphology compression, adaptive mathematical morphology or, as said before, ANNs.

Through AI algorithms it is possible to identify the current heartbeat and potential anomalies, such as episodic atrial fibrillation, structural dysfunctions (valvular heart diseases, ventricular low ejection fraction, channelopathies, cardiomyopathies). Medical professionals can identify some specific characteristics and related abnormalities in ECG traces but there may be huge discrepancies from the normal ECG for specific clinical conditions. Since the ECG is the cumulative recording of the action potential of cardiac cells and instrumental noise, the changes' magnitude can alter the ECG and be invisible for human eye. DL models can detect single feature anomalies not clearly visible and learn potential relations between feature variations.

By far, the most common architecture used for ECG analysis is the CNN. This kind of networks use the operation of convolution to localize key features and to reduce the noise. The term 'convolution' refers to the working mode of taking a small pattern, called 'kernel', and identify where this pattern arises in the input (similar to a sliding window). The 'heat map' helps the identification of where pattern is located in the image, which can be used to localize the most important features.

So, DL methods applied to ECG data have various clinical applications; in particular, five distinct medical applications are identified: CVD diagnosis, ECG analysis, Blood Pressure (BP) estimation, biometric recognition and sleep analysis. In the following sections, the main DL clinical applications are explained in order to have a general systematic review of the DL State of Art in Cardiology.

3.1.1 Deep Learning for ECG Analysis

Historically, the first data analysis tasks performed were the heartbeat classification and segment identification of P-QRS-T, achieved from a signal processing approach. Originally, these ECGs were decomposed, with Fourier and Wavelet transformations or with Hermite techniques, into wavelike components. This procedure may be seen as a feature extraction; for example, wavelet-based convolutional techniques achieved an accuracy of 93% on the MIT-BIH arrhythmia database.

In general, ML and DL models achieve better performances and are favoured. In this case, it is important to identify the best way to represent the signal relying on the task that has to be solved. As shown in Figure 3.1, the ECG signal can be subsampled into individual fixed-length heartbeats to generate many samples from which deriving the features. In addition, the signal can be sent as 2D boolean image formed by zeros or ones, which is suitable for diagnosing conditions and highly compatible with the traditional image-based CNN architectures. Depending on the leads' number, the signal can be one or multi-dimensional, allowing the capture of more information. Finally, the ECG can also be represented as a beats' sequence, linked each other in time, treated as a time series, which can be analysed by an RNN.

The representation type of ECG depends in particular on the available dataset. The first released database was the MIT-BIH AF, which contains 25 two-lead ECGs with a length of around 10 hours. The low number of ECGs can be compensated by their length: in fact, the signals can be subsampled in order to generate small ECGs that contain a beat and that can be used in the attempt of a perfect beat classification.

The Computing in Cardiology Challenge datasets introduced new datasets much larger, useful for novel tasks such as ECG abnormalities identification or AF classification. In addition, also the MIMIC database is useful, even if it has less clean and less extensive annotations. It offers a number > 67000 ECGs of ICU patients. However, the number of institutional datasets has also increased over the last half decade, surpassing the number of annotated ECGs in the open databases.

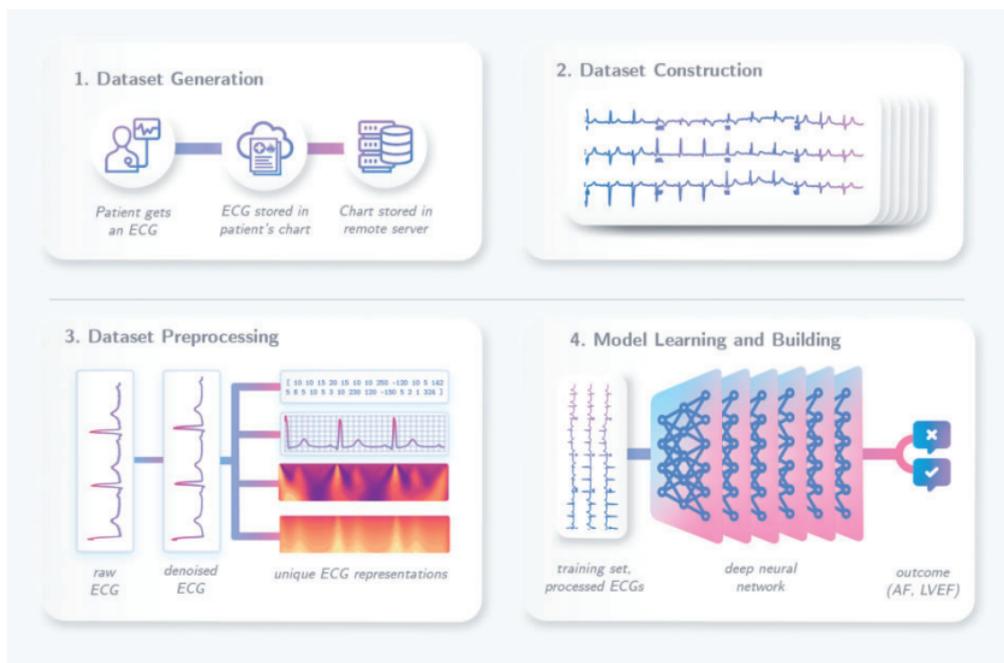


Figure 3.1. Simple deep learning pipeline for ECG analysis.[17]

3.1.2 Deep Learning for CVD Diagnosis

The most natural cardiac disorders in ECGs are given by the conduction system anomalies. Many works are dedicated to AF diagnosis, which is the most common arrhythmia, since it has a high prevalence of around 3% in adult population, while few works are dedicated to ventricular tachyarrhythmias diagnosis.

DL models have shown performances similar to medical professionals' level. Hannun et al., which cured a dataset of ECGs of patients in ambulatory setting [41], noted that the experienced cardiologists' annotations performed worse than the model's performances in detection of arrhythmias except for junctional rhythm and ventricular tachycardia. The same result was seen in a work of Ribeiro et al. [42], in which the performances of the DL model was better than those of medical trainees.

In particular, the model learns low-level features such as peaks, troughs, upward or downward slopes in the signal. This relates to the efforts of the model to remove baseline biases and identify key landmarks for diagnosis. The most classical of pattern recognition problems is the tackling arrhythmias but few works investigated their role in patients' management and their main characteristics for diagnosis purposes.

DL methods in CVD research are the best approach for prediction, management and diagnosis of a disease. Using DL techniques, also efficient data capturing from health records can be achieved.

For the MIT-BIH Arrhythmia Database, Wang et al. [43] achieved the best accuracy of 99.94% with a fused autoencoder-CNN model able to classify 6 different rhythms. For this kind of purposes, the most used model is the CNN structure. Lu et al. [44] and Yu et al. [45] used, in fact, a 1D-CNN to detect respectively arrhythmias (99.31% of accuracy) and premature ventricular contraction (99.7% of accuracy).

Other studies proposed a 1D-capsule NN for Coronary Artery Disease detection [46] and multiple convolutional and pooling layers for arrhythmia classification and myocardial infarction detection [47], reaching accuracy respectively of 99.44% and 99.90%.

Radhakrishnan et al. [48] and Petmezas et al. [49], used a 2D bidirectional LSTM network and a CNN-LSTM model respectively, for AF detection in ECG signals of public databases. In addition, Jahmunah et al. [50] used a CNN architecture for myocardial infarction, coronary artery disease and congestive heart failure detection, while Dai et al. [51] used a CNN for CVD diagnosis reaching very high accuracies.

So far, it is possible to note that most of the models for CVD detection involved CNNs or LSTMs and reached a very high accuracy (above 99%).

Chapter 4

Methodology

4.1 Google Colab

Since the Deep Neural Network's training requires a huge amount of time if performed on a low-end machine, this thesis project was elaborated on Google Colab, which makes available a virtual server with a powerful GPU (in particular, Nvidia Tesla k80 GPU), able to speed the training process.

The network is built in Python language with Tensorflow, an open platform end-to-end source developed by Google for machine learning. It has a complete and flexible ecosystem of community tools, libraries and resources which allow researchers to promote the state of the art in ML and developers to create and derive easily on ML. In Tensorflow, the inputs and outputs are made up of tensors: they are multidimensional arrays with a unified type (called dtype). All tensors are immutable like Python numbers and strings: the contents of a tensor cannot be updated, the only thing that is possible to do is just to create a new tensor. Tensors can be multidimensional.

4.2 Dataset Description

For this thesis purposes, the QT Dataset from Physionet was used [52][53]. This dataset contains 105 two-leads ECG recordings of fifteen minutes with P, QRS and T waves onset, peak and end annotations. These recorded ECG are sampled at 250 Hz and provided as digital signals. Every ECG sample, for this project, is divided into segments of length of 125 data points with an overlap of 10 data points between previous and next segment. The two leads are split into separate recordings, and both used as features. These 125 samples windows are used as input for the deep neural network.

Regarding annotations, within each recording, between 30 and 100 representative beats were manually annotated by cardiologists. They identify the beginning, the peak and the end of the P wave, the beginning and the end of the QRS complex, the beginning, the peak and the end of the T wave and (if present) the peak and the end of the U wave. Overall, 3622 beats were annotated. In the Table 4.1 the number of annotations is indicated for each of the features:

Waves	Annotations
P_{on}	2596
P	2626
P_{end}	2627
QRS_{on}	3130
R	3130
QRS_{end}	3130
T_{on}	1241
T	2932
T_{end}	2996

Table 4.1. Annotations of QT Database.

In particular, this database contains the following recordings (with related annotations):

- 15 recordings from the **MIT-BIH Arrhythmia Database**, which contains a total of 48 half-hour extracts of two-channel ambulatory ECG recordings;
- 6 recordings from the **MIT-BIH ST Change Database**, which includes a total of 28 ECG recordings of various lengths, most of which were recorded during exercise tests and which show transient ST depression;
- 13 recordings deriving from the **MIT-BIH Supraventricular Arrhythmia Database**, which contains a total of 78 half-hour ECG recordings on examples of supraventricular arrhythmias;
- 10 recordings from the **MIT-BIH Normal Sinus Rhythm Database**, which includes a total of 18 long-term ECG recordings of subjects without significant arrhythmias;
- 33 recordings deriving from the **European ST-T Database**, composed in total of 90 annotated extracts of outpatient ECG recordings of 79 subjects, contains a representative selection of ECG abnormalities such as the shift of the basal ST segment resulting from hypertension, ventricular dyskinesia, or drug effects;

- 24 recordings from patients with **BIH “sudden death”**;
- 4 recordings from the **MIT-BIH Long-Term ECG Database**, which contains a total of 7 long-term ECG recordings (14 to 22 hours each), with manually reviewed beat annotations.

MIT-BIH Arrhyt.	MIT-BIH ST DB	MIT-BIH Sup. Vent.	MIT-BIH Long Term	ESC STT	MIT-BIH NSR DB	Sudden Death
15	6	13	4	33	10	24

Table 4.2. Records’ distribution according to the original Database.

Around thirty consecutive beats of the dominant morphology are annotated in each track in order to permit the studies of beat-to-beat variations and, up to twenty beats of non-dominant morphologies are annotated in recordings with significant QRS morphology variations. The beats are annotated only in the final 5 minutes of recordings in order to permit to the algorithms a minimum of ten minutes for learning.

For the manual annotations, each expert added his estimated waveform boundaries and then, looking to both leads simultaneously, the time location of fiducial point is established. The aim of these annotations is to correct inconsistencies such as missing or misplaced annotations.

4.3 Signal Pre-processing

The recordings are extracted from QT Database through WFDB (waveform - database) tool and those ones not useful to the analysis are excluded. Some recordings have annotated complexes just in some few periods and not in the whole duration of the track. So, the areas without annotations have been excluded from the set of useful data.

At the beginning, the work started with the analysis of the most prominent and important feature, the QRS complex, which represents the heartbeat, i.e., the ventricular contraction. This complex is identified in the database with (‘N’) annotation. Therefore, excluding all the tracks that do not contain annotations of this complex, the main function, which extracts the data and annotations from the database and defines an associated index, has been defined. This index corresponds to regions where QRS annotations are present. The non-annotated zones of the dataset have been removed since they could create network problems. One of the signal with annotated QRS complexes is shown in 4.1.

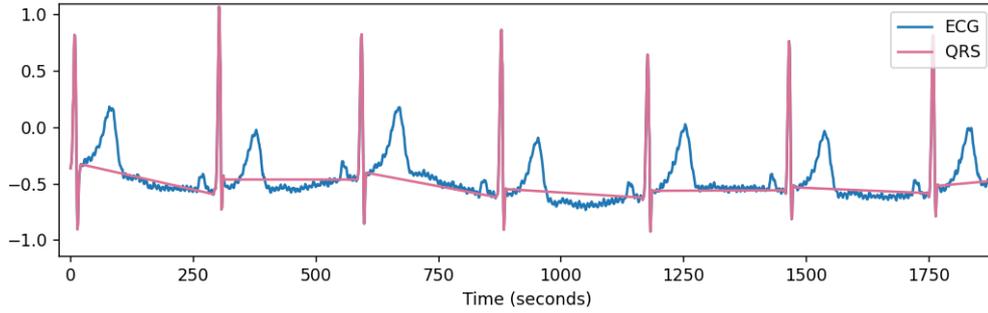


Figure 4.1. Example of signal extracted from 'QT Database' with annotations of QRS complexes.

The signals in the database are already quite clean, so it was enough to apply the linear regression method for the baseline wander removal to get good signals usable by the network.

4.3.1 Linear Regression Method for Baseline Wander Removal

Instead of using a FIR filter, which removes inevitably the frequencies of interest regardless of how well it is designed, the method of local linear regression was used. The idea is to calculate the linear regression over a window of about 1.5 seconds, then define the "baseline" as the centre of that window. After this, the window is moved forward of one point and the process is repeated again.

So, a robust locally weighted linear regression algorithm executes the removal of the baseline variation. In particular, the iterative algorithm takes ECG signal as input and assigns lower weights to the outliers of the regression. In this way, the main features relatively short-lasting such as P waves, QRS complexes and T waves become the outliers in the regression. Therefore, the algorithm converges to a solution which better represents the ECG signal's sole baseline.

