João Pedro de Lima Pestana

Bachelor of Science in Biomedical Engineering

Detection of abnormalities in ECG using Deep Learning

Dissertation submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Biomedical Engineering

Adviser: Prof. Doctor Hugo Gamboa, Auxiliar Professor,
NOVA University of Lisbon

Examination Committee

Chairperson:	Doctor Ricardo Nuno Pereira Verga e Afonso Vigário
Raporteur:	Doctor André Valério Raposo Carreiro
Member:	Doctor Hugo Filipe Silveira Gamboa

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

October, 2019

Detection of abnormalities in ECG using Deep Learning

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] `https://github.com/joaomlourenco/novathesis` [2] `http://www.di.fct.unl.pt`

Acknowledgements

First of all, I would like to thank my adviser, Professor Hugo Gamboa, for believing in me and giving me the opportunity to work on this project that motivated me to be a better Biomedical Engineer.

To David Belo, who was always available to help me, support me and teach me so many things. Without him, I would not have developed so much in the past year and I am extremely grateful to have such an amazing collaborator. Thank you to everyone from Biomedical Engineering Laboratory of LIBPhys, for keeping me company in this journey and welcoming me with joy and kindness.

To FCT NOVA, my home for the last 5 years, thank you for making me the person I am today. To BEST Almada, my other home for the past 4 years, thank you for teaching me to be a better human, for all the great friendships I was able to forge across Europe, for believing me to lead this organization for a year with my incredible Grasshoppers.

To my friends, for supporting me no matter what, for being my family away from home, I couldn't do any of this without your help and love. Especially to my Amêgos, thank you for being my support system.

The biggest thanks to my parents, brothers and sister for always believing in my potential, for understanding my frustrations and for lifting me up when I always needed. For my nieces and nephews, for motivating me to be a good role model to you.

"Somos a prova de que o Universo é bom e conspira a nosso favor."

Abstract

A significant part of healthcare is focused on the information that the physiological signals offer about the health state of an individual. The Electrocardiogram (ECG) cyclic behaviour gives insight on a subject's emotional, behavioral and cardiovascular state. These signals often present abnormal events that affects their analysis. Two examples are the noise, that occurs during the acquisition, and symptomatic patterns, that are produced by pathologies.

This thesis proposes a Deep Neural Networks framework that learns the normal behaviour of an ECG while detecting abnormal events, tested in two different settings: detection of different types of noise, and; symptomatic events caused by different pathologies. Two algorithms were developed for noise detection, using an autoencoder and Convolutional Neural Networks (CNN), reaching accuracies of 98,18% for the binary class model and 70,74% for the multi-class model, which is able to discern between base wandering, muscle artifact and electrode motion noise. As for the arrhythmia detection algorithm was developed using an autoencoder and Recurrent Neural Networks with Gated Recurrent Units (GRU) architecture. With an accuracy of 56,85% and an average sensitivity of 61.13%, compared to an average sensitivity of 75.22% for a 12 class model developed by Hannun et al. The model detects 7 classes: normal sinus rhythm, paced rhythm, ventricular bigeminy, sinus bradycardia, atrial fibrillation, atrial flutter and pre-excitation.

It was concluded that the process of learning the machine learned features of the normal ECG signal, currently sacrifices the accuracy for higher generalization. It performs better at discriminating the presence of abnormal events in ECG than classifying different types of events. In the future, these algorithms could represent a huge contribution in signal acquisition for wearables and the study of pathologies visible in not only in ECG, but also EMG and respiratory signals, especially applied to active learning.

Keywords: Electrocardiogram, Signal Processing, Deep Learning, Artificial Intelligence, Arrhythmia Detection, Noise Detection

Resumo

Uma parte significante dos cuidados de saúde foca-se na informação que os sinais fisiológicos oferecem acerca do estado de saúde de um indivíduo. O Eletrocardiograma (ECG) fornece discernimento do estado emocional, comportamental e cardiovascular de um sujeito. Frequentemente, estes sinais apresentam eventos anormais que afetam a sua análise. Dois exemplos são o ruído, que ocorre durante a aquisição, e padrões sintomáticos, que são produzidos por patologias.

Esta tese propõe uma *framework* de Redes Neuronais Profundas que aprende o comportamento normal de um ECG, também detetando eventos anormais, testada em duas configurações diferentes: deteção de ruído, e; eventos sintomáticos causados por diferentes patologias. Dois algoritmos foram desenvolvidos para a deteção de ruído, utilizando um *autoencoder* e Redes Neuronais Convolucionais, alcançando valores de exatidão de 98,18% para o modelo de classe binária e 70,74% para o modelo de múltiplas classes, que é capaz de diferenciar entre ruído da linha de base, de artefatos musculares e do movimento dos elétrodos. Quanto ao algoritmo de deteção de arritmias, foi utilizando um *autoencoder* e Redes Neuronais Recursivas com arquitetura GRU, alcançando uma exatidão de 56,85% e uma sensibilidade média de 61.13%, comparada com a sensibilidade média de 75.22% para um modelo de 12 classes desenvolvido por Hannun et al. O modelo deteta 7 classes: ritmo sinusal normal, ritmo pacemaker, bigeminismo ventricular, bradicardia sinusal, fibrilação atrial, flutter atrial e pré-excitação.

Foi concluído que o processo de aprendizagem das features do sinal de ECG normal, atualmente sacrifica o valor de exatidão por maior generalização. Apresenta melhor desempenho a discriminar a presença de eventos anormais no ECG do que em classificar diferentes tipos de eventos. No futuro, estes algoritmos poderão contribuir significativamente na aquisição de sinal para *wearables* e para o estudo de patologias visíveis não só no ECG, como também em EMG e sinais respiratórios, especialmente aplicados em active learning.

Palavras-chave: Eletrocardiograma, Processamento de Sinal, Aprendizagem Profunda, Inteligência artificial, Deteção de Arritmias, Deteção de Ruído

Contents

List of Figures xv

List of Tables xvii

Acronyms xix

1 Introduction 1
- 1.1 Motivation and Context 1
- 1.2 Objectives 2
- 1.3 Thesis Overview 3

2 Theoretical Concepts 5
- 2.1 Biosignals 5
  - 2.1.1 Electrocardiogram 5
  - 2.1.2 Biosignal Acquisition and Processing 12
- 2.2 Neural Networks 12
  - 2.2.1 Structure 13
  - 2.2.2 Recurrent Neural Networks 13
  - 2.2.3 Convolutional Neural Networks 14
- 2.3 Evaluation 16
  - 2.3.1 Confusion Matrix 16
  - 2.3.2 Evaluation Metrics 17
  - 2.3.3 Receiver Operating Characteristic Curve 18

3 State of the Art 19
- 3.1 Noise Detection in ECG 19
- 3.2 Abnormal event detection in ECG 20

4 Methods 23
- 4.1 Context 23
- 4.2 Technological Materials 25
- 4.3 Architectures 25
  - 4.3.1 Autoencoder 26
  - 4.3.2 Noise detection neural network 28

4.3.3 Arrhythmia detection neural network 30
4.4 Data 31
4.4.1 Autoencoder 31
4.4.2 Noise detection neural network 32
4.4.3 Arrhythmia detection neural network 32
4.5 Signal Preprocessing 32
4.5.1 Subsampling 33
4.5.2 Smoothing 34
4.5.3 Normalization 34
4.5.4 Segmentation 34
4.6 Training 35

5 Results **37**
5.1 Autoencoder 37
5.2 Noise detection neural network 38
5.2.1 Binary noise detection model 40
5.2.2 Multi-class noise detection model 41
5.3 Arrhythmia detection model 45

6 Conclusions **49**
6.1 Conclusions 49
6.2 Future Work 50

Bibliography **53**

List of Figures

2.1 The ECG cycle 6
2.2 Sinus bradycardia 8
2.3 Atrial fibrillation 8
2.4 Atrial flutter 9
2.5 Premature ventricular contraction 9
2.6 Paced rhythm 9
2.7 Pre-excitation 10
2.8 Baseline wander 11
2.9 Electrode motion 11
2.10 Muscle activation interference 11
2.11 Schematic of a single neuron 14
2.12 Comparision between a simple NN and a DNN 14
2.13 Unfolding of a recursive neural network 15
2.14 LeCun Net 15
2.15 Confusion matrix 17
2.16 ROC curves 18

4.1 GRU architecture 24
4.2 Encoder 26
4.3 ReLU activation function 27
4.4 Hyperbolic tangent activation function 28
4.5 Model architecture for the noise detection, with N classes 28
4.6 Softmax activation function 29
4.7 Dense block used for the 2-class noise detection model 29
4.8 Model architecture for the arrhythmia detection 30
4.9 Dense block used for the arrhythmia detection model 31
4.10 Preprocessing steps 33

5.1 Curve that describes evolution of the Mean Squared Error (MSE) over the training phase 38
5.2 Mean and standard deviation of the mean squared error 39
5.3 ECG 9 signal reconstruction 39
5.4 ECG 30 signal reconstruction 40

5.5 Confusion matrix for binary noise detection 41
5.6 t-SNE representation of NS and NAS 42
5.7 Detection results for binary noise model 43
5.8 Graphical representation and noise detection of the signal acquired using Biosignalsplux wireless device and Open Signals software. 44
5.9 Confusion matrix for multi-class noise detection 45
5.10 t-SNE representation of NS, BW, EM and MA 45
5.11 Confusion matrix for arrhythmia detection 46
5.12 Confusion matrix for arrhythmia detection with AFIB and AFL classes combined 47

List of Tables

5.1 Binary noise detection model: classification performance (%) 41

5.2 Composition of the training data. 42

5.3 Multi-class noise detection model: classification performance for each class (%) 44

5.4 Arrhythmia detection model: classification performance for each class (%) . 46

5.5 Arrhythmia detection model with classes Atrial Fibrillation (AFIB) and Atrial Flutter (AFL) merged: classification performance for each class (%) 47

Acronyms

A/D Analog-to-Digital.

AFIB Atrial Fibrillation.

AFL Atrial Flutter.

AI Artificial Intelligence.

ANN Artificial Neural Networks.

ANS Autonomic Nervous System.

AUC Area Under Curve.

AV Atrioventricular.

B Ventricular Bigeminy.

BPM beats per minute.

BPTT Backpropagation Through Time.

BW Baseline wandering.

CART Classification and Regression Trees.

CNN Convolution Neural Networks.

CPU Central Processing Unit.

DNN Deep Neural Networks.

ECG Electrocardiogram.

EEG Electroencephalogram.

EM Electrode motion.

EMG Electromyography.

FN False Negative.

FP False Positive.

FPR False Positive Rate.

GPU Graphics Processing Unit.

GRU Gated Recurrent Units.

HR Heart Rate.

HRV Heart Rate Variability.

ICA Independent Component Analysis.

KNN K-nearest Neighbour.

MA Muscle artifact.

MLP Multi Layer Percepton.

MSE Mean Squared Error.

NAS Noise affected signal.

NN Neural Networks.

NS Normal signal.

NSR Normal Sinus Rhythm.

P Paced Rhythm.

PNS Paraympathetic Nervous System.

PREX Pre-excitation.

ReLU Rectified Linear Unit.

RF Random Forest.

RNN Recurrent Neural Networks.

ROC Receiver Operating Characteristic.

SA sinoatrial.

SBR Sinus Bradycardia.

SNR Signal-to-Noise Ratio.

SNS Sympathetic Nervous System.

t-SNE T-Distributed Stochastic Neighbor Embedding.

TN True Negative.

TP True Positive.

TPR True Positive Rate.

Chapter 1

Introduction

1.1 Motivation and Context

In the context of medicine and healthcare in general, physiological signals offer information about the health state. Among the electrophysiological data available, Electrocardiogram (ECG) prove to be one of the most relevant and, therefore, studied, where the variability of its cyclic behaviour may provide evidence of emotional, behavioral and cardiovascular changes and pathologies. By studying its characteristics this biosignal is the gold standard of providing effective diagnostics for cardiac diseases in non-invasive way. Nowadays, wearables grant the possibility of the detection of rare cardiac episodes, due to the ambulatory properties of these devices, providing monitoring of ECG during long periods of time. The early detection of these episodes during our daily lives increases the chance of a better lifestyle for people. Automatic detection by using signal processing and pattern recognition methods may provide an asset in providing relevant information without having to rely on hospital resources [1, 2].