This algorithm, due to its complexity, is used locally on windows of a duration of 1.5 seconds and this windowing reduces the time of execution without significant losses in information. The process is extremely efficient because the linear regression can be solved in an analytic solution in closed form, as can be seen in the following formulas.

$$WindowLength = N$$

$$LinearModel : \hat{y}_i = ax_i + b$$

$$\mathcal{L} = \sum_{n=1}^N (\hat{y}_i - y_i)^2 \quad (4.1)$$

$$\mathcal{L} = \sum_{n=1}^N (ax_i + b - y_i)^2 \quad (4.2)$$

, where y_i is equal to the ECG data from this window

$$\frac{\partial \mathcal{L}}{\partial a} = \sum_{n=1}^N 2(ax_i + b - y_i)(x_i) \quad (4.3)$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{n=1}^N 2(ax_i + b - y_i) \quad (4.4)$$

$$\text{Setting : } \frac{\partial \mathcal{L}}{\partial a} = 0, a \sum_{n=1}^N x_i^2 + b \sum_{n=1}^N x_i = a \sum_{n=1}^N x_i y_i \quad (4.5)$$

$$\text{Setting : } \frac{\partial \mathcal{L}}{\partial b} = 0, a \sum_{n=1}^N x_i + Nb = \sum_{n=1}^N y_i \quad (4.6)$$

In matrix form,

$$\begin{pmatrix} \sum_{n=1}^N x_i^2 & \sum_{n=1}^N x_i \\ \sum_{n=1}^N x_i & N \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{n=1}^N x_i y_i \\ \sum_{n=1}^N y_i \end{pmatrix}$$

$A \qquad B \qquad Y$

Solving for B:

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{1}{\text{Det}(A)} \begin{pmatrix} N & -\sum_{n=1}^N x_i \\ -\sum_{n=1}^N x_i & \sum_{n=1}^N x_i^2 \end{pmatrix} \begin{pmatrix} \sum_{n=1}^N x_i y_i \\ \sum_{n=1}^N y_i \end{pmatrix}$$

In Figure 4.2 is shown an example of signal before and after the pre-processing procedures of baseline wander removal. The blue signal represents the original signal extracted from the database, while the light blue one represents the signal after the pre-processing operations. As it can be clearly seen, with the removal of the baseline variation, the signal is centred on zero.

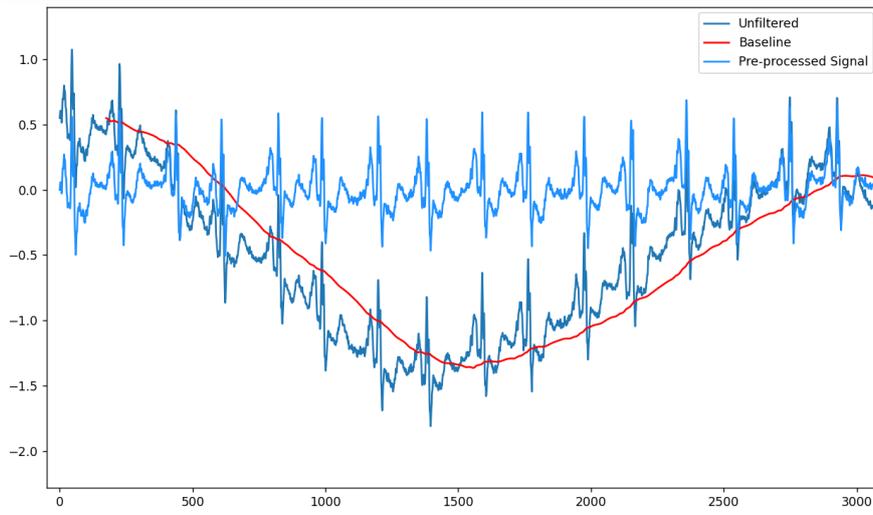


Figure 4.2. Example of signal before and after removal of baseline variation.

4.3.2 Scalogram Derivation

The Continuous Wavelet Transform (CWT) was used to derive the scalogram so that it is possible to transform the 1-D ECG signals into 2-D images. Then the signal is downsampled from 250 Hz to 125 Hz to reduce the computational complexity of the model. Since the ECG signal is made by different frequency components, it has been transformed in time-frequency domain in order to facilitate the features' extraction. The CWT is defined as follows:

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi\left(\frac{t-b}{a}\right) dt \quad (4.7)$$

$$a \in R^+ - \{0\}, b \in R$$

In other terms:

$$CWT(scale, position) = \int_{-\infty}^{\infty} f(t) * \psi(scale, position, t) dt \quad (4.8)$$

The terms of Equation 4.7 represent:

- a: it is a scale factor for resizing the function $\psi(t)$
- b: it is a displacement factor for translating the function $\psi(t)$.
- The result of the CWT is therefore a matrix filled with wavelet coefficients localized for scale and position.

The scalogram is a tool that builds and displays the 2D spectrum of the Continuous Wavelet Transform and represents the absolute value of the CWT coefficients

of the signal. Its colour represents the magnitude of the frequency components: a lighter colour represents the highest components while a darker colour represents the lowest ones.

CWT was calculated using the Wavelet Morlet which has a one-sided spectrum and has complex values in the time domain. Below a sample wavelet equation and its corresponding wavelet are shown (Eq. 4.9 and Fig. 4.3):

$$\psi(t) = e^{-t^2/2} \cos(5t) \quad (4.9)$$

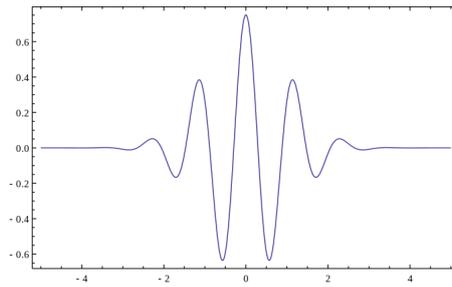


Figure 4.3. Wavelet Morlet.[18]

In this project, the CWT coefficients in the form of scalogram have the aim to become the image input of the deep neural network for features classification.

An example of scalogram with its associated ECG signal is shown in Figure 4.4:

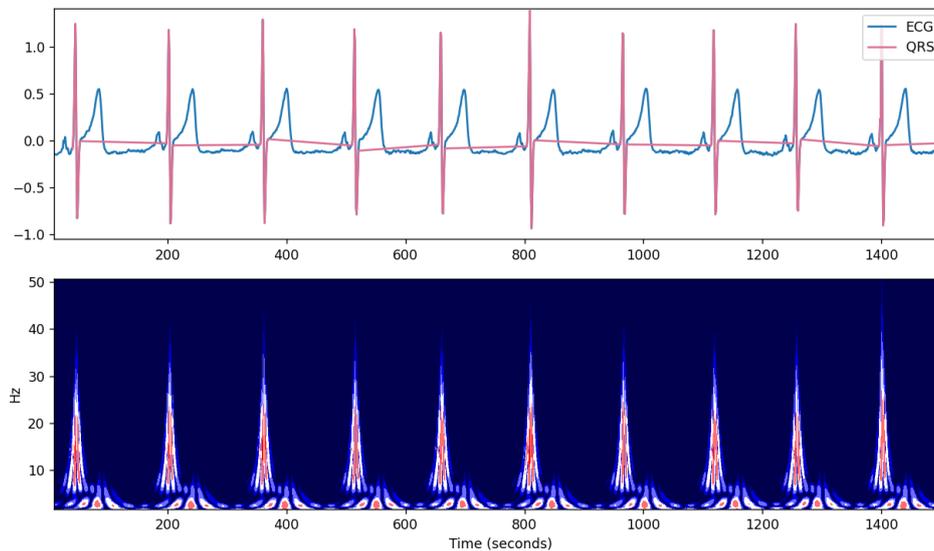


Figure 4.4. Example of ECG signal with its associated scalogram.

The colour variations in the scalogram correspond to the amplitude of the frequency components of the signal. In particular, a darker colour indicates a larger amplitude, while a lighter colour indicates a smaller amplitude.

The differences between the scalograms of a healthy subject and those of a subject with pathology lie instead in the different frequencies in which certain parts of the tracks are located. Some particular cardiac conditions present differences in scalograms of abnormal complexes and normal ones. These differences can be useful eventually for the neural network to detect anomalies of the heart rhythm. An example of what happens in the case of Premature Ventricular Contraction can be seen in the image below.

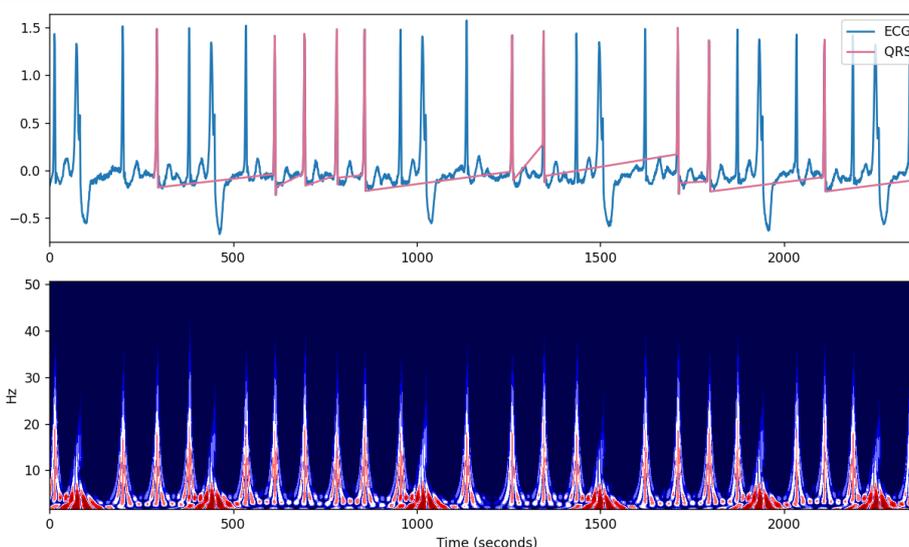


Figure 4.5. Example of Scalogram in presence of Premature Ventricular Contraction Event.

The frequencies in which the PVC event is located, in fact, are slightly lower than those in which the normal QRS complexes are located. As well as in this case, also other anomalies can be observed, such as Atrial Premature Beat, Supraventricular Premature Beat, Paced Beat and Right Bundle Branch Block Beat.

4.4 Data Segmentation

Dynamic DL models, such as CNNs, are used for feature extraction and, for training, require a lot of data. Sometimes, the signals sent to CNN input are lengthy so, the estimated performances may have problems due to degradation. In this situation, the ECG signals and their related label have to be broken to prevent this side effect.

The general strategy has been to create windows which have a length of 1 second (125 points, since the sampling frequency is 125 Hz), with a sliding length of 10 points. At the end of this procedure, approximately 15,000 data windows have been obtained, containing QRS complexes annotations useful to train the network.

Each of the windows contains a QRS complex or part of it with the associated annotation. The total number of windows obtained by this process is equal to 15089.

At the beginning, 90% of the previously obtained windows was used for the training procedure, while 10% was reserved for the testing procedure. This operation is repeated for both channels to get the same subdivision and then the windows are merged together. However, at the end, a k-fold Cross Validation (CV) was performed.

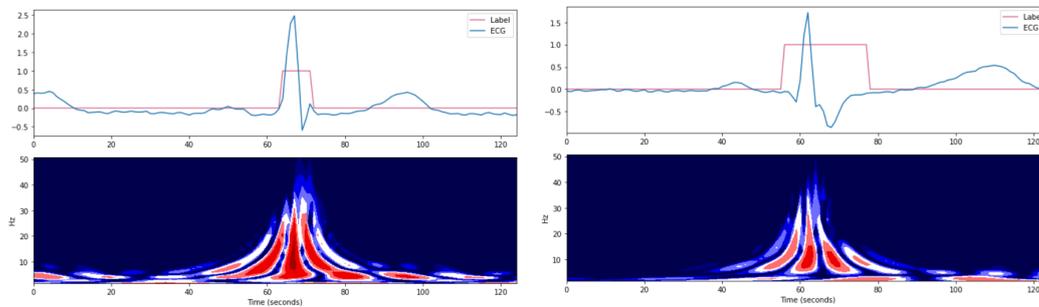


Figure 4.6. Example of ECG data segmentation: each window contains one QRS complex or part of it with its associated label.

4.5 Used Deep Learning structures

The implemented DNN model includes two Convolutional Neural Networks and two Bidirectional Long-Short Term Memory Networks. Let's see in detail how these models work, and which are the main characteristics and parameters involved in the DL models of this thesis project.

4.5.1 Convolutional Neural Networks' Layers

The CNN main functions are convolution, max pooling, classification, and non-linearity. The CNNs in this project have the purpose of extracting temporal features. Its main layers are:

- **A 2D Convolutional Layer:** it is responsible for generation of feature maps from 2D filters.

- A **ReLU Activation Function**: it is critical and its aim is to choose if an information coming from one neuron is useful or not. The ReLU function works as an activation mechanism; it is a non-linear function, which reduces the image's linearity through the deactivation of neurons when they have values less than zero. In particular:

$$Y = ReLU \sum((W * X) + b) \quad (4.10)$$

, where Y is the output value, W are the weights, X represents the inputs and b is the bias.

- A **Batch Normalization Layer**: the previous parameters can affect the distribution of the input of the following layer. The most important layer is the batch normalization because it normalizes the output of the last layer and behaves as a regulator to avoid overfitting. This parameter estimates the input batches' mean and variance. Then, it scales, normalises, and shifts them.
- A **Max Pooling 2D Layer**: it is used for input matrices' downsampling and for reduction of dimensions.
- A **Dropout Regularization Layer**: it avoids the overfitting problem during the model training by the discard of some nodes and the reduction of dependencies between them. A Dropout of 25% is used in this model.

4.5.2 Bidirectional Long Short-Term Memory Neural Networks' Layers

For transition from CNN to LSTM, it is necessary to do a time-distributed flatten in order to transform a 2D matrix into a 1D vector, which can be fed in a fully connected layer. The LSTM neural networks are able to predict the future QRS complexes, thanks to the previous ones since they can deal with ECG features related to time series. The implemented structure consists in two LSTM layers, each one with a dropout layer with 25% drop. The proposed method has two classes as output: QRS complex or not QRS complex.

The idea of Bidirectional Neural Networks involves the duplication of the first recurrent layer in order to obtain two layers side-by-side. Then, the input sequence is provided to the first layer and a reversed copy of it is provided to the second one. In this project, the first hidden layer of the first LSTM has 100 memory units, while the first of the second LSTM has 50 memory units. The output goes through another time-distributed wrapped layer in order to obtain one predicted value for each time-step. Then, two other layers for Batch Normalization and Dropout Regularization (25%) are involved and the final output will be a dense layer with the softmax activation function used to predict the final binary value.

4.5.3 Training of the Model

The training process of the model is divided in two passways: the forward and the backward propagation.

- The **forward propagation** is the phase in which the network processes the inputs thanks to the multiple neurons organized in layers and returns the result of the processing operations on the output layer. This propagation process includes the feed of inputs in the first neurons' layer, the processing of these inputs by the activation function and the feed of output in the next neurons' layer until the reaching of the final output layer.
- The **backward propagation** is the phase in which the difference between the predicted output and the actual target values is minimized. This phase, travelling back through the network, reduces the error by minimizing the weights' neurons that contribute mostly to this error. This error management through the update of the weights is called BackPropagation.

An epoch, or training iteration, is one round of forward and back propagation. The chain rule of differentiation is implemented by the backpropagation algorithm, and it has the aim to calculate the gradients of cost function with respect to its parameters in each layer. So, BackPropagation is the recursively gradient computation from the output layer backwards through the network. It calculates, through the chain rule, the function's derivative given by another function's derivative. The gradient represents the partial derivatives' vector of a target function with respect to the input variables, as shown in Equation 4.11. The first order derivatives of a function represent the slope (a positive derivative represents an upward function's slope).

$$Gradient = \frac{\partial \mathcal{L}}{\partial w} \quad (4.11)$$

, where \mathcal{L} represents the loss function and w represents the weights, which make this function variable.

As the gradient increases to a lower target function, a downward slope follows to carry the negative of the gradient forward (as shown in Figure 4.7). Each step of gradient estimation is called learning rate. This process continues iteratively until the target function's minimum is found or other stopping conditions are met.

When the training process starts, the weights of each layer are randomly initialized. Then, in a forward way, the inputs are processed to obtain the outputs. These outputs are compared with the target value to compute the error, and, at the end, the error's gradient is computed with respect to each weight through the differentiation chain rule. The weights' update is proportional to the learning rate in the direction of error gradient's minimization.

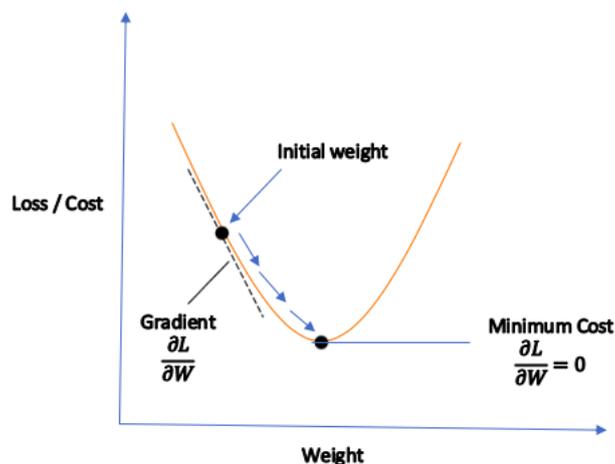


Figure 4.7. Gradient Optimization Process.[19]

Various optimization algorithms can be introduced to reduce the number of iterations necessary to find the correct weights' combination for the minimization of errors between actual outputs and predicted ones, such as **Gradient Descent** (GD) or **Stochastic Gradient Descent** (SGD). The GD performs a backpropagation on the network for finding the loss function minimum. The SGD, which represents a stochastic approximation of GD optimization, is an adaption of GD for the minimization of the loss function of a predictive model. It reduces the computational effort of the optimization problem by replacing the gradient calculated on the whole dataset with an estimation calculated on randomly selected data subset.

4.5.4 Optimizer and Activation Functions

Optimizer

So, finally, in this model, the **categorical cross-entropy loss** is used and the chosen optimizer for the model's training is the **Adaptive Moment Estimation**, also known as **Adam**, which gives a good computational efficiency. This efficient optimization algorithm is used for finding the weights. For each epoch, the accuracy, loss and AUC metrics are calculated.

The Adam optimizer can be used as another variant of the Stochastic Gradient Descent (SGD) algorithm in order to update the iterative weights of network in the training data. SGD maintains, for each weight update, a single learning rate, which doesn't change during the training procedure. Adam, instead, adapts the

learning rate to the single parameter through estimation of the gradients' first and second moments. It can be seen as the combination of the advantages of two others SGD extensions:

- The **Adaptive Gradient Algorithm** (Adagrad): which, at each time step and for each parameter, updates the learning rate based on the gradients calculated for the individual parameter up to that time step. It improves performances on problems thanks to the sparse gradients and allows to avoid the learning rate manual tuning.
- The **Root Mean Square Propagation** (RMSProp): which, for each parameters, updates the learning rate based on the average of gradients' recent magnitude for the weight. So, this algorithm works well on non-stationary problems.

Adagrad has the squared gradients accumulation as main drawback. In fact, in this method, the learning rate can become small and can encounter the vanishing gradient problem.

Anyway, Adam merges Adagrad and RMSProp benefits. In particular, it uses the average of the gradients' second moments (uncentred variance) instead of adapting the learning rate on the base of average first moment (mean) such as in RMSProp. Specifically, this algorithm computes a gradient's exponential moving average and the squared gradient, while its parameters control the rates of decay of these moving averages.

Activation Functions

First of all, the **Rectified Linear Unit** (ReLU) function is used in the two CNNs as activation function. It is the positive part of the function's argument (as expressed in Equation 4.12 and in Figure 4.8).

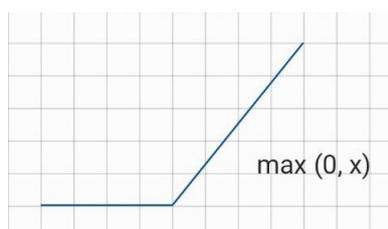


Figure 4.8. ReLU Activation Function.

$$f\left(\sum_{i=1}^n w_i x_i + b_0\right) = \begin{cases} 1, & \text{if } w_i x_i + b_0 \geq c \\ x, & \text{if } w_i x_i + b_0 < c \end{cases} \quad (4.12)$$

This activation function does not activate simultaneously all the neurons, but it activates a neuron only when the linear transformation's output is more significant than zero. The deactivated neurons, however, has a gradient value of zero. So, these neurons' weights and biases are not updated, and these neurons are never activated.

Finally, the **Softmax** activation function is used to measure the occurrences probability distributions. In particular, it computes the probability (ranging from 0 to 1) of a data to belong to a certain target class. This function is expressed in the Equation below:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (4.13)$$

The activation function is one of the hyperparameters which has to be set manually by the user and it depends on the specific task of the single model.

4.6 k-fold Cross Validation

The k-fold Cross Validation is the method employed in this project to compute the metrics and evaluate the performance of the model.

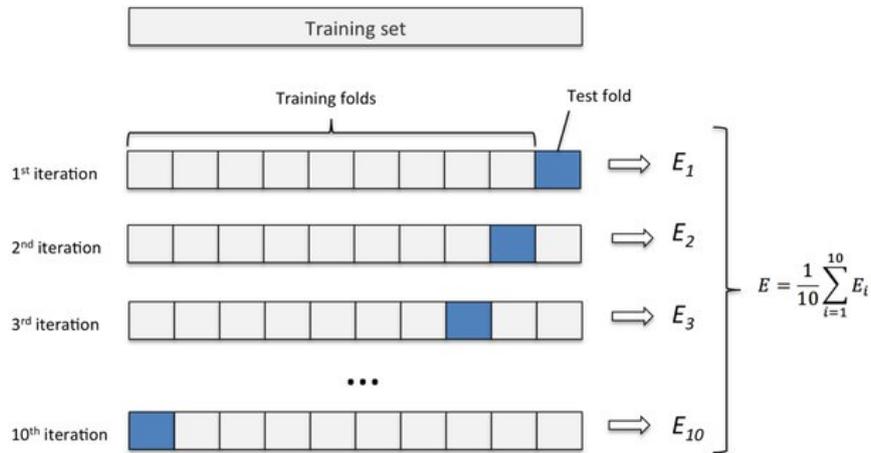


Figure 4.9. k-fold Cross Validation.[20]

This kind of approach assesses the model performances by using a validation set without involving the test set, which has to remain independent in order to be used after the optimization process. This Cross-Validation model is employed as

a validation technique, and it uses randomly split data for iterative training and validation of the model. Its working principle is shown in Figure 4.9.

First, the training set is split in k parts, called folds, with $k = 10$ in this specific case. The model is then trained k times on $k-1$ subsets and validated on the left part. In this way, each subset used for validation is different for each iteration. The estimate of the produced out-of-sample error of each iteration is averaged and, finally, the Cross Validation (CV) error is obtained. This method allows a validation of performances while the model is still training.

4.7 Other Features Extraction

The same procedures and methodologies described before, are applied for P and T wave detection.

The image below shows an example of signal extracted from 'QT Database' together with its labels for P waves. The annotations indicate the beginning and the end of P waves.

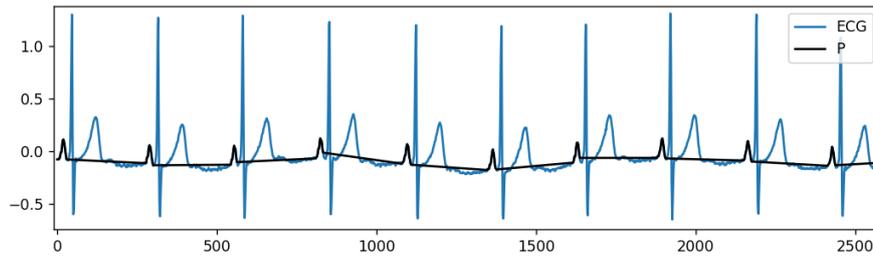


Figure 4.10. Example of signal with annotation of P Waves.

The same pre-processing procedures followed the signal analysis until, as in QRS complexes, the signal segmentation is executed as shown in Figure below.

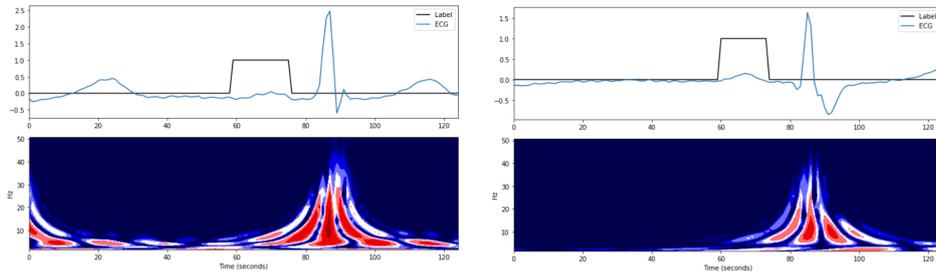


Figure 4.11. Example of ECG data segmentation for P waves: each window contains one P wave or part of it with its associated label.

The image below shows an example of a signal extracted from 'QT Database' together with its labels for T waves. The annotations indicate the beginning and the end of T waves.

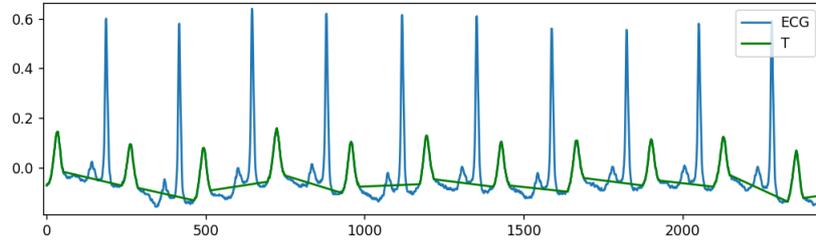


Figure 4.12. Example of signal with annotation of T Waves.

The same pre-processing procedures followed the signal analysis of T wave until the signal segmentation is executed also for this feature.

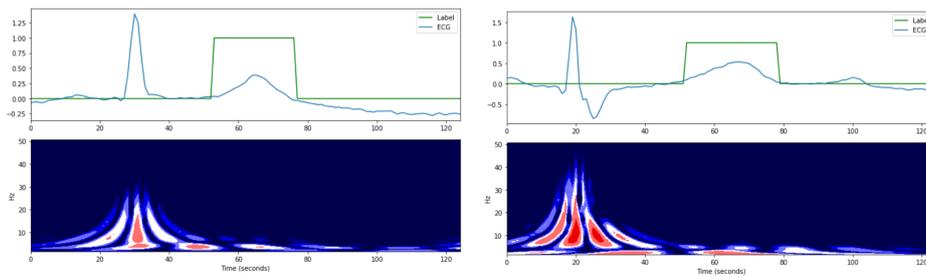


Figure 4.13. Example of ECG data segmentation for T waves: each window contains one T wave or part of it with its associated label.

Chapter 5

Implementation and Results

5.1 Deep Neural Network Implementation

In this chapter, the DNN implementation will be shown together with the obtained metrics and results of the three Neural Networks' models trained respectively for P, QRS and T features of ECG signal. This section will start with a description of the sequential layers which form the proposed model (as described in Section 4.5).

Layer (type)	Output Shape	Parameters
conv2d (Conv2D)	(None, 125, 62, 32)	832
activation (Activation)	(None, 125, 62, 32)	0
batch_normalization (BatchNormalization)	(None, 125, 62, 32)	128
max_pooling2d (MaxPooling2D)	(None, 125, 15, 32)	0
dropout (Dropout)	(None, 125, 15, 32)	0
conv2d_1 (Conv2D)	(None, 125, 15, 32)	25632
activation_1 (Activation)	(None, 125, 15, 32)	0
batch_normalization_1 (BatchNormalization)	(None, 125, 15, 32)	128
max_pooling2d_1 (MaxPooling2D)	(None, 125, 3, 32)	0
dropout_1 (Dropout)	(None, 125, 3, 32)	0
time_distributed (TimeDistributed)	(None, 125, 96)	0

bidirectional (Bidirectional)	(None, 125, 200)	157600
bidirectional_1 (Bidirectional)	(None, 125, 100)	100400
time_distributed_1 (TimeDistributed)	(None, 125, 5)	505
batch_normalization_2 (BatchNormalization)	(None, 125, 5)	20
dropout_2 (Dropout)	(None, 125, 5)	0
time_distributed_2 (TimeDistributed)	(None, 125, 2)	12

Table 5.1: Deep Neural Network Implementation.

The column of parameters in the table indicates the parameters' number that each layer has. First of all, the model input is a 2d-image of size 125x62x1, which represents the scalogram of the ECG signal from both channels of the Database. This input is fed in the model's input layer and then it passes through a 2d-convolutional layer. The convolutional layer has 32 filters, kernel size equal to 5 and produces an output of shape 125x62x32. Then the data go through the activation function, the batch normalization and the max-pooling 2d layer, which has a pool size equal to (1, 4), producing an output of shape 125x15x32. This output is then passed to the dropout layer until it reaches the second 2d-convolutional layer. This other convolutional layer has the same filters, the same kernel size and executes exactly the same operations of the previous one. The data pass again through the activation function, the batch normalization and the max-pooling 2d layer (with the same pool size as before), producing an output with shape 125x3x32, which is passed to the dropout layer.

The output of all these layers is subjected to a time-distributed flattening (as transition from CNN to LSTM) and then passed to a Bidirectional LSTM layer with a total of 200 LSTM cells with 100 LSTM cells in each direction. Then, another Bidirectional LSTM layer is used, which contains 100 LSTM cells (50 in each direction). This architecture is called 2D-CNN-BiLSTM. The output of the final bidirectional layer is fed to a time-distributed dense layer, which is followed by another batch normalization layer and a dropout layer. The last time-distributed dense layer assigns one of the two labels to each of the data points of the entire ECG window signal. In particular, the DNN's output is 1 or 0, depending respectively on whether or not the network finds a QRS complex.