With a better understanding of some pathologies and the development of diagnostic methods and therapies allied to the evolution of technology in the medical field, greater healthcare expectations emerged in terms of efficiency. The fast and personalized care is considered a baseline and the demand for higher performance systems increases through time [3]. In response, Artificial Intelligence (AI) is being used and merged with the medical field, by using its capacity to learn from high density of data to find complex relationships between various parameters. These technologies are useful in assisting medical practitioners for decision making processes, not only in diagnosis but also in treatment by monitoring patients, studying drug efficacy tests, between other applications [3–5].

Entering this new paradigm of medicine, remote streams of biomedical data between the patient and physicians increases the volume of stored data at the cost of quality [3].

Wearables introduce a substantial amount of data with increased noise and motion artifacts in the acquired signals. While in this setting, traditional machine learning methods became less reliable in detecting patterns and, therefore, discerning pathological events. In order to tackle this problem, deep learning is introduced in the medical field [5].

The genesis of AI opened new doors in several fields of science. This new way of reflecting human-computer interaction revealed to be ambitious and controversial, with its interest growing progressively each year [4]. Artificial Neural Networks (ANN) were created when scientists tried to mimic mathematically the intrinsic wiring of the human brain and the nervous system pathways [5]. The evolution of these networks led to more complex architectures. Deep Neural Networks present themselves as a Machine Learning algorithm that learns higher abstractions of data, while dealing with a large amount of data and the decrease the need for feature engineering [3, 5]. These algorithms have proven to provide accurate and reliable results in pattern recognition [4] while being able to tolerate noise affected biosignals [6].

Today, these technologies have been studied heavily in image recognition, natural language processing and human-speech recognition, but applications in the healthcare field are still limited [3]. The ECG signals provides an opportunity to use Deep Neural Networks (DNN) for pattern recognition and decision due to its cyclic behaviour. By learning the morphology of a Normal Sinus Rhythm (NSR), it is hypothesised that by analysing the divergence of an altered morphology this architectures may prove efficient in detecting anomalies. Belo and colleagues [7] have been able to learn, synthesize and compare ECG to develop a biometric system based on the divergence between individual traits. Therefore detecting differences between the provided input and a similar signal from a different source has been proven to be successful in in the LIBPhys Biosignals group.

This thesis proposes a similar deep learning approach to study the ECG biosignal, by comparing the morphology of a normal and clean ECG signal with two types of anomalies: noise affected segments and symptomatic occurrences. The development of this architecture will be a step in providing assistance to medical professionals in the detecting cardiac pathologies and also in the process of analysing the quality of the ECG signal. By detecting various levels of noise it possible to readjust in real-time the external variables that could be the cause of its existence, such as the position of the electrodes, respiratory breathing, muscle contraction, motion and others.

1.2 Objectives

The main objective of this thesis is to develop a DNN that detects and evaluates ECG signals in two specific abnormalities: the presence of noise and pathological events. The scope of this work comprises:

- Application of algorithm on ECG signals;

- Detection and classification of abnormal patterns on ECG biosignals:
  - by detecting muscle artifacts, electrode motion and baseline wandering noise on ECG;
  - by detecting symptomatic episodes of paced rhythm, ventricular bigeminy, sinus bradycardia, atrial fibrillation and atrial flutter and pre-excitation.
- Validation and improvement of predictive model with a validated and publicly available database of ECG signals Physionet database [8];

1.3 Thesis Overview

This dissertation comprises seven chapters. The present chapter introduced the motivation and context of the thesis and its main objectives. Chapter 2 introduces the theoretical concepts on biosignals, neural networks and machine learning evaluation techniques, all relevant for the comprehension of the work developed. A review of the literature regarding abnormal event detection in ECG is provided in chapter 3. Chapter 4 presents the methodologies used, the context in which these methods are used, the technological materials used, the developed DNN architectures, a description of the data used, as well as data preparation (preprocessing techniques) and training parameters. The results are presented in chapter 5 and chapter 6 presents the main conclusions of this thesis and guidelines for future work.

Chapter 2

Theoretical Concepts

In this chapter, certain theoretical concepts are explored, in order to understand the dissertation. First, it introduces the concept of biosignals, relevant to understand physiological and biological events, focusing on the ECG and the relevance of noise and pathological events present in this type of signal. Furthermore, biosignal acquisition and processing are also analysed. Then, notions of Neural Networks (NN) are presented, such as their structure and two types NN used in deep learning. Finally, some machine learning evaluation techniques are explained, more specifically the confusion matrix, the Receiver Operating Characteristic (ROC) curve and other evaluation metrics.

2.1 Biosignals

In order to understand physiological mechanisms and biological events, electrophysiologic signals (biosignals) are extracted from the human body and analysed using various digital methods that involve amplification, filtering, digitization, processing and storage. Biosignals like the Electroencephalogram (EEG), the ECG and the Electromyography (EMG), can be defined as records of biological events, in space and time, from the brain, cardiac and muscle activity, respectively [9].

The machine learning approach to analyse biosignals and understand the physiological phenomena behind them has been in constant development, showing evidence to be a promising path to medical innovation. In this thesis, DNN will be applied to analyse and classify ECG biosignals, which will be introduced bellow.

2.1.1 Electrocardiogram

The ECG measures the electrical activity generated by the heart activity in relation to time by inserting electrodes on the skin. It is a prominent tool amongst physicians for its helpful insight regarding cardiac disease analysis and diagnosis [10]. Through analysis of the signal's characteristics it is possible to detect and measure the heart's rhythm, electrical conduction, anatomy and possible ischemic locations [11].

The cardiac signal is generated by the polarization and depolarization of the cardiac muscles, activated by the electrical impulses generated by specialized myocardial cells. Furthermore, the depolarization allows the heart muscles to contract and pump the blood to various regions of the body [11].

Due to the sequential process of blood pumping performed by the heart's muscles, the cardiac signal can be described as being cyclic. Furthermore, the circulatory system is subjected to regulation activities by the nervous system which affect the signal's behaviour, adding a random component, which is why it is crucial to consider a healthy ECG signal as a series of successively occurring events [12]. These events can be observed by analysing the ECG signal, by inspecting the presence, shape and amplitude of the waves, the relationship between them, and the various segments and intervals. Waves are identified by the letters P, Q, R, S, T and U, although the latter doesn't always occur and the ventricular contraction is often associated by the QRS complex. Segments can be defined as the duration in milliseconds between sequential waves. Intervals are relevant time durations that involve waves and segments that are consecutive. Figure 2.1 presents two cardiac cycles and its waves, intervals and segments [11].

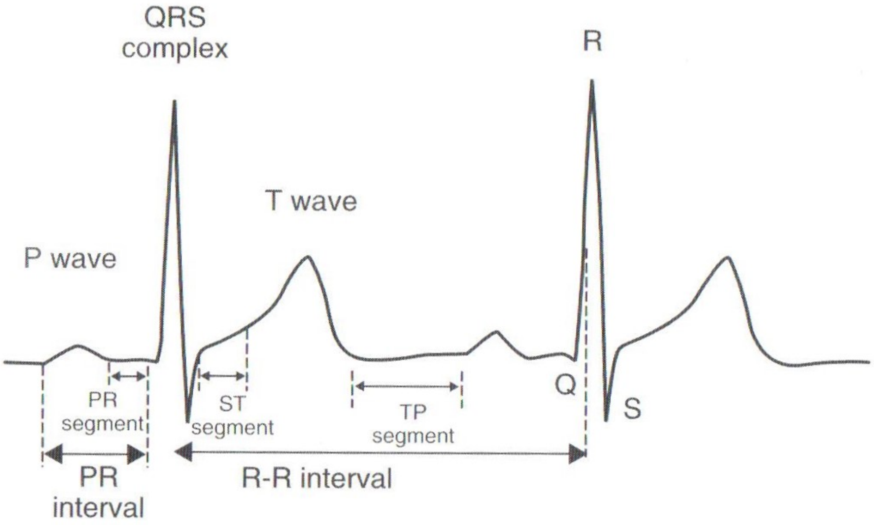


Figure 2.1: The ECG cycle [13].

In order to perform a diagnosis on a person's cardiac health and detect possible abnormalities of their heart functions, waves can be described as reflections of the heart's physiological events. The initial P wave occurs just before the atrial contraction and reflects the depolarisation of the atrial myocardium. The QRS-complex indicates the start of ventricular contraction, responsible for ejecting blood out of the heart, and corresponds to the depolarisation of the ventricular myocardium. The end of ventricular ejection gives rise to the T wave that indicates repolarization of the ventricular myocardium, responsible for its recovery and announcing a new cycle. The U wave remains uncertain in its connection to the physiological mechanisms of the heart, although some theories indicate that it could reflect the repolarization of endocardial structures or late depolarisation of the ventricular myocardium [11].

In addition to the investigation regarding various events inside a given cardiac cycle in the ECG signal, measuring the Heart Rate (HR) and Heart Rate Variability (HRV) could give further insight on a person's health. The HR is calculated by measuring the time interval between two consecutive R waves (RR interval), while the HRV is given by the variation of HR along time [14].

Since the heart is regulated by the Autonomic Nervous System (ANS), either by Sympathetic Nervous System (SNS) stimulation or by Paraympathetic Nervous System (PNS) inhibition, changes in HR, stroke volume and systemic vasoconstriction, for example, are put into action in order for the human body to adapt to unexpected stimuli. Furthermore, the HRV is thought to be a measure of the ANS ability to maintain the human body's healthy status and can be affected by cardiac output, diabetes, age, alcoholism, renal failure, among other factors[14].

2.1.1.1 Pathologies

The list of cardiac pathologies is vast and its symptoms can be inspected in the ECG. The most common causes for cardiac diseases are electrical impulse conduction blocks, enlargement of the myocardium, pericarditis, electrolyte imbalance, respiratory diseases, drugs, hypothermia and accessory conduction pathways[11].

NSR comprises a heart rate between 60 and 100 beats per minute (BPM) at rest, while maintaining the cycle characteristics mentioned above. Variations of the RR interval may occur due to respiration, which can also change HR values to below 60 and above 100 beats per minute. The latter situation is called sinus tachycardia and the former, bradycardia. [11, 15].

Arrhythmias can be defined as rhythms that are considered abnormal, considering the NSR, that are related to malfunctions in the heart's electrical activity and changes in the heart tissue [11, 15]. They can be detected in ECG by analysing the heart rate, the presence or absence of certain waves, the duration of intervals, wave overlap, abnormal timing of cardiac events, the shape, size and direction of the waves, between others. Arrhythmias may be originated from various issues, such as, the generation or propagation of the electrical impulse in or outside of the sinoatrial (SA) node, lack of muscle coordination, premature contractions, changes in the automaticity of heart muscles, extra contractions, delay or lack of propagation of the impulse. The repetitive nature of these events, which often leave repercussions on cardiac health, can be isolated and can be characterised by their generation and manifestation, being categorized as following [11]:

- Sinus node arrhythmias - sinus bradycardia (Figure 2.2), sinus arrest and SA exit block;
- Atrial arrhythmias - wandering atrial pacemaker, premature atrial contractions, atrial ectopic tachycardia, atrial multifocal tachycardia, atrial flutter (Figure 2.4) and atrial fibrillation (Figure 2.3);
- Junctional arrhythmias - premature junctional contractions, junctional escape rhythm, accelerated junctional rhythm, junctional tachycardia and paroxysmal supraventricular tachycardia;
- Ventricular arrhythmias - premature ventricular contractions (Figure 2.5), ventricular tachycardia, ventricular fibrillation, ventricular escape rhythm, accelerated idioventricular rhythm and ventricular asystole;
- Atrioventricular (AV) blocks - wenkebach, mobitz type II, advanced AV block and complete AV block;
- Bundle branch blocks - right bundle branch block and left bundle branch block (Figure 2.7);
- Fascicular blocks - left anterior fascicular blocks and left posterior fascicular block.

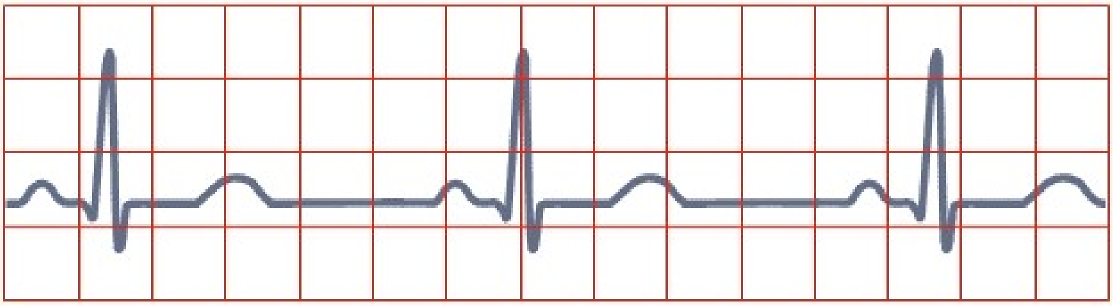


Figure 2.2: Sinus bradycardia: characterized by a sinus rhythm inferior to 60 BPM [16].

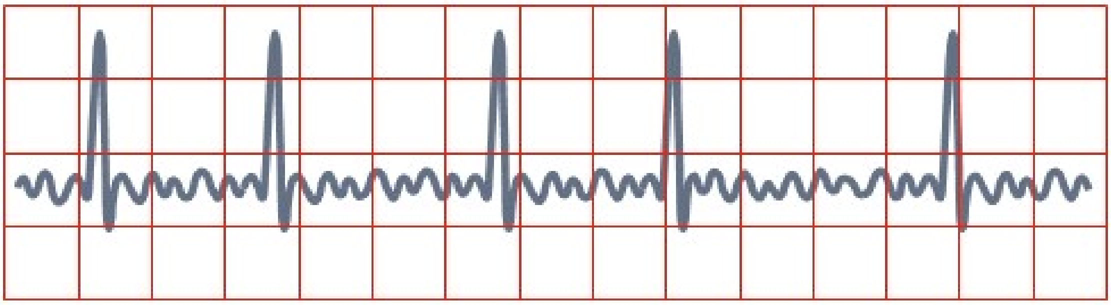


Figure 2.3: Atrial fibrillation: chaotic activation in the atria, which produces irregular fluctuations in the baseline, with usually a normal QRS wave [16].

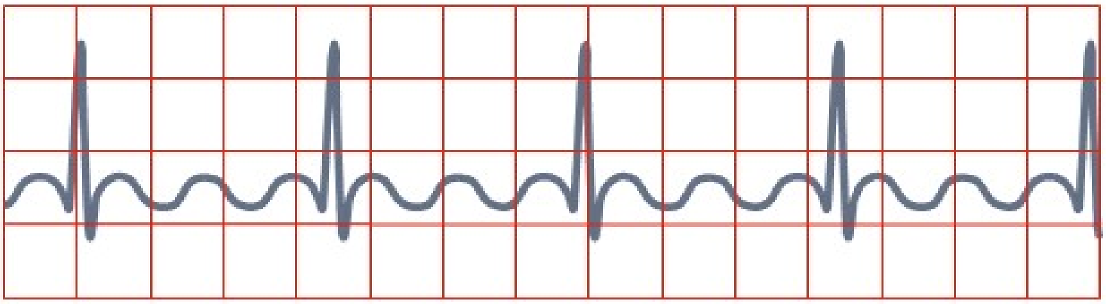


Figure 2.4: Atrial flutter: elevated HR causes rapid flutter waves that cover the isoelectric interval between the T and P waves [16].

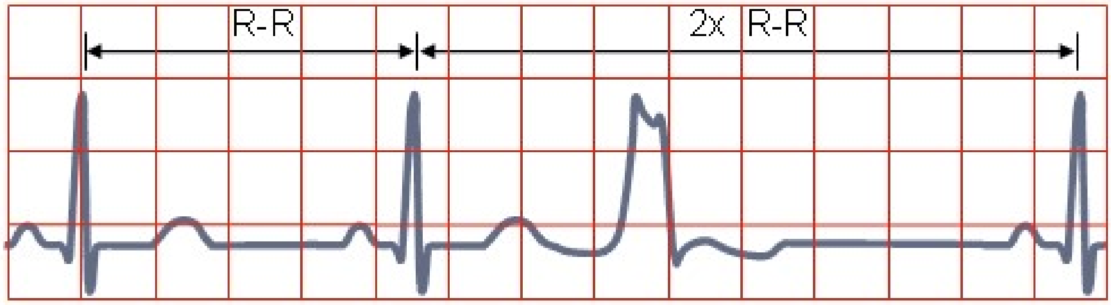


Figure 2.5: Premature ventricular contraction: has its origin in the atrium or in the AV node, creating an early contraction that generates an abnormal QRS complex, without a P wave associated to it. [16]. A Ventricular Bigeminy occurs when every other beat is a premature ventricular contraction [17].

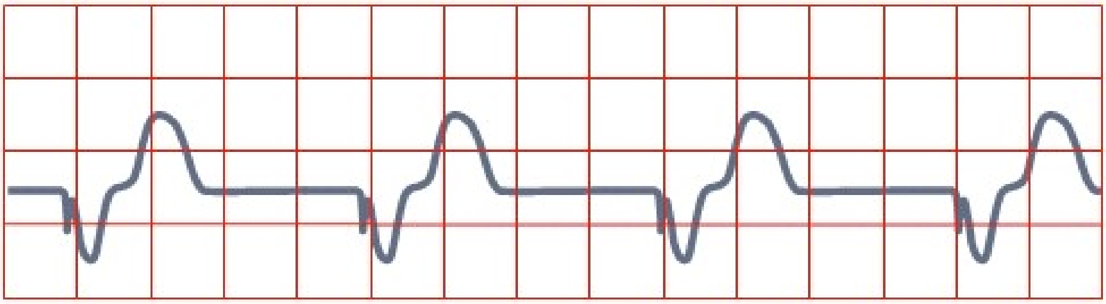


Figure 2.6: Paced rhythm: ventricular rhythm that originates from a cardiac pacemaker, creating wide QRS complexes preceded by a pacer impulse spike [16].

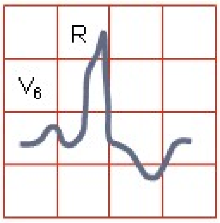


Figure 2.7: Wolff-Parkinson-White syndrome (Pre-excitation): a type of left bundle branch block. An early ventricular excitation causes the formation of the delta wave which transforms the QRS complex to a wider, irregular shape and shortening the interval between the P wave and the QRS complex [16].

Another type of abnormal rhythm is the pacer (or paced) rhythm that is generated by a cardiac pacemaker, as can be seen in Figure 2.6 [16].

Besides arrhythmias, certain cardiac pathologies can be diagnosed through the ECG signal such as enlargement of the myocardium (atrial enlargement, ventricular hypertrophy), pericarditis, electrolyte imbalance (hyperkalemia, hypokalemia, hypercalcemia and hypocalcemia), drug effects, pulmonary disease (chronic obstructive pulmonary disease, acute pulmonary embolism and chronic cor pulmonale), hypothermia, and many others. By analysing several aspects, cardiologists can also identify presence of ischemia, injury and infarction [11].

2.1.1.2 ECG noise

Not all abnormalities in the ECG signal are pathological. Very frequently, these signals are affected by many types of noise [11]. Noise can be defined as any signal that is not related to the heart's electrical activity that obstructs the ECG signal through interference [18]. Usually, the signal is obtained in a controlled environment, although the usage of ambulatory ECG is more associated with certain variables that influence the noise [19].

Noise interference has many possible sources, which can be classified as biological [20], also known as physiological [18], like motion artifact and muscle contraction, and environmental [20] or non-physiological noise [18], related to nearby electrical devices and variables associated with the equipment.

Bellow we can observe different types of noise, for example Figure 2.8 shows artifacts, called baseline wander, that are caused by movement such as the contraction and relaxation of the thoracic cage muscles, which is evident if the event is periodic. Noise affected signals can also originate from the misplacement of electrodes and loss of contact between the sensor and the skin, as seen on Figure 2.9 [13, 20]. Another type of noise is power line interference, which refers to the 50/60 Hz obstruction caused by the alternating current due to electromagnetic noise induction [21]. Similar effects can be seen with the presence of nearby radio frequencies. As for the activation of the muscles shown in Figure 2.10 resulted from the EMG signal interference in the ECG signal, revealing muscle artifacts.

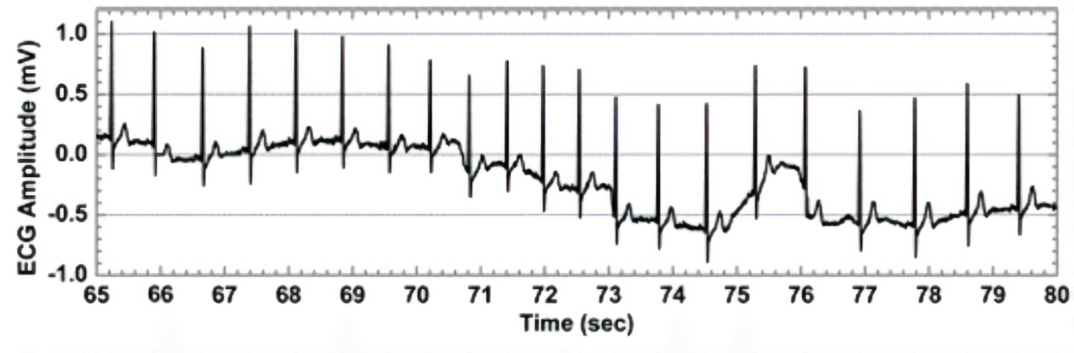


Figure 2.8: Baseline wander [22].

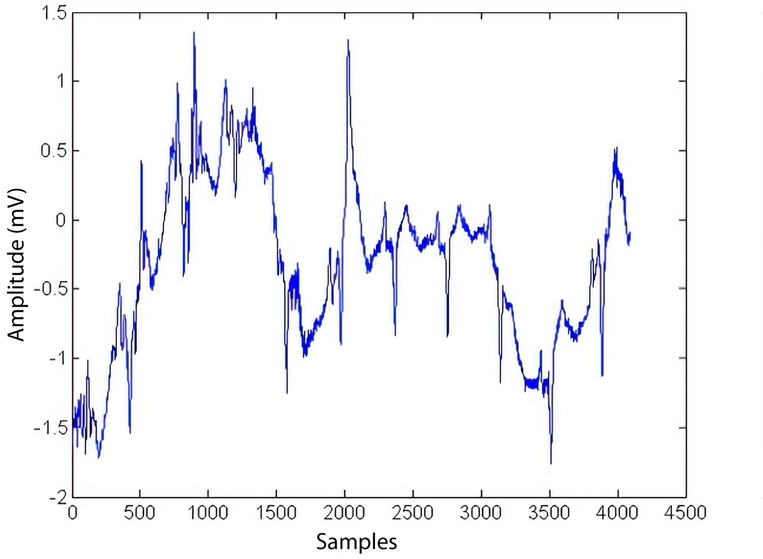


Figure 2.9: Electrode motion [22].

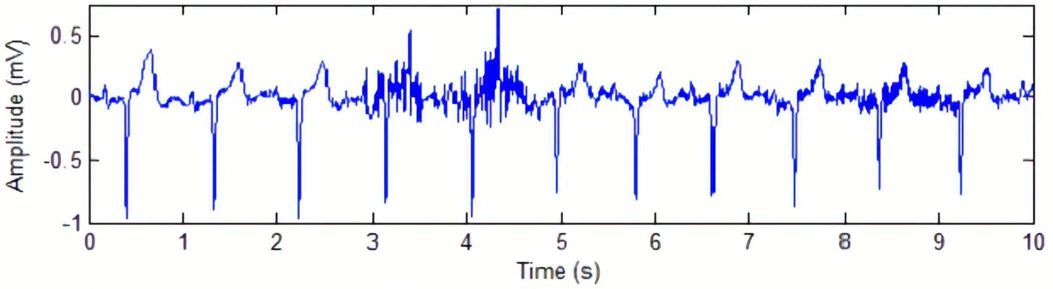


Figure 2.10: Muscle activation interference [22].