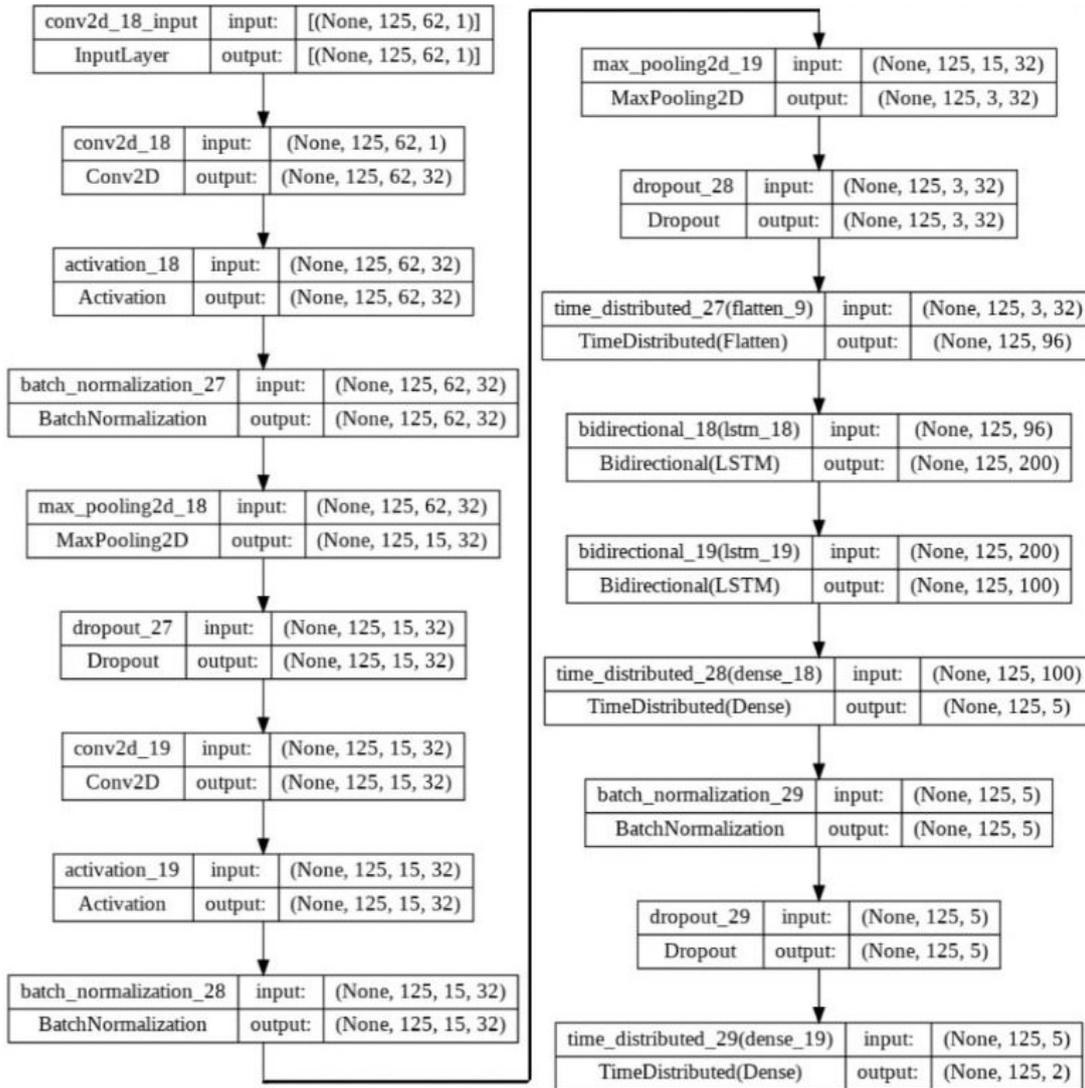


Figure 5.1. Deep Neural Network Model Architecture.

The DNN proposed architecture is implemented in Python using TensorFlow, an open-sourced end-to-end platform, and Keras, a high-level NN library that runs on top of TensorFlow. Some of the functions, such as the k-fold CV, are implemented using scikit-learn library.

To avoid overfitting, the minimum change of validation loss calculated by the model in each training step is set, monitoring this quantity for early stopping. This mechanism stops the training process when the model's performances on the validation dataset do not improve with respect to the defined metric (in this case

the loss). In particular, if the absolute change of the validation loss is less than the imposed delta equal to 0.001, the step is considered to have no improvements, so the mechanism stops the training process earlier.

For better training purposes, the k-fold Cross Validation (with $k = 10$) is used. The validation dataset is composed of samples randomly selected from the dataset and it is different for each of the 10-folds of the CV. The monitored metrics for each of the folds are accuracy, loss and AUC while the calculated evaluation metrics are the ROC, the Sensitivity, the Specificity and the F-score.

The model is trained over 30 epochs and the categorical cross entropy is used as the loss metric throughout the training process.

In the following sections, the metrics and results obtained after model's training will be analysed, firstly for each of the single trained networks (QRS, P and T detectors) and finally for all the models used together.

5.2 Results for QRS Complexes Detection

In the table below, the results of k-fold CV are shown for DNN model which detects QRS complexes. The dataset is divided into folds, and, for each iteration, k-1 folds are used for training and one for validation. In particular, table 5.2 shows the Accuracy, Loss and AUC results of each of the folds used for Cross Validation.

QRS Results	Loss	Accuracy	AUC
Fold-1	0.0593	97.505%	99.767%
Fold-2	0.0681	97.172%	99.684%
Fold-3	0.0622	97.408%	99.733%
Fold-4	0.0775	96.787%	99.592%
Fold-5	0.0697	97.067%	99.672%
Fold-6	0.0633	97.427%	99.718%
Fold-7	0.0713	97.021%	99.668%
Fold-8	0.1038	95.871%	99.325%
Fold-9	0.0756	96.903%	99.616%
Fold-10	0.0763	96.814%	99.611%

Table 5.2. Results of k-fold Cross Validation for QRS Detection.

The learning curves plot the accuracy and the loss of the model against the training epochs. Training the model for 30 epochs is enough for the model to converge. In Figure 5.2, it is possible to observe that the accuracy training and validation both increase and stabilize at a specific point around 0.97 (and around epoch 14). The same happens in Figure 5.3, where the loss training and validation

both decrease until a point near to zero. This behaviour indicates an optimal fit, i.e., the model does not overfit or underfit.

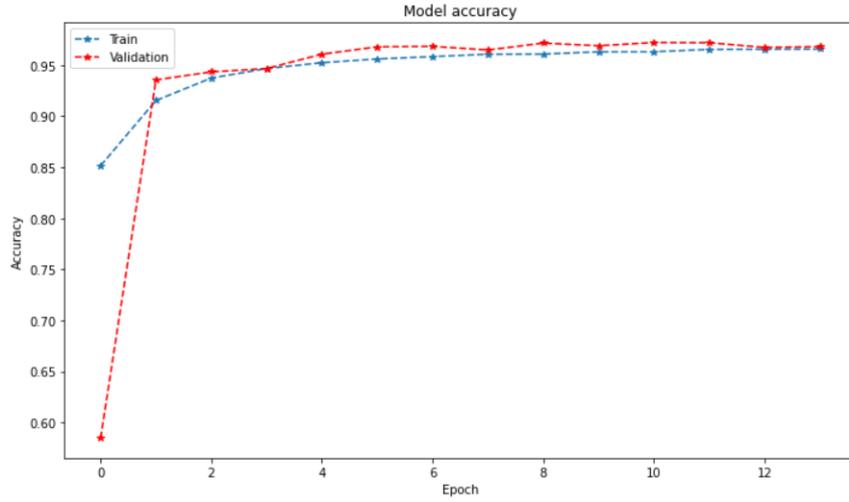


Figure 5.2. Accuracy for QRS Detection during each epoch.

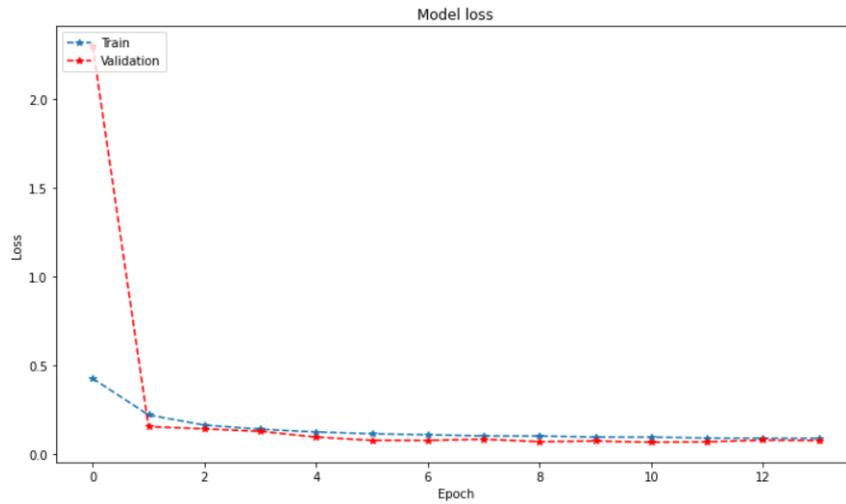


Figure 5.3. Loss for QRS Detection during each epoch.

	Loss	Accuracy	AUC
Average Scores for all folds	0.0727	96.997%(±0.447)	99.639%

Table 5.3. Average Scores for all folds of CV.

The area under the ROC curve (which is shown in Figure 5.4) is the AUC. The AUC provides a measure more accurate of prediction accuracy (the higher is the better is) because it indicates the probability that a randomly selected positive sample is higher than the corresponding negative sample.

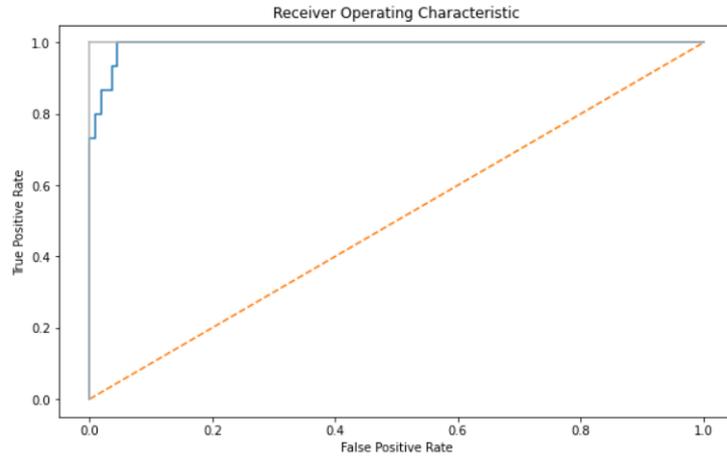


Figure 5.4. ROC for QRS Detection.

The validity of the approach can be tested on some random data. In particular, in Figure 5.5, the predicted outputs of the model are plotted for some test inputs. Since this is a binary classifier, just the first column of probabilities is taken. Then an optimum threshold value is applied to the obtained probabilities in order to get a coherent result.

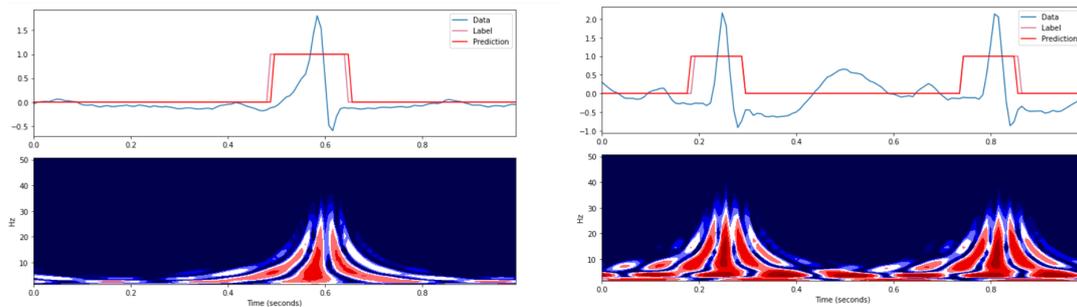


Figure 5.5. Model's predictions on new signals.

The classification performances are estimated analysing the confusion matrix, shown in 5.6. This matrix is normalized and obtained by dividing each column element by the sum of the entire column.

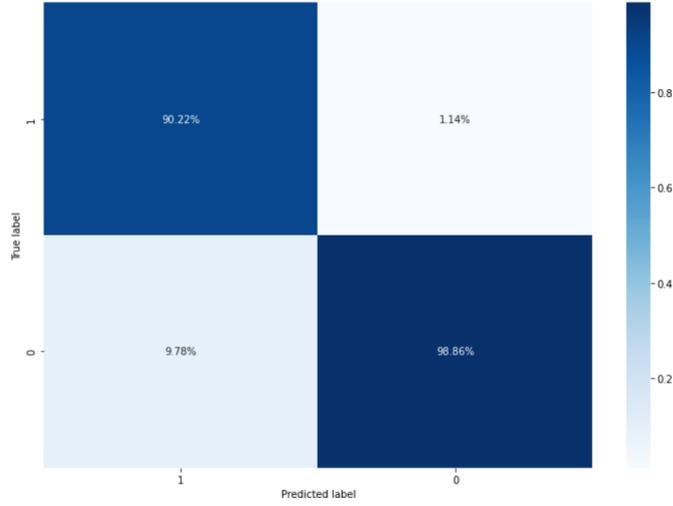


Figure 5.6. Confusion Matrix for QRS Detection.

From TP, FN, FP and TN values obtained by the confusion matrix, it is possible to calculate some of the most important metrics that define the validity of the model. In this case, Accuracy, Sensitivity, Specificity and F-score are calculated for validation of model's performances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 0.9771$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} = 0.9022$$

$$\text{Specificity} = \frac{TN}{TN + FP} = 0.9815$$

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 0.9127$$

Accuracy	Sensitivity	Specificity	F-score
0.9771	0.9233	0.9815	0.9127

Table 5.4. Various Metrics for QRS Detection Results.

5.3 Results for P Waves Detection

In the table below, the results of k-fold CV are shown for DNN model which detect P waves. The dataset is divided into folds, and, for each iteration, k-1 folds are used for training and one for validation. In particular, table 5.5 shows the Accuracy and Loss results of each of the folds used for Cross Validation.

P Results	Loss	Accuracy	AUC
Fold-1	0.0940	96.211%	99.522%
Fold-2	0.0848	96.563%	99.352%
Fold-3	0.0990	95.918%	99.463%
Fold-4	0.0808	96.571%	99.497%
Fold-5	0.1068	95.592%	99.552%
Fold-6	0.0955	96.044%	99.408%
Fold-7	0.1000	95.849%	86.980%
Fold-8	0.3799	87.348%	99.493%
Fold-9	0.0918	96.198%	99.538%
Fold-10	0.094	96.028%	99.175%

Table 5.5. Results of k-fold Cross Validation for P Detection.

The learning curves plot the accuracy and the loss of the model against the training epochs.

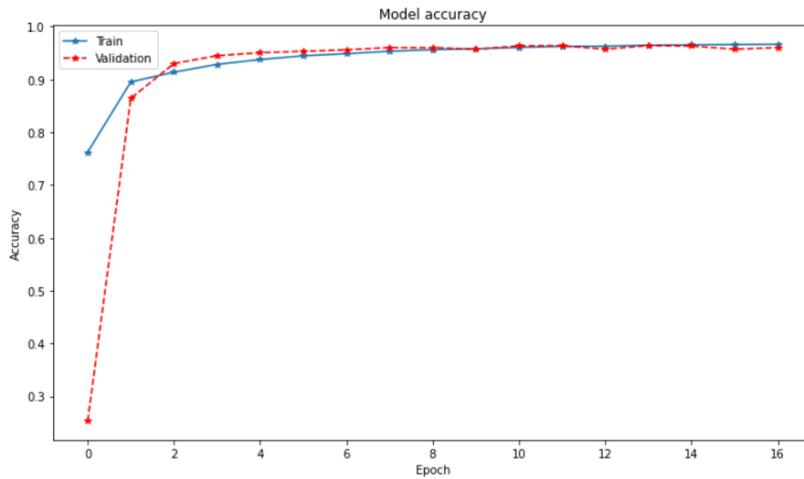


Figure 5.7. Accuracy for P Detection during each epoch.