2.1.2 Biosignal Acquisition and Processing

Biosignals can be acquired through body sensors or off-person by placing electrodes on objects, so that we can obtain important information about biological mechanisms and phenomena. However, in order to extract this information, biosignals need to be processed. Possible causes that mask the important information may be due traces of other biological signals present in the human body or additive noise due to electrode placement [18]. Consequently, it is important to perform preprocessing techniques such as amplification and filtering, in order to remove undesirable effects and to meet the hardware requirements [23].

In order to properly study the signals there is the necessity of extracting the relevant information while preserving the original signal waveform. Therefore, the preprocessed signal needs to be converted from a continuous time waveform into a digital signal, discretized in time and amplitude without extracting pathological alterations and producing any kind of distortion that may eliminate crucial events, like arrhythmias in the ECG signal. This conversion is made using an Analog-to-Digital (A/D) converter, which takes the sampling and quantization processes into action. In the sampling step, the signal is converted into a discrete samples in time and then, in the quantization step, amplitude values are given to those samples [23].

By converting a continuous analog signal into a discrete digital signal, many processing techniques can be applied, given that the latter can be described as an accurate representation of the former. Although, this this depiction is only authentic when in the A/D conversion phase the sampling theorem, otherwise known as the Nyquist theorem, is applied. This theorem requires that, in order to avoid aliasing, the highest frequency of the signal must be less that half of the sampling frequency [24]. Considering that the ECG signal comprises the frequencies between 0 and 40 Hz, the sampling frequency must not be no more than 80 Hz. Finally, the digital signal is ready to be processed. There are various techniques that could be implemented such as filtering, averaging, spectral estimators, among others [23].

2.2 Neural Networks

For many years, humanity has been confronted by complicated tasks in their day-to-day lives, especially when they are repetitive or when they involve finding patterns within the chaos, resulting in the inception of AI, a solution that relied on computer to perform and solve tasks that are difficult for human beings but relatively uncomplicated for computers. On the other hand, it is a harder task to simulate the problem solving people engage in simple tasks that feel intuitive and automatic, like recognizing faces or calligraphy through machine learning [25].

With the development of technology came the possibility to understand patterns in our everyday lives using algorithms that could mimic the human capacity, in order to achieve better results in less time. The creation of ANN was then motivated by the capacity of the human brain to compute data in a complex, nonlinear, and parallel way through neurons that connect and work together in order to navigate the world around us, through perception and pattern recognition, almost automatically [26].

The ability to perform these cognitive processes derives from the brain plasticity that allows neurons to adapt and develop according to a person's experience, in order to respond rapidly to new stimuli. Likewise, NN can be considered as machines that simulate how the brain performs a desired task, through learning and generalization. The learning process refers to the training phase of a NN by feeding raw input data that adjusts the synaptic weights of the network in order to reach a certain goal. A functional network can then use the model to produce generalizations, which can be explained as valid outputs for new inputs [26].

Nowadays, it's safe to say that machine learning is highly applicable for a variety of biosignal processing and classification, for they meet great results in complex applications, even when the signals contain a considerable amount of noise [9].

2.2.1 Structure

ANN are modeled after a simplistic version of the biological neural networks present in the brain by a network of neurons. An artificial neuron consists in a processing unit that performs simple operations. It contains connections that are characterized by its weight, which represents the strength the connection possesses. A given connection j, with weight w_{kj}, that is connected to neuron k, has an input signal associated, x_j, that is weighted and added to the other existing input signals, in order to perform a certain operation. This results in the creation of an output signal y_k, limited by the activation function in its amplitude, that is propagated to the next neuron. In Figure 2.11 we can observe a diagram of a single neuron, including the bias b_k, an external independent neuron that manipulates the net input of the activation function mentioned before, through the polarity of its output unit signal [9, 26].

The interconnected neurons are organized in layers and can be identified by input, output and between them one or more hidden layers. DNN are created when there are more than one hidden layers, being that a bigger number of layers offers more memory to the DNN, which provides a bigger capability of dealing with complex data. Figure 2.12 shows the comparison between a simple neural network, and a DNN in its structure [9, 27].

2.2.2 Recurrent Neural Networks

A Recurrent Neural Networks (RNN) is a particular type of neural network that can be described as a dynamic sequential data processing model where its current internal state is affected by the previous one, providing a memory based approach to learn and adapt

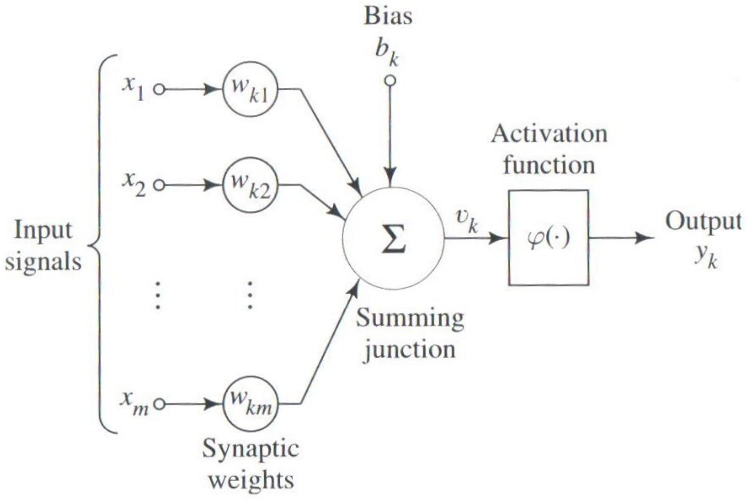


Figure 2.11: Schematic of a single neuron [26].

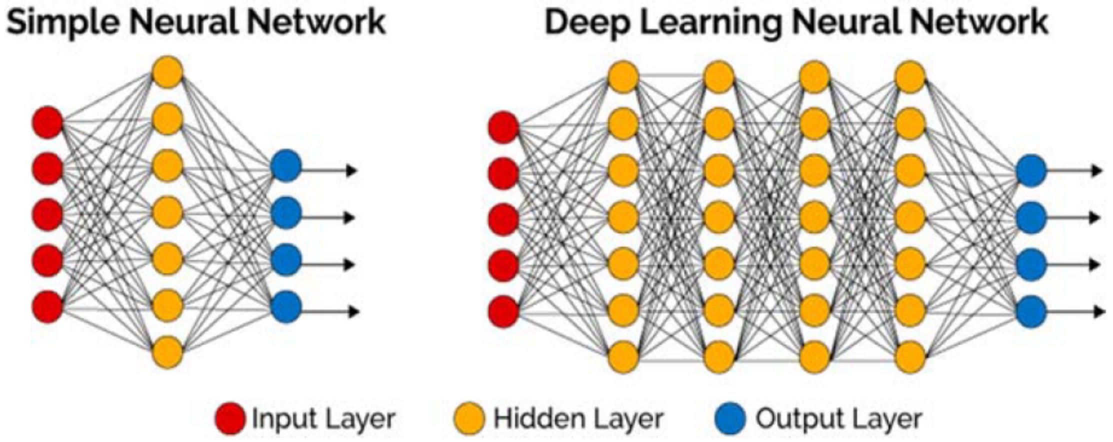


Figure 2.12: Comparison between a simple neural network and a DNN; both possess input and output layers, differing only in the number of hidden layers between them [27].

through data. One may represent this recursive approach by unfolding it, generating a very deep neural network (Figure 2.13) [28].

The internal memory factor contributes by altering the weights associated with the connections each iteration, generating the desired output that is then fed into the network's input in the next step [29].

Today, RNN are often used for speech recognition, machine translation, image recognition and characteristics definition, spelling correction and text generation, and other pattern recognition based approaches[29].

2.2.3 Convolutional Neural Networks

The Convolution Neural Networks (CNN) model, developed by Fukushima [30] and revised by LeCun et al [31] (Figure 2.14), bases on the brain's visual cortex neurobiology

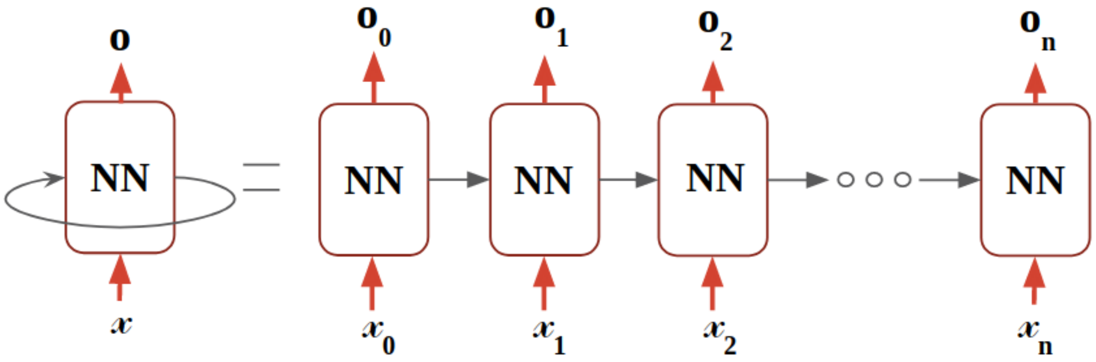


Figure 2.13: Unfolding of a recursive neural network

by connecting convolutional layers [32] in order to learn and organize itself by generating a combination of feature maps that provide insight to the pattern recognition at hand [33].

The convolutional layers generate feature maps of the data through convolutional filters, which are then combined and passed through the network to the pooling layers. The latter then performs statistical operations on the feature maps with the goal of extracting the essential features [34]. Finally, the fully connected layers represent the ones where each neuron of the previous layer is connected to the current layer, determining the number of classes [5].

This type of model has many applications in the real world, especially in object recognition, image classification and handwriting classification, being also applied in the medical field for diagnostic purposes [35]. Its main advantage lies on the fact that it reveals to be useful in data sets with a large number of nodes and parameters to be trained [32].

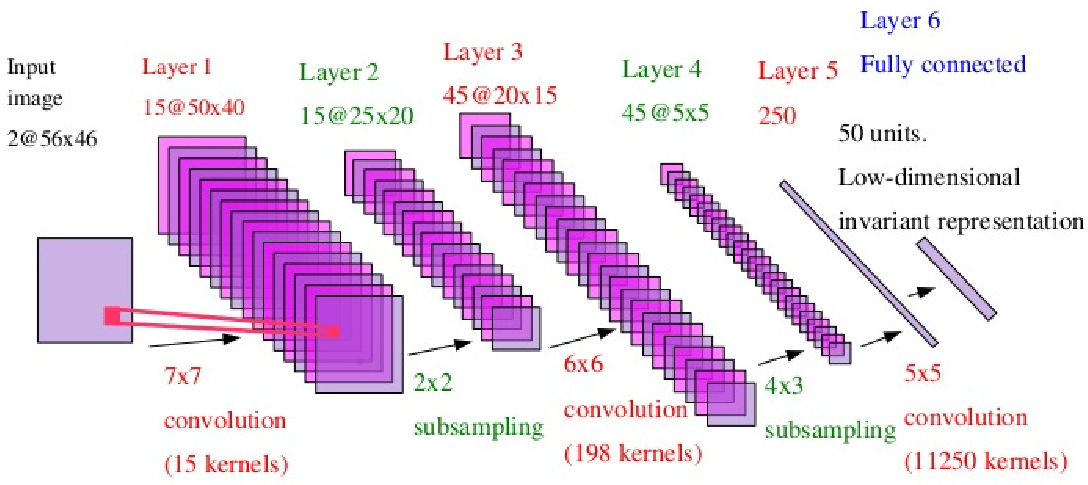


Figure 2.14: LeCun Net [36].

2.3 Evaluation

Throughout the development of the DNN algorithm, with the use of the relevant datasets, there comes the necessity to evaluate and perform experiments to determine the classifier's performance and ability to produce valid and trustworthy outcomes. Evaluation techniques are then applied in the process in order to improve the algorithm [37].

When it comes to model evaluation, it is important to note that machine learning is not concerned with the ability to learn and predict from the training data, but with the confidence in which the model predicts correct outcomes for new data, what we previously called generalization [38].

To evaluate the performance of a certain DNN model and to avoid over-fitting, it is necessary to cross-validate by dividing the training data into two parts. The majority of a given training set is then used for the algorithm's learning phase (fitting) and the remaining for model evaluation purposes, through error calculations based on the generalizations the classifier makes after the learning phase is performed. The former is called the training set and the latter, the validation set. Given the training and validation steps, a third data set (test set) that is independent from the previous ones is used to perform the real generalization evaluation of the model [37].

2.3.1 Confusion Matrix

Once the test step is performed, the results of the model classification is often displayed using a confusion matrix, containing rows for each actual class and columns for each predicted class [39].

That being said, each element of the matrix refers to the number of test examples that belongs to a certain class, indicated by the row that it is located, with a certain prediction given by the classifier, which is indicated by the column where it is placed [39].

This tool is often used to assess the performance of the model and identifying classes that are often badly predicted or mistaken for another. The optimal confusion matrix, that shows good confidence in the results, has a large number of examples down the main diagonal, where the predicted class and the actual class are the same, and a small amount of examples, or ideally none, in the other elements where the predicted class is different from the actual class [39].

A confusion matrix can also be represented using color scales, as seen on Figure 2.15. In this case, predictions are mapped by the value of sensitivity, meaning that a stronger relation between the true class (or label) and the predicted class is seen as a more intense shade of blue. By observing the matrix, one can observe, for example, that the class AVB_TYPE2 that refers to a type 2 atrioventricular block is sometimes mislabeled as a Wenckebach block, which may result from their similar expressions in the ECG signal [41].

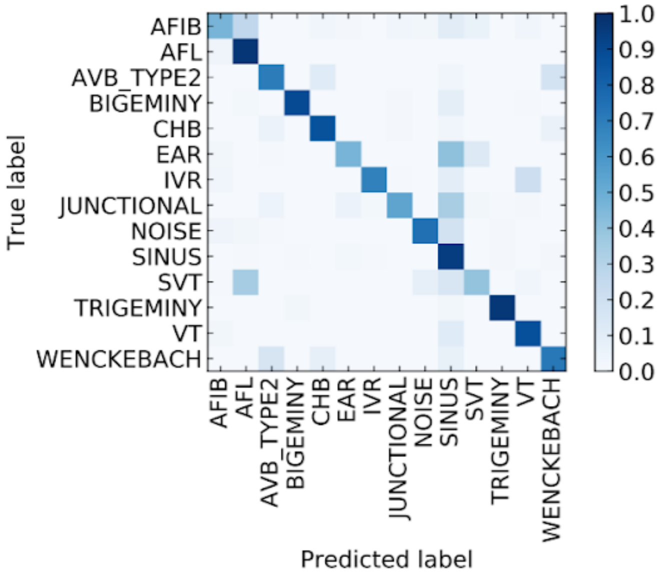


Figure 2.15: Confusion matrix adapted from [40]

2.3.2 Evaluation Metrics

When discussing classification evaluation it is important to mention various extensively used measures. For a binary class representation, with positive and negative as possible values, there are four possible cases to consider: True Positive (TP), when a positive is predicted correctly; True Negative (TN), when a negative is predicted correctly; False Positive (FP), when a negative is classified as positive, and; False Negative (FN) when a positive is classified as negative [38].

The accuracy of a classifier refers to the fraction of correct predictions and can be described by the following expression:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

Furthermore, a model's sensitivity, also known as True Positive Rate (TPR), can be defined as the proportion of true positives out of all positives, as it reflects the ability of the classifier to detect positives, and is calculated by:

$$sensitivity = \frac{TP}{TP + FN} \tag{2.2}$$

Finally, the specificity of a model relates to the proportion of negatives that are correctly identified, as it reflects the ability of the classifier to reject negatives, as such:

$$specificity = \frac{TN}{TN + FP} \tag{2.3}$$

As an example, in the medical field, sensitivity could refer to as the fraction of patients that are correctly diagnosed as to having a certain disease or condition, whereas specificity often is seen as the proportion of healthy people that were identified as to not having the condition that is being tested [39].

2.3.3 Receiver Operating Characteristic Curve

The ROC curve is a graphical plot that characterizes a certain model's predictions without resorting to class distribution or error costs [39]. It is especially useful to compare different models for its visual representation [37].

In the vertical axis the fraction of true positives out of all positives, i.e. the sensitivity, is depicted, while in horizontal axis, the proportion of false positives out of all the negatives, which is called the False Positive Rate (FPR) and can be calculated by 1 - specificity [37].

A good classifier has sensitivity near 1 and a FPR near 0, meaning that a good indicator of a model's performance is the shape of the ROC curve. If the curve is leaning closer to the left corner of the plot, better results are expected. In Figure 2.16 we can observe the ROC curves of classifiers A, B and C and conclude that by examining their paths, models B and C are preferred over model A [38].

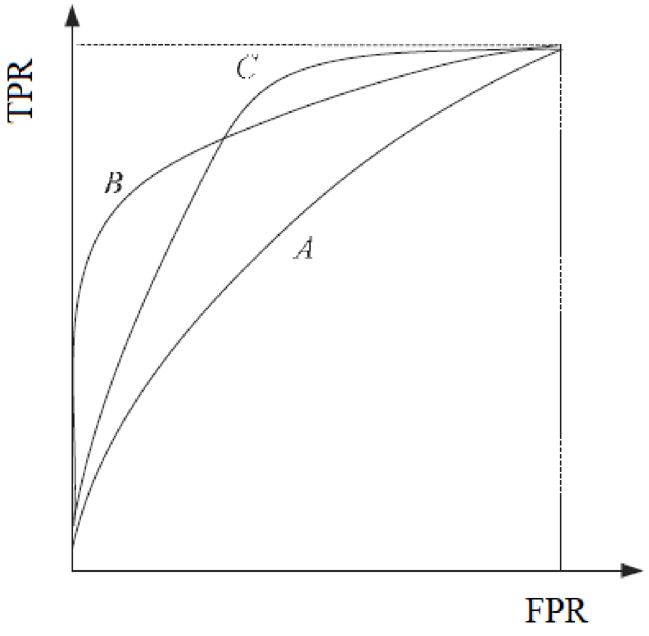


Figure 2.16: ROC curves of models A, B and C. Adapted from [38]

Even though this method provides a visual analysis of the models, another evaluating approach can be extracted by these graphics. The Area Under Curve (AUC) is calculated by integrating the functions that define the model's curves and an ideal classifier has an AUC of 1. This technique is especially advantageous when the visual representation of the curves of different classifiers doesn't quite clarify which one is better. Therefore, taking Figure 2.16 as an example, between B and C, the best classifier would be the one with the higher AUC value [38].

Chapter 3

State of the Art

Over the years, various methods have been studied for ECG noise and arrhythmia detection. The following sections show an overview of the general developments in these areas.

3.1 Noise Detection in ECG

With the development of ECG monitoring technologies, noise detecting and denoising algorithms and methods became a reality in the medical field. Various processing approaches have been studied, either by decomposing the ECG signal into secondary waves, through signal conditioning, ECG reconstruction by adaptive filtering, through comparison of subsequences inside the signal and by using ANN [18].

Despite the newfound popularity of ANN, its implementation in ECG noise detection and reconstruction at first hasn't been prominent [18]. However, in recent years, this machine learning method is becoming more popular in the study of ECG noise.

Rodrigues and Couto (2012) proposed a deep neural network approach for ECG denoising by using a feed forward method with three hidden layers. This particular method used all records from the MIT BIH Arrhythmia database, with added electrode motion noise. This algorithm was able to achieve good performance for ECG denoising and retain most of the information even for high level noise. This research showed promising advantages for Holter records, despite not being applicable in real time [42].

Ansari et al. (2018) developed a CNN to detect usable and unasable ECG segments, in terms of calculating the HRV. This algorithm was able to detect noise in ECG that would affect the QRS peak detection by determining a critical noise threshold. Then, by performing HRV analysis on the noisy signal without the usage of the algorithm and on the segments deemed usable by the DNN, lower error rates were obtained using the CNN model, proving to facilitate HRV analysis using wearable ECG devices. This approach resulted in an AUC of 0.96 for the classification of noise affected segments, compared to an AUC of 0.87 for a Support Vector Machine model that was also developed for this study [43].

A decision rule-based algorithm is proposed by Satija et al. (2018), in which ECG signals are first preprocessed using modified ensemble empirical mode decomposition, which discriminates the ECG components from relevant noise and artifacts. After this step, certain features are extracted such as maximum absolute amplitude, number of zero crossings and local maximum peak amplitude of the autocorrelation function. The proposed framework is able to classify the ECG signals between noise free ECG, Baseline wandering (BW), Muscle artifact (MA) noise, and combinations of BW with MA and BW and , totaling six signal groups. This model was able to achieve an accuracy of 98.93%. [44].

An implementation of ANN for noise detection in ECG is proposed by John et al. (2018), creating a 16 layer CNN which predicts one prediction per second on a 10 second inputs. An AUC of 0.977 was reached for this binary classification model with a 88,7% sensitivity [45].

3.2 Abnormal event detection in ECG

The search for the perfect algorithm for event detection in ECG has been ongoing for many years and, consequently, many approaches have been developed to tackle this challenge. This area of research is mostly motivated by the difficulty that lies on diagnosing cardiac pathologies by visualising an abundance of information and sequential mechanisms regarding ECG records. Parallel to the increase of performance of machine learning algorithms, scientists have considerations in terms of computational load, data requirements, classification time and the possibility to integrate in a device to use during daily activities. Martis and his colleagues (2013) investigated and compared various machine learning techniques for arrhythmia classification, integrating Independent Component Analysis (ICA) with Classification and Regression Trees (CART), Random Forest (RF), K-nearest Neighbour (KNN) and ANN. His study revealed good results for each method, being KNN with most accuracy. The ANN used in this study had a feed forward structure composed by 3 layers [46].

Wang et al. (2001), was concerned with minimizing the continuous and high computation load of arrhythmia detection, proposed a short-time multifractal perspective using a 2 layer fuzzy kohenen network in order to provide more computational flexibility without detriment to the accuracy of the predictions. Although the promising results, classification was limited by only 4 classes: normal, ventricular fibrillation, ventricular tachycardia and atrial fibrillation [47].

An implementation of an arrhythmia detection algorithm was also proposed for mobile communication in the hospital context by Sufi and Khalil using data mining techniques. By applying attribute selection and expectation maximization based clustering methods for a rule based classifier, compressed ECG signals were used in order to save computer memory and to avoid overloading network traffic, without the need of decompression, avoiding prediction delays [48].

More recently, Acharya et al. (2017) took the deep learning approach by using a CNN composed of 11 layers, analysing segments of ECG with either 2 or 5 seconds, instead of the common technique of using only one ECG beat. Using a CNN revealed to be successful for its ability to classify between normal, atrial fibrillation, atrial flutter and ventricular fibrillation, without needing QRS detection or feature extraction. However the lack of examples for signals with ventricular fibrillation condemned more than a third of this cases to be wrongly classified. Accuracy, sensitivity and specificity values of 92.50%, 98,09% and 93,13%, respectively, were achieved for two seconds ECG segments. As for five second segments, this study was able to achieve an accuracy of 94,90%, a sensitivity of 99.13% and a specificity of 81.44% [35].

Savalia and Emamian (2018) proposed a deep learning algorithm that consisted in a Multi Layer Percepton (MLP), containing 4 hidden layers, and a CNN with 4 layers. The MLP predicts between a normal or abnormal sinus rhythm, while the CNN predicted between 9 different classes, NSR, second degree AV block, first degree AV block, atrial flutter, atrial fibrillation, malignant ventricular, ventricular tachycardia and ventricular bigeminy. The test results showed high accuracy of 88.7% for MLP and 83.5% for CNN, even though some confusion between atrioventricular block and ventricular bigeminy was detected, due to similar features [27].

Also, a combination of CNN with RNN was developed by Andersen et al. (2018) in order to distinguish between atrial fibrillation and normal sinus rhythm in segments of Holter ECG recordings of 24 hours. By examining RR intervals, this algorithm used a CNN to extract features directly from the data in order to be processed by a RNN. This combination allowed to achieve a sensitivity of 98.98% and a specificity of 96.95%, proving itself to be efficient not only in its classification but also in time, producing predictions of 24 hours of ECG signal in under one second. Despite being a promising deep learning approach, it was noted that noise present in some segments affected the classification, so the detection of noise corrupted segments should be implemented in order to discard those segments [49].