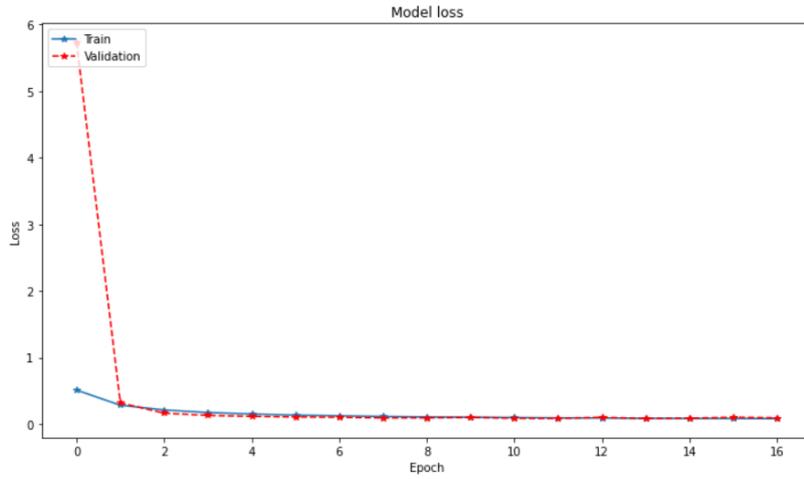


Figure 5.8. Loss for P Detection during each epoch.

Training the model for 30 epochs is enough for the model to converge. In Figure 5.7, it is possible to observe that the accuracy training and validation both increase and stabilize at a specific point around 0.96 (and around epoch 16). The same happens in Figure 5.8, where the loss training and validation both decrease until a point near to zero. This behaviour indicates an optimal fit, i.e., the model does not overfit or underfit.

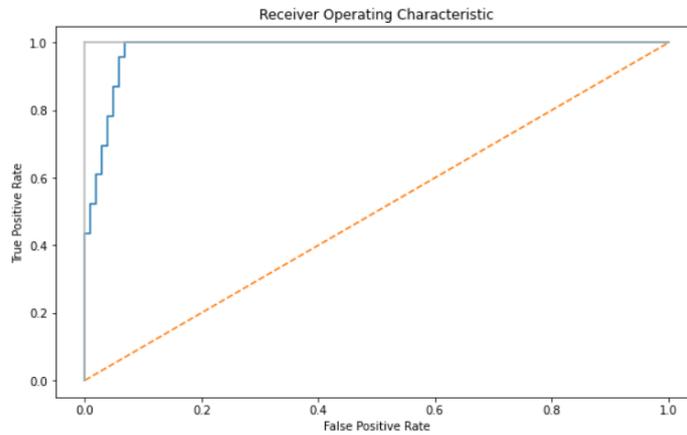


Figure 5.9. ROC for P Detection.

Figure 5.9 represents the ROC curve for this model. Results for P waves are slightly worse than those of the QRS complexes but however they reach good results in terms of performances.

	Loss	Accuracy	AUC
Average Scores for all folds	0.1227	95.232% (± 2.644)	98.198%

Table 5.6. Average Scores for all folds of CV.

The validity of the approach can be tested again on some random data. In particular, in Figure 5.10, the predicted outputs of the model are plotted for some test inputs.

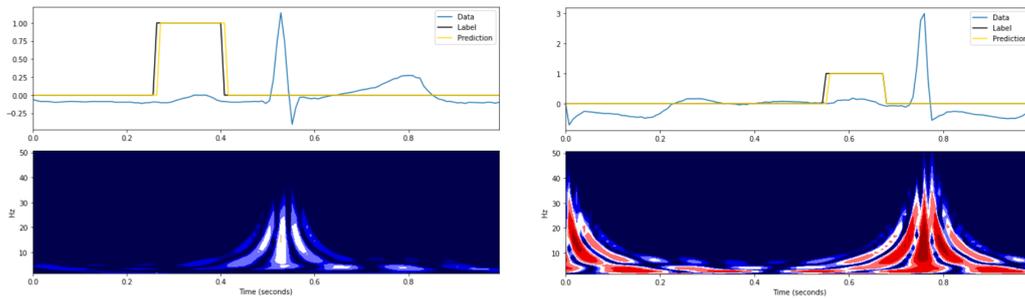


Figure 5.10. Model's predictions on new signals.

The classification performances are estimated analysing the CM, shown in 5.11.

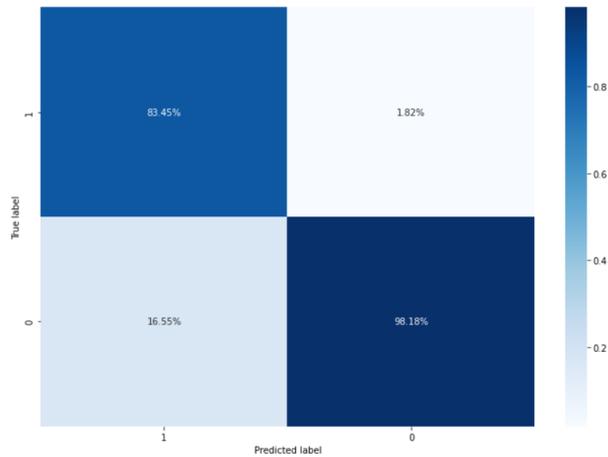


Figure 5.11. Confusion Matrix for P Detection.

From TP, FN, FP and TN values obtained by the confusion matrix, it is possible to calculate some of the most important metrics that define the model's validity.

In this case, Accuracy, Sensitivity, Specificity and F-score are calculated for validation of model's performances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 0.9619$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} = 0.8782$$

$$\text{Specificity} = \frac{TN}{TN + FP} = 0.9742$$

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 0.8558$$

Accuracy	Sensitivity	Specificity	F-score
0.9619	0.8782	0.9742	0.8558

Table 5.7. Various Metrics for P Detection Results.

5.4 Results for T Waves Detection

In the table below, the results of k-fold CV are shown for DNN model which detects T waves. The dataset is divided into folds, and, for each iteration, k-1 folds are used for training and one for validation. In particular, table 5.8 shows the Accuracy and Loss results of each of the folds used for Cross Validation.

T Results	Loss	Accuracy	AUC
Fold-1	0.1208	95.383%	99.349%
Fold-2	0.1088	95.664%	99.425%
Fold-3	0.0933	96.053%	99.283%
Fold-4	0.0986	96.041%	99.150%
Fold-5	0.1371	94.444%	99.247%
Fold-6	0.1239	95.165%	99.397%
Fold-7	0.1113	95.598%	99.354%
Fold-8	0.1452	94.125%	99.061%
Fold-9	0.1237	94.827%	99.442%
Fold-10	0.1106	95.646%	99.446%

Table 5.8. Results of k-fold Cross Validation for T Detection.

The learning curves plot the accuracy and the loss of the model against the training epochs.

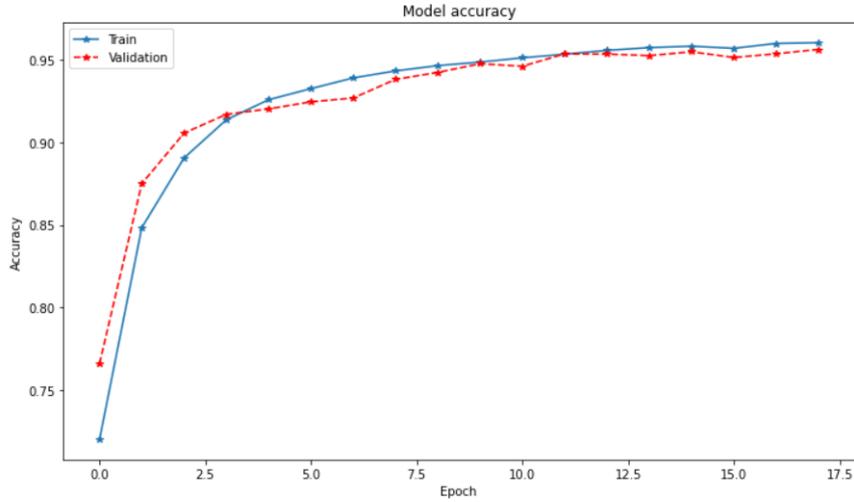


Figure 5.12. Accuracy for T Detection during each epoch.

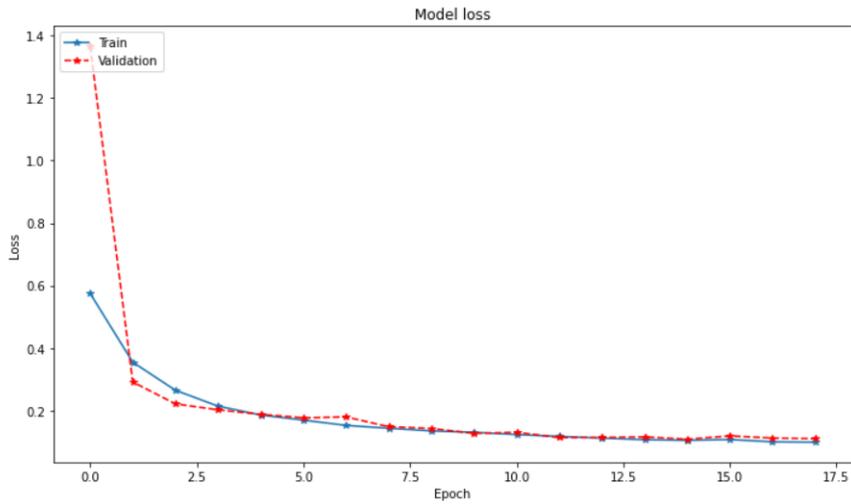


Figure 5.13. Loss for T Detection during each epoch.

	Loss	Accuracy	AUC
Average Scores for all folds	0.1173	95.294%(±0.618)	99.315%

Table 5.9. Average Scores for all folds of CV.

Training the model for 30 epochs is enough for the model to converge. In Figure 5.12, it is possible to observe that the accuracy training and validation both increase and stabilize at a specific point around 0.95 (and around epoch 17.5). The same happens in Figure 5.13, where the loss training and validation both decrease until a point near to zero. This behaviour indicates an optimal fit, i.e the model does not overfit or underfit.

Figure 5.14 represents the ROC curve for this model. Results for T waves are slightly worse than those of the other features but however they reach satisfying results.

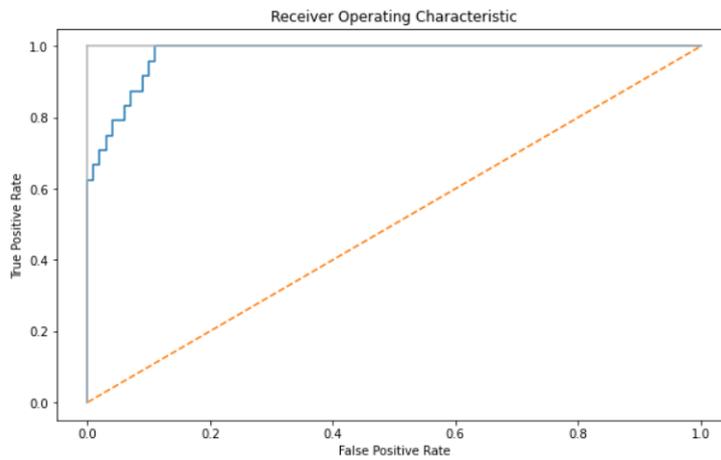


Figure 5.14. ROC for T Detection.

Also in this case, the validity of the approach can be tested on some random data. In particular, in Figure 5.15, the predicted outputs of the model are plotted for some test inputs.

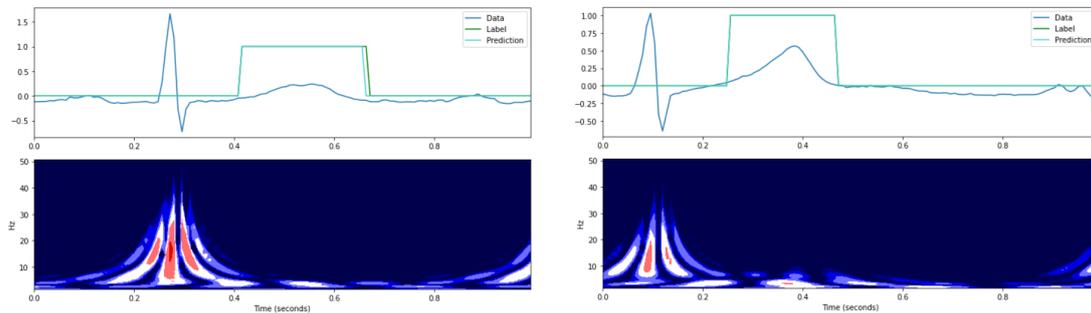


Figure 5.15. Model's predictions on new signals.

The classification performances are estimated analysing the confusion matrix, shown in 5.16. This matrix is normalized and obtained by dividing each column element by the sum of the entire column.

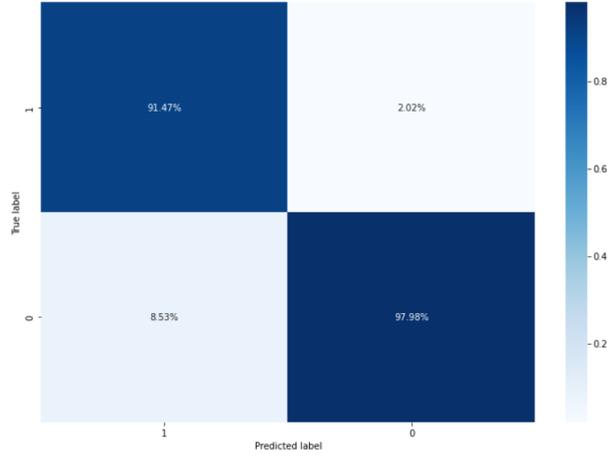


Figure 5.16. Confusion Matrix for T Detection.

From TP, FN, FP and TN values obtained by the confusion matrix, it is possible to calculate some of the most important metrics that define the validity of the model. In this case, Accuracy, Sensitivity, Specificity and F-score are calculated for validation of model’s performances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = 0.9641$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} = 0.9347$$

$$\text{Specificity} = \frac{TN}{TN + FP} = 0.9738$$

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 0.9246$$

Accuracy	Sensitivity	Specificity	F-score
0.9641	0.9347	0.9738	0.9246

Table 5.10. Various Metrics for T Detection Results.

5.5 Features Measurements

5.5.1 Features Segmentation with Developed Models

Finally, the obtained results for each of the analysed features are grouped in the table 5.11, shown below:

	Accuracy	Loss	Sensitivity	Specificity	F-score
P waves	96.19%	0.1227	87.82%	97.42%	0.8558
QRS complexes	97.71%	0.0727	92.33%	98.15%	0.9127
T waves	96.41%	0.1173	93.47%	97.38%	0.9246

Table 5.11. Obtained Results for each of the Features' Predictions.

Once the single features are detected, the next step of this project is to detect them together in a signal, putting the developed DNN's together. In particular, at the end of the work, it will be a model composed by three deep neural networks, one for each of the main features (P waves, QRS complex and T waves) in order to obtain the data windows with all the features labeled at the same time. This procedure permits to measure the features and the intervals between them.