Recently, Hannun and colleagues (2019) developed a 34 layer DNN that could detect between 10 different arrhythmias and NSR, while also detecting noise corrupted segments, constituting 12 prediction classes in total. Without needing any preprocessing, raw data from single-led ECG signals is introduced, producing outputs that scored higher than average cardiologists in terms of sensitivity and matching specificity values. The average AUC was of 0.97 and by comparing confusion matrices between the DNN and a selected number of cardiologists' predictions, similar results were found, accentuating the same problematic rhythm classes for both. This reveals that most misclassifications made by the classifier also occur in manual reviews by specialized physicians, which factors like lack of context, limited signal duration and the single led method may justify. These misclassifications were most common between supraventricular tachycardia and atrial fibrillation, sinus and junctional rhythms, and between ectopic atrial and sinus rhythms. Overall, this algorithm's accuracy performed better than the cardiologist gold standard, except for ventricular tachycardia (despite higher levels of sensitivity) and is yet to be applied in 12-lead ECG, being that one possible difficulty would be to tailor the DNN to the target applications [40].

Chapter 4

Methods

4.1 Context

The current research on DNN applied to biosignals of the LIBPhys (Laboratory for Instrumentation, Biomedical Engineering and Radiation Physics) laboratory inspired this thesis. The development of a network that learns from a biosignal morphology and syntheses the learned sequence was tested on ECG, EMG and respiratory signals [7]. In the context of this thesis, only ECG signals will be used.

The proposed DNN, portrayed in Figure 4.1, uses a RNN, more specifically, Gated Recurrent Units (GRU) architecture, that provides efficient memory management, preventing the curse of the vanishing gradient, using reset and update gates and a faster conversion without sacrificing the model's accuracy[7].

Firstly, a preprocessing step is carried out in order to reduce the noise present in the signal, followed by a quantization and signal segmentation to generate the vector x, composed by integers, that is fed to the network. The reduction of the signal dimensions is important to increase the computational performance. Then, each sample (x_n) represents a column in the embedded in matrix E, that gives a vectorial representation of the input value ($\hat{x}_n$), inside the network. This matrix adjusts its elements from random numbers in the training phase[7].

That vectorial representation passes through a sequence of three GRU layers together with the last state of the corresponding GRU layer, which contains gates responsible for choosing the information that passes through to the next layer and state. When the new information is deemed irrelevant, reset gates are activated in order to filter it, while update gates hold the previous state information or update to the state to a new value. Together, they decide what information should be passed to the output[7].

Finally, the outputs produced by the last GRU layer is computed by a regression node,

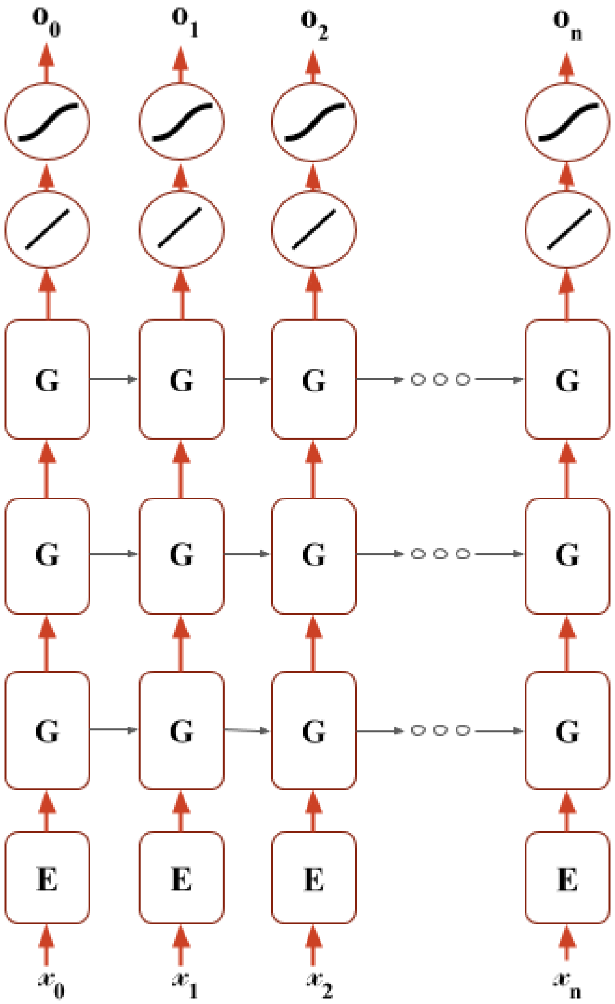


Figure 4.1: GRU architecture.

also named as fully connected layer, that translates the information through the vector $\hat{o}$, which is transformed into a probability density vector by a softmax function, resulting in the output vector o. This vector gives the probability of the next sample - x_{n+1} - of being of the value k, the integer representation of the signal[7].

The training process occurs by feeding the network with signal windows as input to predict outputs, in this case the signal window dephased by one sample, in order to learn parameters, initialized randomly, while reducing the error, which is calculated by a loss function, which compares the amount of misclassifications with the classes associated though the mean squared error. The equation that defines the loss function is given bellow:

$$\frac{1}{N}\sum_{N}^{n}(o_n - \hat{o}_n)^2 \tag{4.1}$$

The minimum loss is obtained by an optimizer that uses the Backpropagation Through Time (BPTT) method, RMSProp. Each one of the models was trained using an individual's biosignal, creating networks that assessed biometry by comparing the source morphology with the model that trained it[7].

Regarding the ECG generator, difficulties were found in training with a high number of steps, resulting in the training the algorithm with lower resolutions in the first stages. However, the typical morphology was visible in the synthesized signals. Besides, the model learned the signal's frequency and wave characteristics, including local minima and maxima, being able to make adjustments when errors were made[7].

Through ECG biometry, it was noted that the model recognized with high accuracy the individual's signal used for the training phase, by inspecting the mean error and standard deviation[7]. Parallel to this method, this thesis pretends to use this finding in order to find abnormal segments inside the individual's normal signal. These abnormal segments could then refer to noise affected samples or pathological events.

4.2 Technological Materials

The materials that will be used in this thesis are the following:

- Biosignalsplux [50]: wireless device that will be used to acquire ECG signals;
- Open Signals [51]: software designed by PLUX that allows visualization and storage of the ECG signals in the computer;
- PyCharm [52]: code editor made specifically for Python language that will be used to process the biosignals;
- TensorFlow™ [53]: an open source software library used for creating the DNN.
- SciPy [54]: an open source python library used for signal processing and optimization.

4.3 Architectures

In order to perform ECG classification according to the two tasks at hand, two different architectures were chosen and optimized to classify noise affected segments and to classify different types of arrhythmia and NSR. The model chosen for each task is a key part of the process, being that both algorithms were implemented using an autoencoder, most importantly, the encoding aspect of this model. In this chapter, the autoencoder that is implemented as well as the models used to perform the classifications will be explained in detail.

The approach used for this thesis is of learning the characteristics of a normal ECG signal that doesn't contain any noise or pathological events, using the autoencoder. Then, given the correct structure of a healthy and noise free ECG signal, when presented with ECG signals containing different types of noise or pathological events, the algorithm learns the differences in the signal through the produced features, in order to identify ECG signals that are not normal.

The signal data is sorted in windows of 64 samples, for each classification, while the training of each algorithm is performed in batches of 256 shuffled windows, in order to increase overall generalization. According to the data used, each window contains roughly a cycle of ECG signal.

4.3.1 Autoencoder

Deep neural network autoencoders are able to encode the input data, as well as reconstruct it at the output layer with the goal of reproducing the input signal as similar as possible. This is achievable by training the neural network to transform the input data while making sure the important information isn't lost in the encoding process. This process allows the machine to produce a feature extraction mechanism by finding the most relevant parameters in the ECG signal. Furthermore, the latent layer, which is the vector that contains the produced features by the encoder, can be read by the decoder that reconstructs the signal with the minimum loss possible.

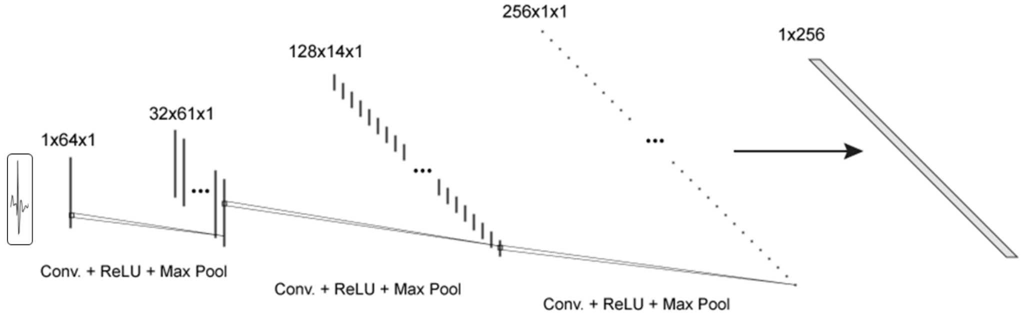


Figure 4.2: Encoder.

The autoencoder developed for this dissertation is composed by an encoder, a latent layer and a decoder. The encoder (Figure 4.2) comprises three convolutional layers, followed by Rectified Linear Unit (ReLU) activation and max pooling, resulting in the latent layer of size 256. Activation functions such as the ReLU function, are used to map the resulting values of a layer in between a certain range of values. This type of non-linear activation function, given by $R(z) = max(0,z)$, is currently the most used in CNN, depicted in Figure 4.3. This function only lets the positive values to pass, while the negative transform to 0.

The convolutions performed in this network are made between the input vector that contains the raw ECG signal and the kernels, which are moving vectors with weights with a fixed size. In order to obtain a latent layer that holds information and characteristics of the ECG signal, while also obtaining its hierarchical relationships, each kernel was set with the size of 4 and the number of kernels in each convolution increases from 32,

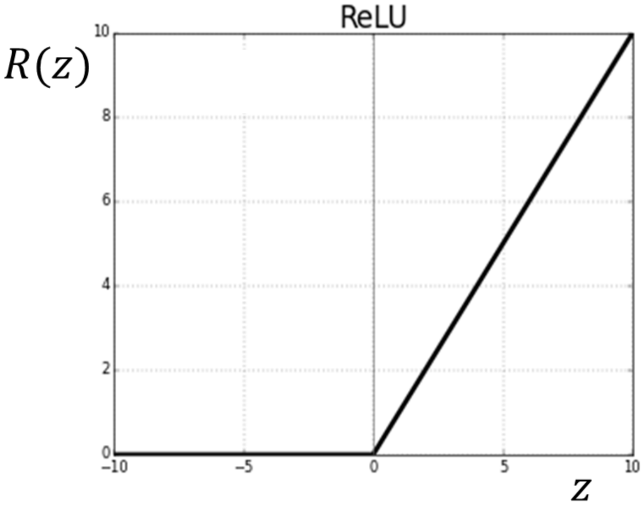


Figure 4.3: ReLU activation function.

to 128, and ending with 256 in its last layer. Furthermore, the stride number in which the kernels step over the input samples also increases in each layer, starting with 1 and increasing to 3.

In order to reconstruct the input signal, the latent layer is then fed to the decoder, containing five layers in total. The first four layers are convolutional layers, followed by ReLU activation. In order to decode the values in the latent layer, transposed convolutions were performed using kernels of size 4, with its number decreasing in each convolutional layer, at the same rate of the encoder, while also using only one kernel in the fourth layer. The stride value was 2, except in the second and fourth layer, which were 1 and 4, respectively.

The last layer produces the output signal, obtained by batch normalization, followed by an activation function, resulting in a tensor of size 64, being that this last layer has the same number of neurons of the autoencoder's input signal. The activation function used is the hyperbolic tangent, which maps the results to a range of values between -1 and 1, according to the following graph (Figure 4.4):

Even though the decoder won't be used specifically for the classification of abnormal events in ECG, this component is vital for the training phase of the autoencoder as it learns the features during this process. The minimum loss value between the input signal and the reconstructed signal is then obtained using RMSProp, with an initial learning rate of 0.001. This loss value, that corresponds to the autoencoder's error, is calculated through the mean squared error function:

$$MSE(y_i, \hat{y}_i) = \frac{1}{N}\sum_{i=1}^{N}(y_i, \hat{y}_i)^2 \tag{4.2}$$

where $\hat{y}_i$ and y_i represent the predicted and real values, respectively, whereas N represents the total number of samples.

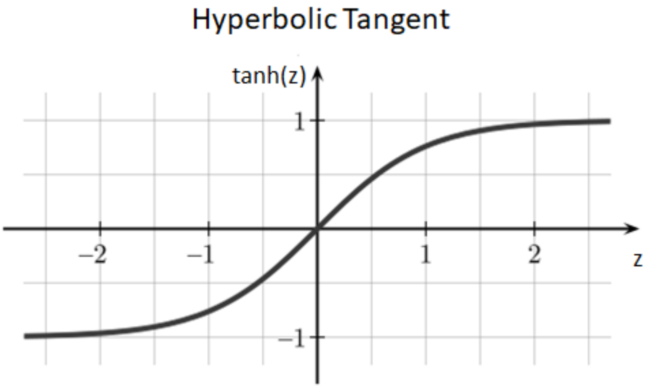


Figure 4.4: Hyperbolic tangent activation function.

4.3.2 Noise detection neural network

As stated before, the noise classification is not performed directly from the raw ECG signal, but from the feature tensor created by the encoder, being that the input signal in this model is the coded signal.

For this task, two algorithms were developed according to the type of classification they performed. The first one classified the signal into four different classes: Normal signal (NS), BW, MA and Electrode motion (EM) noise. The second algorithm aimed to detect two classes: NS and Noise affected signal (NAS). Both algorithms follow a similar structure, in terms of both having several dense layers following the encoder, as described in Figure 4.5.

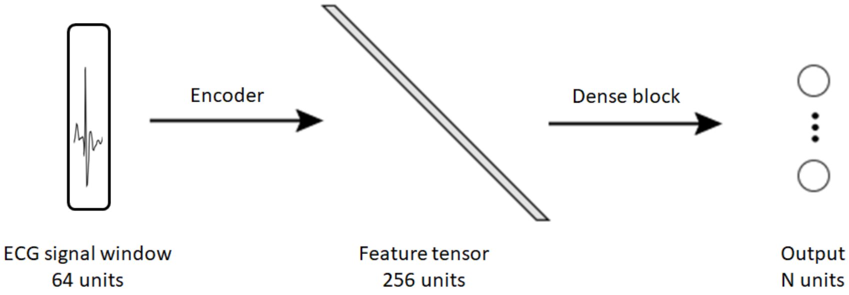


Figure 4.5: Model architecture for the noise detection, with N classes.

The first algorithm contains 8 layers whereas the second contains 9 layers. For each dense layer in both models, there is a given number of neurons, that decreases by half in the following layer, starting with 256 units in the first layer and the number of classes in the last layer, in both algorithms. Also, each dense layer is followed by a ReLU activation function, except the last layer, which is followed by a softmax activation function. The latter is a form of logistic regression that transforms the input values by normalizing them into a vector of values that follows a probability distribution function, whose total sums up to 1. These values range between 0 and 1, according to the probability of belonging in a given class. Since the last layer contains N neurons, the one with the highest value will determine in which class the window segment belongs.

The softmax activation function can be described in Figure 4.6, and is given by:

$$S(z) = \frac{e^{z_i}}{\sum_j^K e^{z_j}} \tag{4.3}$$

for $i = 1, \ldots , K$ and $z = (z_1, \ldots, z_K)$, where K is the number of neurons in the layer.

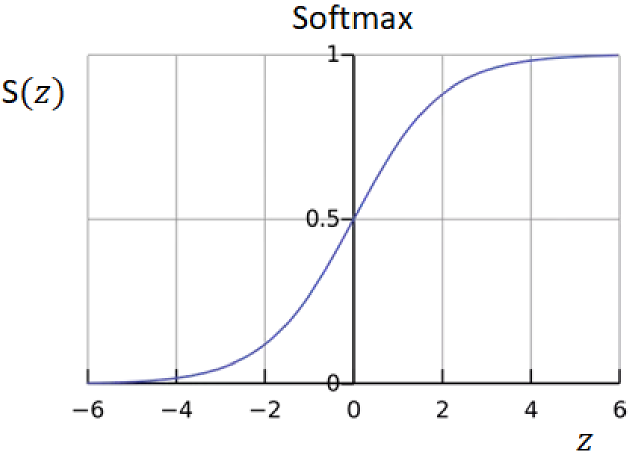


Figure 4.6: Softmax activation function.

Finally, in both models, a dropout layer is introduced between the fourth and fifth layer, in order to prevent data overfitting, which randomly disconnects 50% of the neurons in itself. The dense block for the 2-class model is shown in Figure 4.7, describing the different types of layers and its number of neurons (units), as well as the activation functions used.

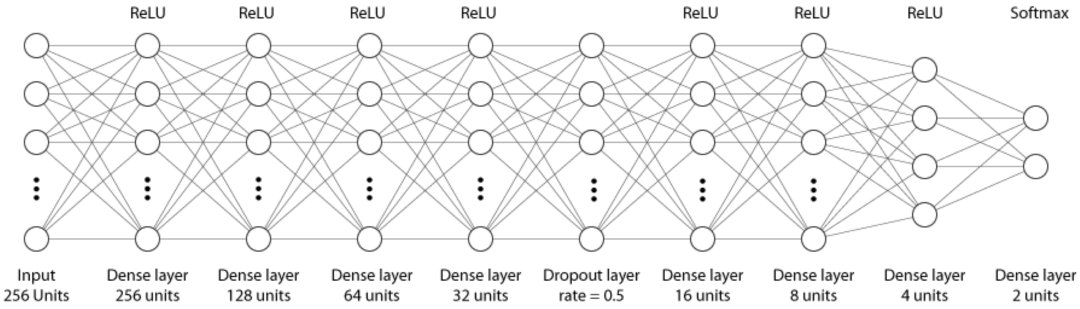


Figure 4.7: Dense block used for the 2-class noise detection model.

Regarding the training of both neural networks, RMSProp optimizer was also used to obtain the minimum loss between the predicted class and the labels associated with the ECG signals. The cost function utilized to obtain the loss values is calculated using the cross entropy method:

$$H(y_i, \hat{y}_i) = -\sum_{i=1}^{C} y_i log(\hat{y}_i) \tag{4.4}$$

4.3.3 Arrhythmia detection neural network

The proposed model for arrhythmia detection, portrayed in Figure 4.8, resorts to the use of an encoder, dense layers and a RNN, with GRU architecture, forming a sufficiently deep neural network.

After the ECG signal is preprocessed, each signal window then generates a vector of with dimension $F_D = 256$ (feature dimension), which integrates the feature tensor N signal windows of 64 samples. This tensor is then transformed by a dense layer, with no activation function, generating a tensor with dimension $H_D = 128$ (hidden dimension), in which each row represents a signal window.

Then, sequences of 32 rows of the previous tensor are fed into a GRU net, which contains three GRU layers, that pass on the data sequentially together with the last state of the previous layer. The output of the last GRU layer is then carried into the dense block that ultimately generates the the classification vector with size $C_D = 7$ (classification dimension), which is the number of classes in this neural network.

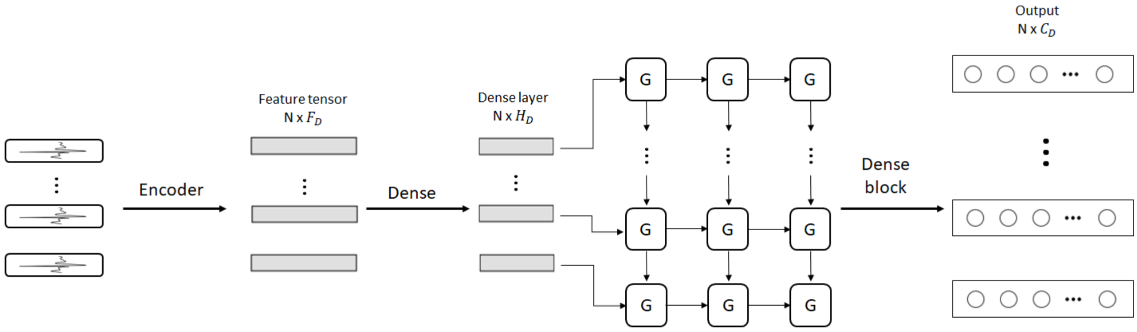


Figure 4.8: Model architecture for the arrhythmia detection, with $C_D = 7$ classes.

The dense block of this architecture, as seen in Figure 4.9, starts of with a batch normalization layer, which is implemented to speed up the training phase by adjusting and scaling the values generated til this point. This layer normalizes the output of the previous GRU layer by subtracting the batch mean and dividing by the batch standard deviation. Following this, several dense layers are applied, by decreasing the vectors dimension from $H_D = 128$, to the classification dimension, $C_D = 7$. Each dense layer decreases its number of neurons by half, starting with 128 units to 7. With the exception of the final layer, which implements the softmax activation function, every dense layer is followed by a ReLU activation function.

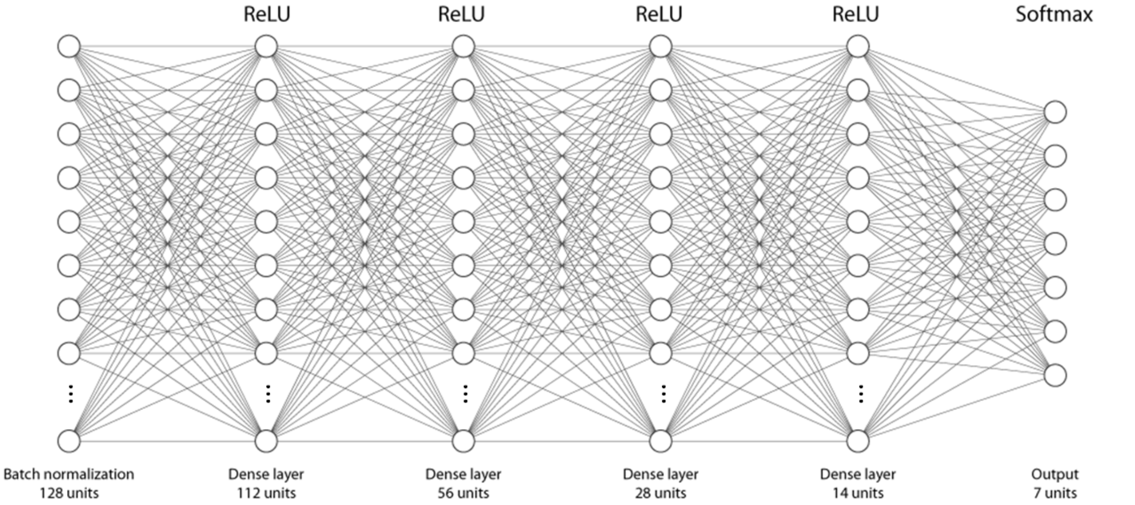


Figure 4.9: Dense block used for the arrhythmia detection model.

4.4 Data

It is important to choose the right data for each model that suits the classification task at hand, being that the presence or absence of noise or pathological events in the ECG signal is crucial for the algorithms to learn important parameters. That said, different datasets were used for the autoencoder, noise and arrhythmia detection models.

4.4.1 Autoencoder

In order to reconstruct an ECG signal, it is important to choose a dataset that contains clean signals, with good signal-to-noise ratio and without other abnormal morphologies such as included in arrhythmic events. Finally, the signal morphology must be easily recognizable by a human, even with limited cardiological knowledge background. These were the principles followed in the LIBPhys laboratory, regarding biosignals learning and synthesis, as explained by Belo et al. [7].