So, first of all, the pre-trained three developed models are saved separately. After this procedure, a signal from the database is taken and pre-processed in order to obtain for it the same shape of the neural networks' input.

The signal chosen as example in this part is 'sel871' but then, all the other signals coming from the database are subjected to the same procedure in order to validate the obtained results. The chosen signal is firstly pre-processed to remove baseline wander, then a scalogram is obtained in order to get a 2d image to be fed to the networks. After the signal segmentation to obtain windows, the signal is fed to the pre-trained three models saved before. The outputs coming from the neural networks are shown in Figure 5.17.

Every predicted output of each of the networks is a sequence of 1 and 0 depending on the probability to have or not a predetermined feature in each sample. For example, with reference to the Figure 5.17, the yellow label represents the network's prediction of the probability in each sample that the signal represents a P wave. The same happens for QRS complexes and T waves, with red and light blue labels, respectively.

So, in detail, the yellow signal's starting and ending points (sequences of 1) represent the prediction of the network for the beginning and the end of P wave, the starting and ending of the red one are the beginning and the end of QRS complex, while the starting and ending of the light blue sequence of 1 represent the beginning and the end of T wave.

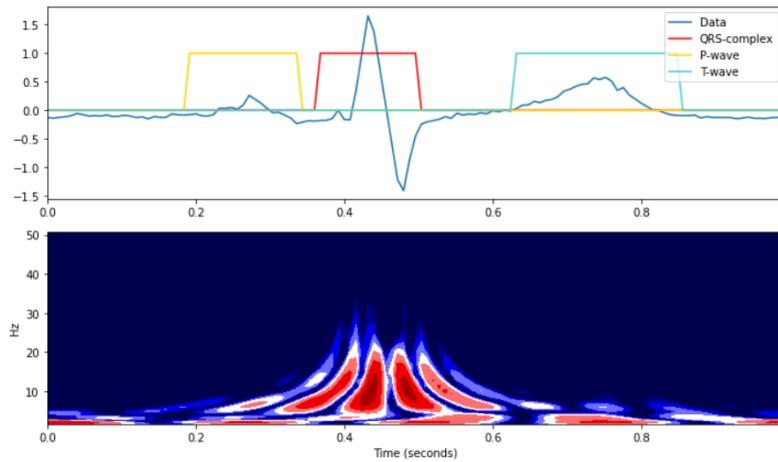


Figure 5.17. Predictions of the three networks on a random signal from Database.

The same procedure is used for signals completely external to the trained database in order to verify that the DNN model is able to predict outputs correctly also on signal that it has never seen before.

The following signal, with its 12-leads, is pre-processed and given as input to the DNN model, obtaining for each window a result similar to that one shown below.

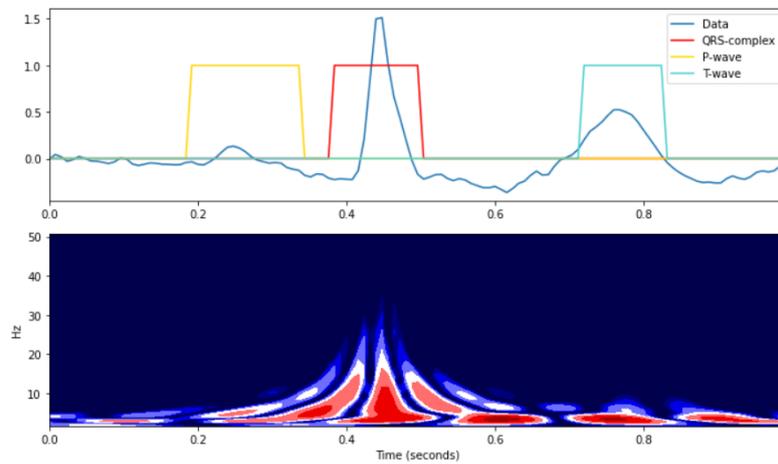


Figure 5.18. Predictions of the three networks on a new signal never seen before.

5.5.2 Widths and Intervals Measurements

Since CVDs affect the electrophysiological activity of the heart, ECG features are quantitatively and extensively analysed to assess their potential incremental contribution to an event's risk.

Once the features have been obtained, the last step is to calculate their widths and the intervals between them, since they are a great starting point for developing of automated cardiac diagnostic methods and facilitating large-scale screening. The main features calculated are PR intervals, QRS complex widths and QT intervals. These features are analysed in detail in the following subsections.

PR Interval

PR interval in ECG, as seen before, is the period between the starting of P wave, which is the atrial depolarization's onset, and the starting of QRS complex, which is the ventricular depolarization's onset (measured in msec). Its normal duration extends from 120 and 200 ms (around 3-5 small horizontal boxes) and it is also called PQ interval (shown in Figure 5.19).

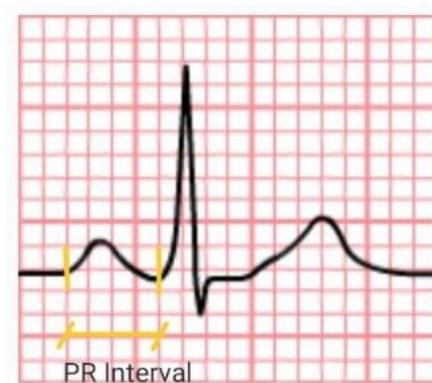


Figure 5.19. Normal PR Interval on ECG grid.

PR interval's variations could be associated with some medical conditions:

- Long PR Interval: over 200 ms. It is a sign of atrioventricular block (AVB). It is often due to a slow conduction of the atrioventricular node and it can indicate a slow conduction between atria and ventricle. This condition is also known as first degree heart block and it can be associated with AV node's fibrosis, medicament which slow the AV node (beta-blockers), high vagal tone, carditis (associated to Lyme disease), acute rheumatic fever or hypokalaemia [54].

- Short PR Interval: less than 120 ms. It is often due to pre-excitation syndromes (Wolf-Parkinson-White or Lown-Ganong-Levine) and junctional arrhythmia (atrioventricular reentrant tachycardia or junctional rhythm).

Other types of heart blocks can be assumed by a variable PR.

So, given the Figure 5.17, PR interval has been calculated through the following algorithm:

```

#For PR Interval
savep = []
saveqrs1 = []
#To save all the indexes of the rising edges of P and QRS in a vector
for g in range(start, len(predictions_temp_p)-1):
    if (predictions_temp_p[g] == 0):
        if (predictions_temp_p[g+1] == 1):
            savep.append(g+1)
for c in range(start, len(predictions_temp)-1):
    if (predictions_temp[c] == 0):
        if (predictions_temp[c+1] == 1):
            saveqrs1.append(c+1)
#To calculate PR Interval
for q in range(0, len(saveqrs1)-1):
    if(len(saveqrs1)>len(savep)):
        if(savep != []):
            if(savep[q]<saveqrs1[q+1]):
                PR = (saveqrs1[q+1]-savep[q])/125
                if((PR-0.04) > 0):
                    print('PR Interval:')
                    print(f'> {PR}')
for b in range(0, len(savep)):
    if(len(saveqrs1) == len(savep)):
        if(savep[b]<saveqrs1[b]):
            PR = (saveqrs1[b]-savep[b])/125
            if((PR-0.04) > 0):
                print('PR Interval:')
                print(f'> {PR}')

```

Figure 5.20. Algorithm for Calculation of PR Interval in Each Window.

, where `predictions_tmp_p` indicates the vector of predictions (the yellow one) for P waves, `predictions_tmp` indicates the vector of predictions (the red one) for QRS complexes, and the other variables are set to make possible the calculations. Finally, the obtained result for this interval calculation is shown below:

$$PR_Interval = 136ms$$

, which is in the normal range of supposed values.

QRS Widths

QRS complex on typical ECG is the combination of the three central and most visually obvious deflections (Q, R and S) of the track. It is the main feature and corresponds to the depolarization of the two heart's ventricles and to the large ventricular muscles' contraction.

The normal QRS width, in adults, ranges from 80 to 100 ms, while it may be shorter in children. The three deflections are in rapid succession and they reflect a single event (for this reason they are considered together). These waves don't appear all in all leads, but their sequence is:

- Q wave: which is a downward deflection that immediately follows the P wave.
- R wave: which is an upward deflection that follows the Q wave.
- S wave: which is a downward deflection that immediately follows the R wave.

For measuring purposes, QRS interval starts at the beginning of the Q wave (or at the end of the PR interval) and finishes at the end of the S wave. It is shown in Figure 5.21.

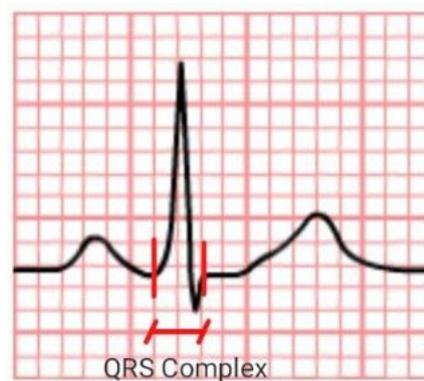


Figure 5.21. Normal QRS Complex on ECG grid.

A normal QRS complex has an amplitude in range of 5 to 30 mm and a duration between 0.06 and 0.12 sec (around 1.5-3 small horizontal boxes). QRS interval's variations in width, amplitude and shape could be associated with some medical conditions and they are useful for diagnosis of cardiac arrhythmias, ventricular hypertrophy, myocardial infarction, electrolyte derangements and other diseases.

An abnormal QRS complex can vary its shape from nearly normal to slurred and notched to wide and bizarre. QRS complexes lower in voltage or abnormally small can be seen in obese or hyperthyroid patients and in pleural effusion, while

high QRS complexes are usually seen in hypertrophy of the ventricles or in an abnormal pacemaker or in aberrantly conducted beats.

The various shapes that a QRS complex may assume are showed in the figure below.

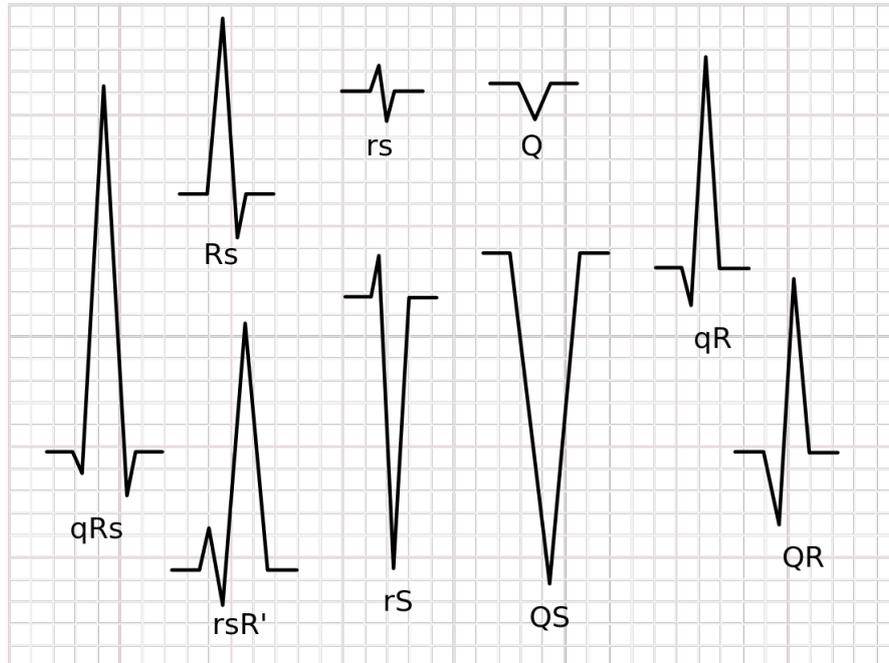


Figure 5.22. Different QRS Shapes for Various Anomalies.[21]

QRS width, under depolarization hypothesis, plays an important role because it quantifies the duration of the depolarization of the ventricles, and it can identify anomalies due to delayed conduction and dysfunction of sodium current.

- Long QRS complex: over than 120 ms. It can indicate intraventricular conduction delays (such as bundle branch block) or hyperkalaemia.
- Narrow QRS complex (normal): less than 120 ms. It indicates that the ventricles are depolarized normally.

So, given the Figure 5.17, QRS complex width has been calculated through the following algorithm:

```

#For QRS Width
#To remove the first cropped QRS Label
if(predictions_temp[0] == 1):
    start = 1
    for p in range(len(predictions_temp)-1):
        if(predictions_temp[p] == 1):
            if flag == 0:
                start = start + 1
                if(predictions_temp[p+1] == 0):
                    flag = 1
#To reset QRS counter whenever one interval finishes
for j in range(start, len(predictions_temp)-1):
    if (predictions_temp[j] == 1):
        count = count + 1
        flag1 = 1
    if (predictions_temp[j+1] == 0):
        if (flag1 == 1):
            save.append(count)
            k = k+1
            flag1 = 0
            count = 0
#To calculate QRS Width
for l in range(0, k):
    avg = save[l]/125
    if((avg-0.04) > 0):
        print('QRS Width:')
        print(f'> {avg}')

```

Figure 5.23. Algorithm for Calculation of QRS Width in Each Window.

, where `predictions_tmp` indicates the vector of predictions (the red one) for QRS complexes and the other variables are set to make possible the calculations. Finally, the obtained result for this interval calculation is shown below:

$$QRS_Width = 96ms$$

, which is in the normal range of supposed values.

QT Interval

QT interval in ECG is usually used to assess the heart's electrical properties. It measures the time starting from the beginning of Q wave until the end of T wave and it indicates approximately the time from the starting of ventricles' contraction to their final relaxation.

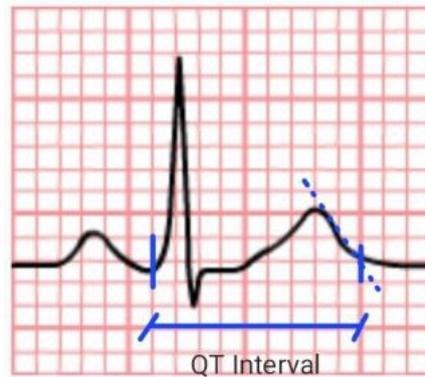


Figure 5.24. Normal QT Interval on ECG grid.

An increased risk of developing sudden cardiac deaths and abnormal heart rhythms can be associated to an abnormally long or short QT interval. Normal values range of QT interval is from 350 to 450 ms for males and from 360 to 460 ms for females.

So, with reference to the Figure 5.17, QT interval has been calculated through the following algorithm.

```

#For QT Interval
savet = 0
saveqrs = 0
flag2 = 0
for m in range(start, len(predictions_temp)):
    if (predictions_temp[m] == 1):
        if (flag2 == 0):
            saveqrs = m
            flag2 = 1
for m in range(start, len(predictions_temp_t)-1):
    if (predictions_temp_t[m] == 1):
        savet = m
#To calculate QT Interval
if (saveqrs < savet):
    if (saveqrs != 0):
        if(predictions_temp_t[m] != 1):
            QT = ((savet-saveqrs)/125)
            print('QT Interval:')
            print(f'> {QT}')

```

Figure 5.25. Algorithm for Calculation of QT Interval in Each Window.

, where `predictions_tmp` indicates the vector of predictions (the red one) for QRS complexes, `predictions_tmp_t` indicates the vector of predictions (the light blue one) for T wave and the other variables are set to make possible the calculations. Finally, the obtained result for this interval calculation is shown below:

$$QT_Interval = 440ms$$

, which is in the normal range of supposed values.

Anomalies of QT interval are generally due to genetic conditions (such as long QT syndrome), to certain medicament (such as sotalol or pitolisant), to concentration disturbances of some salts in the blood (such as hypokalaemia) or to hormonal derangements (such as hypothyroidism).

After these specifications, the final output results of the three networks are the following for each of the windows created during data segmentation.

```
QRS width:
> 0.096
QT Interval:
> 0.44
PR Interval:
> 0.13599999999999998
```

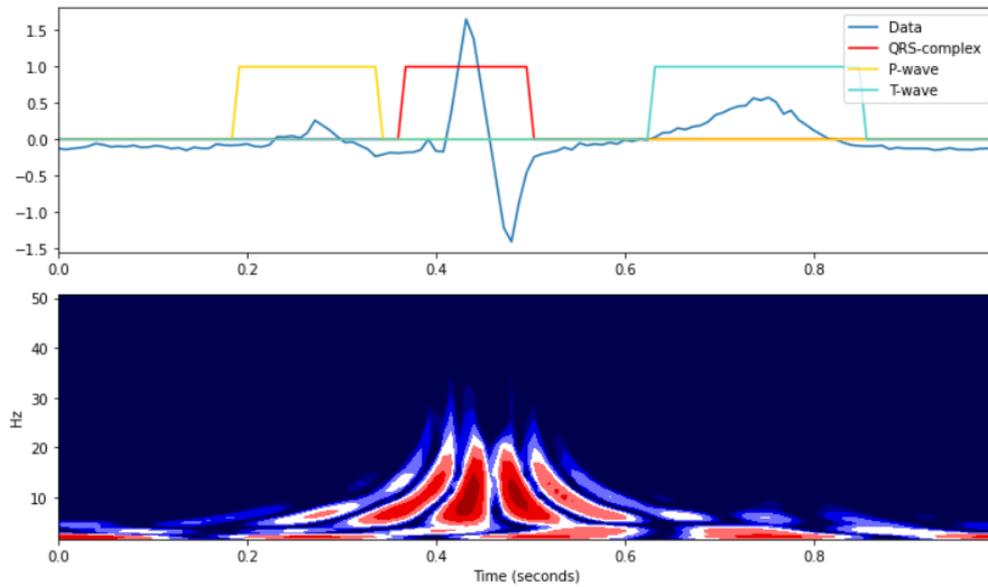


Figure 5.26. Results of Intervals Measurements on a Prediction.

Chapter 6

Validation and Discussion

In this chapter, a results' validation is performed in order to demonstrate the correct working of the DNN algorithm also on signals that have never be seen by the neural networks. Then, a discussion part follows enumerating the various DNN models used nowadays and their performances with respect to the proposed method.

6.1 Results' Validation

The results' validation is performed on fifteen signals coming from the simulator "Fluke ProSim 4 Vital Signs Simulator". This simulator is able to generate an ECG signal and export it, with its 12-leads, on an XML file, permitting to use it for other analysis purposes.

In particular, the signals coming from the simulator are firstly pre-processed as the other signals used in the NN model, getting their scalogram. Then, they are segmented into windows with length of 125 samples and they are fed to the neural networks in order to extract the main features seen before (P and T waves and QRS complexes).

An important detail to note is that, in the evaluation of a learning algorithm, the dataset must all have the same characteristics, such as the same normalization and the same sampling rate (it has to be re-sampled).

Finally, the intervals' measurements are performed obtaining the following results. The table 6.1, in particular, shows the actual values of the intervals calculated by the simulator, with respect to the predicted values coming from the DNNs' output.

Tracks	PR Interval (ms)		QRS Width (ms)		QT Interval (ms)	
	Actual	Predicted	Actual	Predicted	Actual	Predicted
File_1	168	168	86	88	390	384
File_2	148	152	86	79	348	336
File_3	160	128	86	88	362	328
File_4	164	184	88	96	368	360
File_5	166	160	88	88	366	360
File_6	160	152	86	88	412	408
File_7	160	136	84	88	425	408
File_8	158	152	84	88	438	432
File_9	152	152	70	78	462	440
File_10	/	/	100	96	443	456
File_11	162	176	90	96	370	368
File_12	154	160	86	88	370	368
File_13	/	/	100	96	522	488
File_14	152	136	70	80	372	368
File_15	/	/	80	80	379	376

Table 6.1: Results' Validation.

As it is possible to see from the table, the predicted results for PR intervals and QRS complexes are very close to the actual values. In other hand, the QT values are slightly underestimated since the T end is a difficult parameter to be recognized by the network.

For a better evaluation of the table's results, the values of the average and standard deviation are calculated for each of the features, as follows:

$$\mu(x) = \frac{1}{N} \sum_{i=1}^N x_i \tag{6.1}$$

$$\sigma(x) = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2} \tag{6.2}$$

The fiducial intervals' variations are analysed by calculating mean and standard deviation of each interval. Their results for actual and predicted values are shown in the table below.

Mean and Standard Deviation		
	Actual	Predicted
PR Interval	158.67 ms \pm 6.11 ms	154.67 ms \pm 16.48 ms
QRS Width	85.60 ms \pm 8.32 ms	87.80 ms \pm 6.34 ms
QT Interval	401.80 ms \pm 47.95 ms	392.00 ms \pm 45.56 ms

Table 6.2: Mean and Standard Deviation for Each of the Analysed Feature.

So, the mean errors for PR Interval, QRS Width and QT Interval measurements are respectively equal to 4 ms, 2.2 ms and 9.8 ms. The standard deviation (SD) measures the variation or dispersion amount of a set of values. A high standard deviation means that the values cover a wider range. In contrast, a low standard deviation indicates that the values are closer to the mean of the set. The SD of predicted values is useful to understand the values' dispersion in different models.

Then, also the maximum and minimum deviations are calculated for each of the intervals, as the absolute value of the difference between predicted and actual values. So, for each of the fifteen simulated case, three deviations are obtained, one for every interval. At this point, the minimum and maximum between the deviations are taken and shown in following table:

Deviation	Minimum	Maximum
PR Interval	0 ms	32 ms
QRS Width	0 ms	10 ms
QT Interval	2 ms	34 ms

Table 6.3: Minimum and Maximum Deviations for Each of the Analysed Feature.

6.2 Discussion

In this thesis project, a deep neural network model is proposed for detection of ECG signal's most important features. The proposed deep learning model performs significantly better than other deep learning ECG segmentation models, especially in segmentation of P waves and QRS complexes. Accurate automated ECG segmentation would help to develop automated cardiac diagnostic methods and facilitate large-scale screening.

ECG have a lot of variations, and this is the main challenge in ECG features segmentation. The variations of ECG can be present in shape, duration, frequency and amplitude and they can derive from the differences between patients, placement of ECG leads or monitoring equipment. For this reason, an automatic approach is useful to handle all the possible variations of ECG waves in an efficient way.

The traditional automatic models usually identify the R wave peak and then, through rule-based approaches [55] or Hidden Markov Models (HMM) [56], identify the remaining waves' positions. But these approaches are often not generalizable and not robust enough to manage all the variations of ECG signals, so they have very low average performances. The proposed method has instead the following main contributions:

- The 2D-CNN-BiLSTM approach labels each data point as P-wave, QRS complex or T-wave, creating a neural network model able to segmenting ECG waves.
- The CNN and BiLSTM combination is able to remove the noise, creating additional features and achieves a better accuracy with respect to other methods of deep learning or rule-based.

Automatic segmentation of ECG waves can be obtained through multiple methods such as Fourier transforms, wavelet transformations or pattern recognition. The focus of these methods is to detect just some parts of the signal, often the QRS complexes. The most used algorithm employed to find these complexes is the Pan-Tompkins [57], a derivative based algorithm. The Discrete Wavelet Transform (DWT) approach [58] is used to improve the extraction of information from P waves, QRS complexes and T waves. Also, adaptive threshold algorithms can be used to identify the important parts of the ECG signal. Finally, another method to identify the various part of the ECG is to train a Hidden Markov model with labeled data.

Deep Learning approaches are widely used in signal processing for biomedical applications. In particular, the LSTM architecture is used for ECG waves segmentation [59]. Using ECG raw signals, in addition, other features can be extracted through different filtering kernels also including the ECG first and second derivatives. The obtained results in this thesis project are better than those of other existing methods and prove that deep learning techniques are a good method to automatically extract most important features from ECG wave.

The comparison of performances of 2D-CNN-BiLSTM with respect to the other existing methods are shown in table 6.4 and in Figure 6.1.

Methods	P(%)	QRS(%)	T(%)
2D-CNN-BiLSTM	96.19%	97.07%	96.49%
ECG-SegNet (Bidirectional LSTM)	92.00%	94.00%	92.00%
HMM on raw ECG data	5.50%	79.00%	56.03%
HMM on wavelet encoded ECG	74.20%	94.40%	88.23%
LSTM	95.00%	98.00%	97.00%

Table 6.4. Results for each of the Features' Predictions with Different Methods.

Given the task of automatic detect the most relevant ECG features, the best results are generally obtained with HMM on wavelet encoded ECG data but, as it is possible to see from the bar graph, the proposed method is able to reach better performances.

The 2D-CNN-BiLSTM approach can produce relevant improvements, in particular in P waves and QRS waves segmentation. Moreover, this model overcomes also the performances of existing Bidirectional LSTM architectures, such as ECG-SegNet, and of other traditional methods.

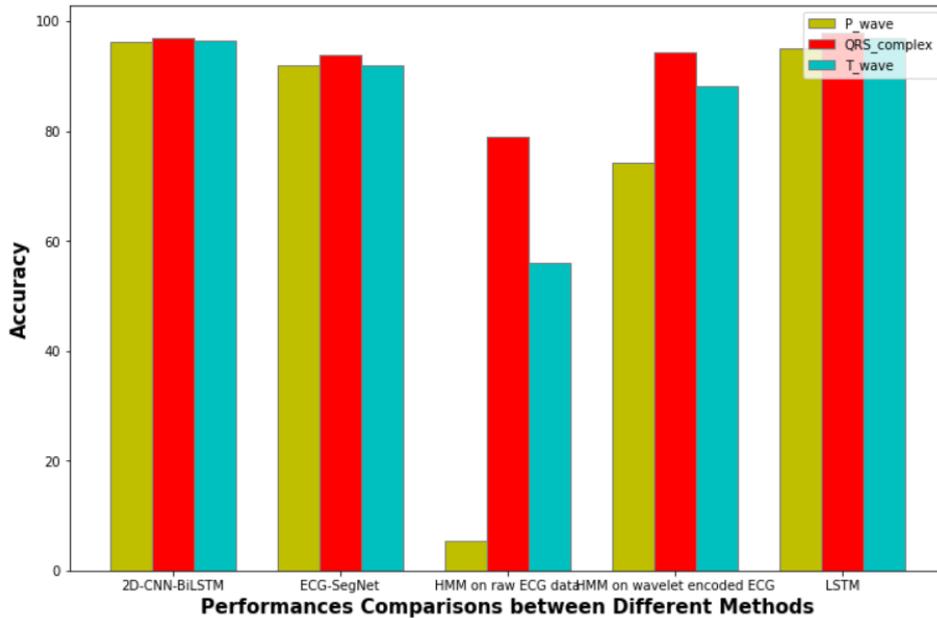


Figure 6.1. Performances Comparisons between Different Methods.

The disadvantage of this algorithm is that it has an higher computational processing cost.

6.3 Conclusions

The computerized ECG systems increased the collection of more ECG data, and these produced data require to be analysed by cardiologists and require a huge amount of time. ECG signal processing tools already exist and are available for supporting these processes, but they require an engineering human-assisted to perform the feature extraction.

The systems based on ECG classification can work simultaneously with systems that perform prediction on cardiovascular diseases, which are based on data mining. These systems, in case of emergency, can automatically alert public safety points or hospital, making faster the aid and increasing the prevention of diseases. The implementation of mobile monitoring system can affect positively the benefits of patients which require access to continuous care.

This thesis project proposes a 2D-CNN-BiLSTM model for automatic ECG features segmentation and extraction, and it performs significantly better than other existing methods based on signal processing or HMM. The LSTM architecture can understand the temporal relations and it is very able to perform the time series segmentation task. Specifically, the Bidirectional LSTM is able to capture the long-term temporal dependencies [60] and to solve the vanishing or exploding gradient problem (where the error derivative after a very short period tends to zero or infinity). The CNN works as a feature extractor, while the LSTM considers only the temporal information of the ECG waves in the input. The 2D-CNN-BiLSTM network uses both spatial and temporal information to segment the ECG wave.

Summarizing, the proposed method involves mainly the use of two Convolutional and two LSTM Neural Networks. After the pre-processing operations, removing the baseline through the local linear regression method, the scalograms of the signals are derived. Through the use of Continuous Wavelet Transform, in fact, it was possible to transform signals from time domain to frequency domain, turning 1D signals into 2D coloured images, in order to make them an ideal input to feed the CNN. Scalograms are very powerful method and represent a very robust approach since they exhibit optimal performances for the classification of morphological images. For the extraction of ECG features, 2D scalogram images are then segmented into windows in order to obtain images of size 125x62x1. Around 15000 scalogram images are obtained to train the CNN-BiLSTM model with a k-fold Cross Validation. After enhancing the model (using layers such as batch normalisation, time-distributed layer, and fully connected layer), a confusion matrix, and other performance metrics are used to assess the classifier's performances.

The performed experiments show the usefulness and ability of this model to outperform other approaches in terms of classification performances. The confusion matrix for P, QRS and T detection in the testing dataset shows respectively 96.19%, 97.71% and 96.41% of validation accuracy. Furthermore, sensitivity and specificity for P wave are 87.82%, 97.42%, while sensitivity and specificity for QRS Interval are 92.33% and 98.15%, and for T wave, sensitivity and specificity are 93.47% and 97.38% respectively.

The obtained results are better than other techniques and they greatly reduce the interventions' amount required by doctors. The model's performances are also better than other deep learning segmentation methods, especially in P-waves and QRS complexes. This kind of automated algorithm can help to automate diagnosis of cardiac diseases and to simplify large-scale screening tasks. Likewise, the suggested model can assist clinicians in correctly identifying ECG features during routine ECG monitoring.

Acronyms

SVC Superior Vena Cava

IVC Inferior Vena Cava

SA Sinoatrial Node

AV Atrioventricular Node

ECG Electrocardiogram

CVD Cardiovascular Disease

CAD Coronary Artery Disease

CHF Congestive Heart Failure

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

NN Neural Network

ANN Artificial Neural Network

DNN Deep Neural Network

MLP Multi-Layer Perceptron

CNN Convolutional Neural Network

RNN Recurrent Neural Network

LSTM Long Short Term Memory

BiLSTM Bidirectional Long Short Term Memory

ReLU Rectified Linear Unit
GD Gradient Descent
SGD Stochastic Gradient Descent
CM Confusion Matrix
TP True Positive
FN False Negative
FP False Positive
TN True Negative
TPR True Positive Rate
TNR True Negative Rate
FPR False Positive Rate
FNR False Negative Rate
PPV Positive Predictive Value
NPV Negative Predictive Value
ROC Receiver Operating Characteristic
AUC Area Under Curve
FDA Food and Drug Administration
SVM Support Vector Machine
DT Decision Tree
RF Random Forest
KNN k-Nearest Neighbour
BP Blood Pressure
AF Atrial Fibrillation
ICU Intensive Care Unit
GPU Graphics Processing Unit

FIR Finite Impulse Response

CWT Continuous Wavelet Transform

DWT Discrete Wavelet Transform

PVC Premature Ventricular Contraction

ADAM Adaptive Moment Estimation

ADAGRAD Adaptive Gradient Algorithm

RMSProp Root Mean Square Propagation

CV Cross Validation

HMM Hidden Markov Model

SD Standard Deviation

Bibliography

- [1] Jason Shar. *Unraveling the Etiologies of Discrete Subaortic Stenosis: A Focus on Left Ventricular Outflow Tract Hemodynamics*. PhD thesis, Kennesaw State University, 04 2021.
- [2] Cardiology Associates of Michigan, 10 2019. URL <https://www.cardofmich.com/anatomy-human-heart-fun-facts/>.
- [3] Majid Mafi, Sreeraman Rajan, Miodrag Bolic, Voicu Groza, and Hilmi Dajani. Blood pressure estimation using oscillometric pulse morphology. *Conference proceedings : ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, 2011:2492–6, 08 2011. doi: 10.1109/IEMBS.2011.6090691.
- [4] Wikimedia Commons. File:conductionsystemoftheheart.png — wikimedia commons, the free media repository, 2021. URL <https://commons.wikimedia.org/w/index.php?title=File:Conductionsystemoftheheart.png&oldid=563625748>.
- [5] Amboss. Conduction system of the heart, 06 2022. URL https://www.amboss.com/us/knowledge/Cardiac_physiology.
- [6] Cables and sensors. 12-lead ecg placement guide, 2022. URL <https://www.cablesandsensors.com/pages/12-lead-ecg-placement-guide-with-illustrations>.
- [7] Wikimedia Commons. File:ecg complex.svg — wikimedia commons, the free media repository, 2021. URL https://commons.wikimedia.org/w/index.php?title=File:ECG_Complex.svg&oldid=569365896.
- [8] Biology Dictionary. Pr interval, 11 2020. URL <https://biologydictionary.net/pr-interval/>.
- [9] Oleg Starostenko, Vicente Alarcon-Aquino, J. Rodriguez-Asomoza, Oleg Sergiyenko, and Vera Tyrsa. *Remote Health/Vital Sign Monitoring*, pages

- 221–243. Springer, 01 2015. ISBN 978-3-319-12816-0. doi: 10.1007/978-3-319-12817-7_10.
- [10] Arden Dertat. Applied deep learning - part 1: Artificial neural networks, 08 2017.
- [11] Fatimah Hussain. Your handbook to convolutional neural networks, 12 2020. URL <https://medium.com/analytics-vidhya/your-handbook-to-convolutional-neural-networks-628782b68f7e>.
- [12] Aliaa Rassem and El-Beltagy et al. Cross-country skiing gears classification using deep learning. *ArXiv*, 06 2017.
- [13] Kazi Nazmul Haque, Mohammad Abu Yousuf, and Rajib Kumar Rana. Image denoising and restoration with cnn-lstm encoder decoder with direct attention. *ArXiv*, abs/1801.05141, 2018.
- [14] Jerry An. How to remember all these classification concepts forever, 06 2020.
- [15] Antony Paulson Chazhoor. Roc curve in machine learning, 07 2019. URL <https://towardsdatascience.com/roc-curve-in-machine-learning-fea29b14d133>.
- [16] Saurabh Saxena. Precision-recall curve, 10 2022. URL <https://towardsai.net/p/1/precision-recall-curve>.
- [17] Sulaiman Somani and Russak et al. Deep learning and the electrocardiogram: review of the current state-of-the-art. *EP Europace*, 23(8):1179–1191, 02 2021. ISSN 1099-5129. doi: 10.1093/europace/euaa377. URL <https://doi.org/10.1093/europace/euaa377>.
- [18] Wikimedia Commons. File:wavelet - morlet.png — wikimedia commons, the free media repository, 2020. URL https://commons.wikimedia.org/w/index.php?title=File:Wavelet_-_Morlet.png&oldid=451151706.
- [19] Jasmeet Singh. Implementing sgd from scratch, 07 2020. URL <https://towardsdatascience.com/implementing-sgd-from-scratch-d425db18a72c>.
- [20] Johar Ashfaq Aatqb and Amer Iqbal. Introduction to support vector machines and kernel methods. *ResearchGate*, 04 2019.
- [21] Wikimedia Commons. File:qrs nomenclature.svg — wikimedia commons, the free media repository, 2020. URL https://commons.wikimedia.org/w/index.php?title=File:QRS_nomenclature.svg&oldid=477144226.

- [22] Jussi Koivumäki, Topi Korhonen, and Pasi Tavi. Impact of sarcoplasmic reticulum calcium release on calcium dynamics and action potential morphology in human atrial myocytes: A computational study. *PLoS computational biology*, 7:e1001067, 01 2011. doi: 10.1371/journal.pcbi.1001067.
- [23] Leonard S. Lilly. *Braunwald's Heart Disease: A Textbook of Cardiovascular Medicine (5th ed.)*. Elsevier, 1997. ISBN 0-7216-5666-87. URL https://books.google.it/books?hl=it&lr=&id=blq42K8AY5AC&oi=fnd&pg=PP2&dq=E.+Braunwald,+5a+ed.,+Philadelphia,+W.B.+Saunders+Co&ots=_g6t3akJaq&sig=1-LwXov_Q8Wmilwn4dxd0PUIdSc&redir_esc=y#v=onepage&q&f=false.
- [24] Malcolm S. Thaler. *The Only EKG Book You'll Ever Need*. Lippincott Williams et al., 2007. ISBN 0-7817-4176-9.
- [25] J. Gordon et al. Betts. *Anatomy and Physiology*. OpenStax, 04 2013. URL <https://openstax.org/books/anatomy-and-physiology/pages/1-introduction>.
- [26] Gan-Xin Yan, Ramarao Lankipalli, James Burke, Simone Musco, and Peter Kowey. Ventricular repolarization components on the electrocardiogram - cellular basis and clinical significance. *Journal of the American College of Cardiology*, 42:401–9, 09 2003. doi: 10.1016/S0735-1097(03)00713-7.
- [27] Jordan M Prutkin. Ecg tutorial: Basic principles of ecg analysis, 04 2022. URL <https://www.uptodate.com/contents/ecg-tutorial-basic-principles-of-ecg-analysis>.
- [28] World Health Organization. Cardiovascular diseases (cvds), 2021. URL [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)).
- [29] Matthew Kelly and Christopher Semsarian. Multiple mutations in genetic cardiovascular disease. *Circulation: Cardiovascular Genetics*, 2(2): 182–190, 2009. doi: 10.1161/CIRCGENETICS.108.836478. URL <https://www.ahajournals.org/doi/abs/10.1161/CIRCGENETICS.108.836478>.
- [30] Jacques E Rossouw. Hormones, genetic factors, and gender differences in cardiovascular disease. *Cardiovascular Research*, 53(3):550–557, 02 2002. ISSN 0008-6363. doi: 10.1016/S0008-6363(01)00478-3. URL [https://doi.org/10.1016/S0008-6363\(01\)00478-3](https://doi.org/10.1016/S0008-6363(01)00478-3).
- [31] Michael E. Mendelsohn and Richard H. Karas. The protective effects of estrogen on the cardiovascular system. *New England Journal of Medicine*, 340

- (23):1801–1811, 1999. doi: 10.1056/NEJM199906103402306. URL <https://doi.org/10.1056/NEJM199906103402306>. PMID: 10362825.
- [32] Sushman Biswas. Advantages of deep learning, plus use cases and examples, 11 2021. URL <https://www.width.ai/post/advantages-of-deep-learning>.
- [33] Avijeet Biswal. Top 10 deep learning algorithms you should know in 2023, 10 2022. URL <https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm>.
- [34] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, 2019. ISBN 9781492032618. URL <https://books.google.it/books?id=HHetDwAAQBAJ>.
- [35] Dan Cireşan, Ueli Meier, and Juergen Schmidhuber. Multi-column deep neural networks for image classification. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 02 2012. doi: 10.1109/CVPR.2012.6248110.
- [36] Felix Gers, Nicol Schraudolph, and Jürgen Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 3:115–143, 01 2002. doi: 10.1162/153244303768966139. Bobra, Neraj, lstm-grs-detector, 2020.
- [37] Valentina Alto. Neural networks: parameters, hyperparameters and optimization strategies, 07 2019.
- [38] Chayakrit Krittanawong, HongJu Zhang, Zhen Wang, Mehmet Aydar, and Takeshi Kitai. Artificial intelligence in precision cardiovascular medicine. *Journal of the American College of Cardiology*, 69:2657 – 6 4, 05 2017. doi: 10.1016/j.jacc.2017.03.571.
- [39] Gary Friesen, Thomas Jannett, Manal Jadallah, Stanford Yates, Stephen Quint, and Hubert Nagle, Jr. A comparison of the noise sensitivity of nine qrs detection algorithms. *IEEE transactions on bio-medical engineering*, 37: 85–98, 02 1990. doi: 10.1109/10.43620.
- [40] Jinkwon Kim and Hangsik Shin. Simple and robust realtime qrs detection algorithm based on spatiotemporal characteristic of the qrs complex. *PloS one*, 11:e0150144, 03 2016. doi: 10.1371/journal.pone.0150144.
- [41] Awni Y. et al. Hannun. Cardiologist-level arrhythmia detection with convolutional neural networks, 2017. URL <https://arxiv.org/abs/1707.01836>.

- [42] Antônio Luiz et al. P. Ribeiro. Tele-electrocardiography and bigdata: The code (clinical outcomes in digital electrocardiography) study. *Journal of electrocardiology*, 2019.
- [43] Jilong et al. Wang. Towards interpretable arrhythmia classification with human-machine collaborative knowledge representation. *IEEE Transactions on Biomedical Engineering*, 68(7):2098–2109, 2021. doi: 10.1109/TBME.2020.3024970.
- [44] Peng Lu and Gao et al. Kecnet: A light neural network for arrhythmia classification based on knowledge reinforcement. *Journal of Healthcare Engineering*, 2021:1–10, 04 2021. doi: 10.1155/2021/6684954.
- [45] Junsheng Yu, Xiangqing Wang, Xiaodong Chen, and Jinglin Guo. Automatic premature ventricular contraction detection using deep metric learning and knn. *Biosensors*, 11(3), 2021. ISSN 2079-6374. doi: 10.3390/bios11030069. URL <https://www.mdpi.com/2079-6374/11/3/69>.
- [46] Ertan Butun, Özal Yildirim, Muhammed Talo, Ru San Tan, and U Rajendra Acharya. 1d-cadcapsnet: One dimensional deep capsule networks for coronary artery disease detection using ecg signals. *Physica Medica*, 70, 01 2020. doi: 10.1016/j.ejmp.2020.01.007.
- [47] Jayaraman J. Thiagarajan, Deepta Rajan, Sameeksha Katoch, and Andreas Spanias. Ddxnet: a deep learning model for automatic interpretation of electronic health records, electrocardiograms and electroencephalograms. *Scientific Reports*, 10, 10 2020. doi: 10.1038/s41598-020-73126-9.
- [48] Tejas Radhakrishnan, Jay Karhade, S.K. Ghosh, P.R. Muduli, R.K. Tripathy, and U. Rajendra Acharya. Afcnnnet: Automated detection of af using chirplet transform and deep convolutional bidirectional long short term memory network with ecg signals. *Computers in Biology and Medicine*, 137: 104783, 2021. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.compbimed.2021.104783>. URL <https://www.sciencedirect.com/science/article/pii/S0010482521005771>.
- [49] Georgios Petmezas et al. Automated atrial fibrillation detection using a hybrid cnn-lstm network on imbalanced ecg datasets. *Biomedical Signal Processing and Control*, 63:102194, 2021. ISSN 1746-8094. doi: <https://doi.org/10.1016/j.bspc.2020.102194>. URL <https://www.sciencedirect.com/science/article/pii/S1746809420303323>.
- [50] V. Jahmunah, E.Y.K. Ng, Tan Ru San, and U. Rajendra Acharya. Automated detection of coronary artery disease, myocardial infarction and congestive

- heart failure using gaborcnn model with ecg signals. *Computers in Biology and Medicine*, 134:104457, 2021. ISSN 0010-4825. doi: <https://doi.org/10.1016/j.combiomed.2021.104457>. URL <https://www.sciencedirect.com/science/article/pii/S0010482521002511>.
- [51] Hao Dai, Hsin-Ginn Hwang, and Vincent S. Tseng. Convolutional neural network based automatic screening tool for cardiovascular diseases using different intervals of ecg signals. *Computer Methods and Programs in Biomedicine*, 203:106035, 2021. ISSN 0169-2607. doi: <https://doi.org/10.1016/j.cmpb.2021.106035>. URL <https://www.sciencedirect.com/science/article/pii/S0169260721001103>.
- [52] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000. doi: 10.1161/01.CIR.101.23.e215. URL <https://www.ahajournals.org/doi/abs/10.1161/01.CIR.101.23.e215>.
- [53] Laguna P., Mark R.G., Goldberger A.L., and Moody G.B. A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg. *Computers in Cardiology*, 24:673–676, 1997. URL <https://archive.physionet.org/physiobank/database/qtdb/doc/index.shtml>.
- [54] Wikipedia contributors. Pr interval — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=PR_interval&oldid=1119553835.
- [55] Willi Kaiser, Thomas S. Faber, and Martin Findeis. Automatic learning of rules: A practical example of using artificial intelligence to improve computer-based detection of myocardial infarction and left ventricular hypertrophy in the 12-lead ecg. *Journal of Electrocardiology*, 29:17–20, 1996. ISSN 0022-0736. doi: [https://doi.org/10.1016/S0022-0736\(96\)80004-5](https://doi.org/10.1016/S0022-0736(96)80004-5). URL <https://www.sciencedirect.com/science/article/pii/S0022073696800045>.
- [56] Nicholas Hughes, Lionel Tarassenko, and Stephen J Roberts. Markov models for automated ecg interval analysis. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL <https://proceedings.neurips.cc/paper/2003/file/b23975176653284f1f7356ba5539cfcb-Paper.pdf>.
- [57] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985. doi: 10.1109/TBME.1985.325532.

- [58] J.P. Martinez, R. Almeida, S. Olmos, A.P. Rocha, and P. Laguna. A wavelet-based ecg delineator: evaluation on standard databases. *IEEE Transactions on Biomedical Engineering*, 51(4):570–581, 2004. doi: 10.1109/TBME.2003.821031.
- [59] Hedayat Abrishami. Supervised ecg interval segmentation using lstm neural network. *Research Gate*, 08 2018.
- [60] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.