The Fantasia Database is a extensively used dataset due to the described requirements and therefore it was used for training and testing algorithm. It contains acquisitions of ECG signals from twenty individuals from ages 21 to 34 years old and twenty individuals from ages 68 to 85 years old, with equal gender distribution. Each signal was recorded during 120 minutes with a sampling frequency of 250 Hz [55]. The data is also subsampled by a factor of 4, in order to use less amount of samples, while mitigating the corruption of the signal's characteristics, due to computational limitations.

In order to obtain a higher level of generalization, 25% of each signal was used for the training phase, containing an equal distribution of each individual available in the database, and with equal number of data belonging to men and women, from both age groups.

4.4.2 Noise detection neural network

For this experiment, it was important to gather clean signals with the same characteristics described above, while also gathering signals with clear indication of specific types of noise. For this reason, signals from the Fantasia Database were also used, both for the clean signals pertaining to normal ECG signal and noise affected segments, constructed by adding different types of noise to said signals.

Signals containing noise were assembled by adding noise from the MIT-BIH Noise Stress Database, containing baseline wander, muscle artifact and electrode motion raw noise [56]. This method was used in order to label in a controlled way, while also obtaining a wider variety of noise corrupted signals, by adding said signals to the clean signals from the Fantasia Database and in order to guarantee the same sampling frequency.

For the training phase, equal amounts of each type of noise were used, while also using the same amount of data of clean ECG signals with NSR. Thus, the overall training data had the same percentage of each class.

4.4.3 Arrhythmia detection neural network

In this task, it is essential to gather data that contains different types of arrhythmias, as well as segments of ECG signal that don't contain pathological events (NSR), while also being obtained from individuals with pathologies.

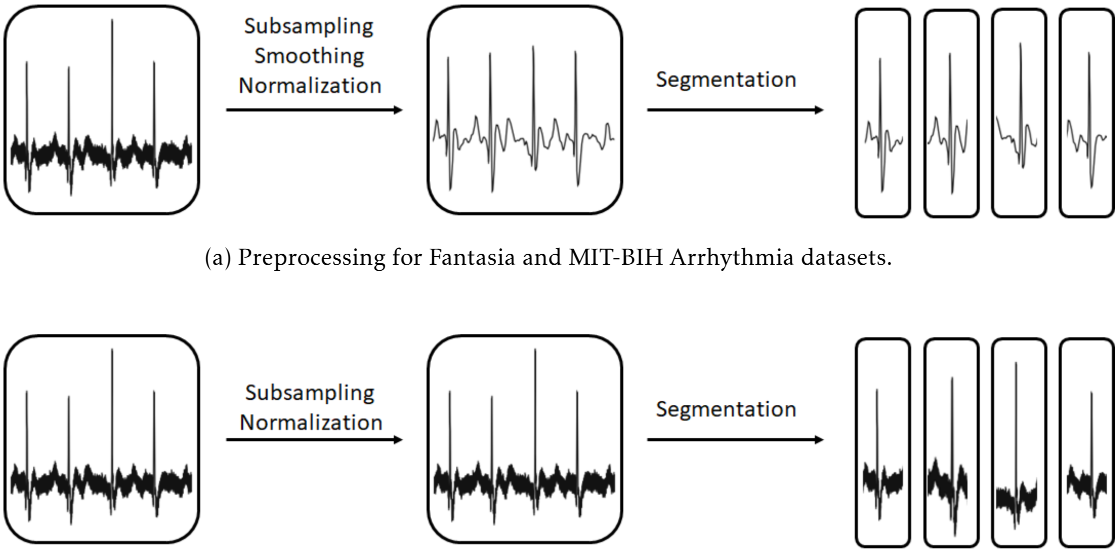
The database used for this experiment is the MIT-BIH Arrhythmia Database, which contains ECG 48 half hour recordings from 47 different subjects, with a sampling frequency of 360 Hz. In this case, the ages and gender of the subjects are unknown. This database contains 14 different types of arrhythmia: Atrial Bigeminy, Atrial Fibrillation, Atrial Flutter, Ventricular Bigeminy, Second-degree Heart Block, Idioventricular Rhythm, Nodal A-V junctional Rhythm, Paced Rhythm, Wolff-Parkinson-White Syndrome (Pre-excitation), Sinus Bradycardia, Supraventricular Tachyarrhythmia, Ventricular Trigeminy, Ventricular Flutter and Ventricular Tachycardia [57].

By investigating the occurrences of each arrhythmia in this database, the annotations provided were analysed and counted, in order to select which types of arrhythmia could be considered for training, given that for a small sample of events regarding to particular pathology, this amount wouldn't be enough to learn its key parameters.

In order to obtain at least 500 signal windows for each arrhythmia, 7 classes were defined for the algorithm: NSR, Paced Rhythm (P), Ventricular Bigeminy (B), Sinus Bradycardia (SBR), AFIB, AFL and Wolff-Parkinson-White Syndrome (Pre-excitation (PREX)).

4.5 Signal Preprocessing

Before the signals are fed to the deep learning algorithms, preprocessing steps are implemented in order to transform the raw data. These are imperative for the models to learn the underlying patterns, to provide better generalization and overall quality. In this



(a) Preprocessing for Fantasia and MIT-BIH Arrhythmia datasets.

(b) Preprocessing for noise corrupted datasets.

Figure 4.10: Preprocessing steps for the different datasets used for the autoencoder, noise detection neural network and arrhythmia detection neural network.

chapter, preprocessing methods applied to the ECG signals in both tasks are explained in detail.

The preprocessing methods, described in Figure 4.10, are:

- Subsampling;
- Smoothing;
- Normalization;
- Segmentation;

4.5.1 Subsampling

Subsampling, or decimation, refers to the process of reducing a signal's sampling rate as a mean to reduce the computational complexity of the deep neural network in order to speed up the training process and save memory. This step of reducing the number of samples is especially necessary to obtain a whole ECG cycle in signal windows of 64 and can be adjusted depending on the signal's sampling rate.

This downsampling process is taken into action using SciPy's decimate function [54]. This technique is especially useful in order to prevent signal aliasing, since it applies a low pass anti-aliasing filter that suppresses all frequencies that may cause aliasing, before the decimation process, which consists in the application of an oder 8 Chebyshev type I filter [58]. This process does not affect the ECG signal since it possesses a low frequency, guaranteeing that the sampling rate after subsampling is higher than the double of the frequency of an ECG signal.

4.5.2 Smoothing

This preprocessing technique is applied to every signal except the data meant to represent noise corrupted ECG signal. This step is important in order to gather signals that are as clear as possible, without the presence of noise, for training purposes.

This method consists on applying a moving average filter that smooths out undesirable signal fluctuations for each window of 64 samples, followed by the application of a Hanning window filter, which is able to reduce the lower frequencies in the signal, as well as removing the higher frequencies, while maintaining the signal's characteristics. The Hanning Window is given by:

$$w(n) = \sin(\frac{\Pi n}{N}) \tag{4.5}$$

where $N + 1$ stands for the window size. The window size selected for the smoothing preprocessing step is 64.

4.5.3 Normalization

Normalization is a common step used in machine learning for data uniformization. It consists on altering the values of the signal in order to fit a determined scale, while maintaining the signal characteristics.

This technique allows the neural networks to learn, while being able to perform better generalizations. In order to facilitate the training process, the inputs are normalized so that the average of each input variable is as close to zero, which allows the variable weights of the first layer to converge faster when they update in each epoch, whereas with an input signal with only positive values, the variable weights can only decrease or increase all together [59].

Each normalization step is performed in windows of 64 samples, by subtracting from the signal its mean value, $\overline{x}$, and dividing this result by the difference between the signal's maximum and minimum values, according to the following function:

$$x' = \frac{x - \overline{x}}{max(x) - min(x)))} \tag{4.6}$$

where the normalized signal is x' and x denotes the raw signal.

4.5.4 Segmentation

This procedure consists on separating the signals in chunks of samples of a fixed sized - time-windows - that are directly fed to the model for classification, with the possibility of window overlapping, that allows information to be carried to the next window. With this method it is possible to label and classify each window. For each algorithm, each window contains 64 samples, in order to represent a single ECG cycle, with all of its known characteristics.

4.6 Training

Several measures were taken into consideration for the training process, with the intention of reaching faster convergence. The chosen optimizer was RMSprop, because it experimentally provided good convergence rates and results [60]. For this optimizer, each update is done according to the following equations:

$$v_t = \rho v_{t-1} + (1 - \rho\ ) * g_t^2 \tag{4.7}$$

$$\Delta w_t = -\frac{\eta}{\sqrt{v_t + \epsilon}} * g_t \tag{4.8}$$

$$w_{t+1} = w_t + \Delta w_t \tag{4.9}$$

where η refers to the initial learning rate, v_t is the exponential average of squares of gradients and g_t is the gradient at time t along the parameter w^j.

For the models proposed, the initial learning rate was 0.001, which decreased in the course of the training phase, according to certain parameters. This adaptation of the learning rate, results in an increasing speed of convergence by decreasing this value by half, according to a set of rules: (1) if the loss value doesn't change for a certain amount of iterations; (2) if the loss value increases fifteen times, and; (3) if the relative loss is in the order of 10^{-4}.

The input data is separated into batches of 256 signal windows, which means that for each batch the internal model parameters are updated in the training process. These batches are also shuffled so that the models could reach better generalizations while avoiding overfitting.

Chapter 5

Results

In this chapter, the results obtained from all algorithms will be presented and discussed. All models were developed, trained and tested using Tensorflow a platform developed to build and deploy machine learning models. This open source library provides multiple levels of abstraction to build and train models by using the high-level Keras API that allows developers to create large-scale NN with many layers [53]. The training and testing of the neural networks were performed using NVIDIA GeForce GTX 960 graphics cards (Graphics Processing Unit (GPU)), as also the latter was performed using Intel Core i5 processor (Central Processing Unit (CPU)), in order to determine if the algorithms could be used without the need of a GPU.

5.1 Autoencoder

Upon completing 2082 epochs in the training phase, a minimum error of 0.0001 for the training dataset was reached. Figure 5.1 shows the progress of the error over the epochs, which shows that the loss value was minimized from the initial value of 0.0165 to 0.0001. This error is calculated using the Mean Squared Error function, as explained in chapter 4. Each epoch performed variable optimization for each batch of data, containing 256 windows, by calculating the prediction error for each batch, and later performing variable optimization considering the entire training dataset.

For the testing phase, a portion of every signal available in the Fantasia Database was used, while being careful not to overlap with the training data. The MSE was calculated for each signal and the mean value is of 0.0026, with a standard deviation of 0.0012.

Furthermore, this error was calculated for each window for further calculation of each signal's average error and standard deviation. In Figure 5.2, these values are presented for the first 10 individuals with ages between 21 and 34 years old, and also for the first

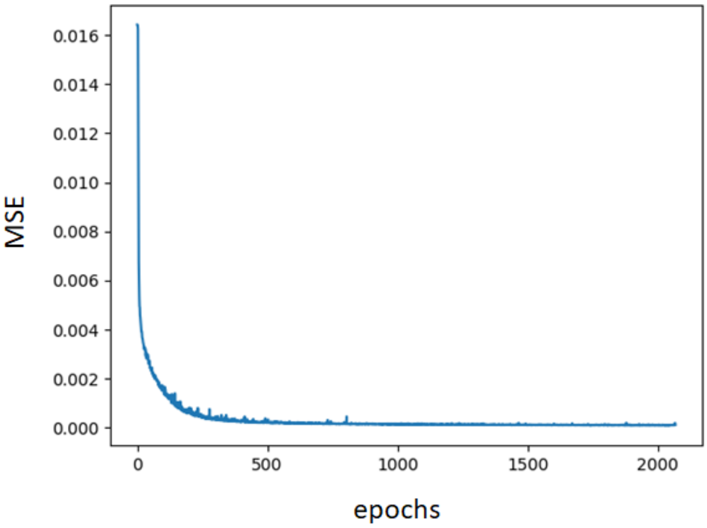


Figure 5.1: Curve that describes evolution of the MSE over the training phase.

10 individuals with ages between 68 and 81 years old.

Since the autoencoder was trained with a variety of subjects from the Fantasia Database, the model was able to reach a considerable low error value across all signals used for testing, with the highest error of 0.0050 ± 009 occurring in ECG 30, and the lowest in ECG 9, with 0.0010 ± 002.

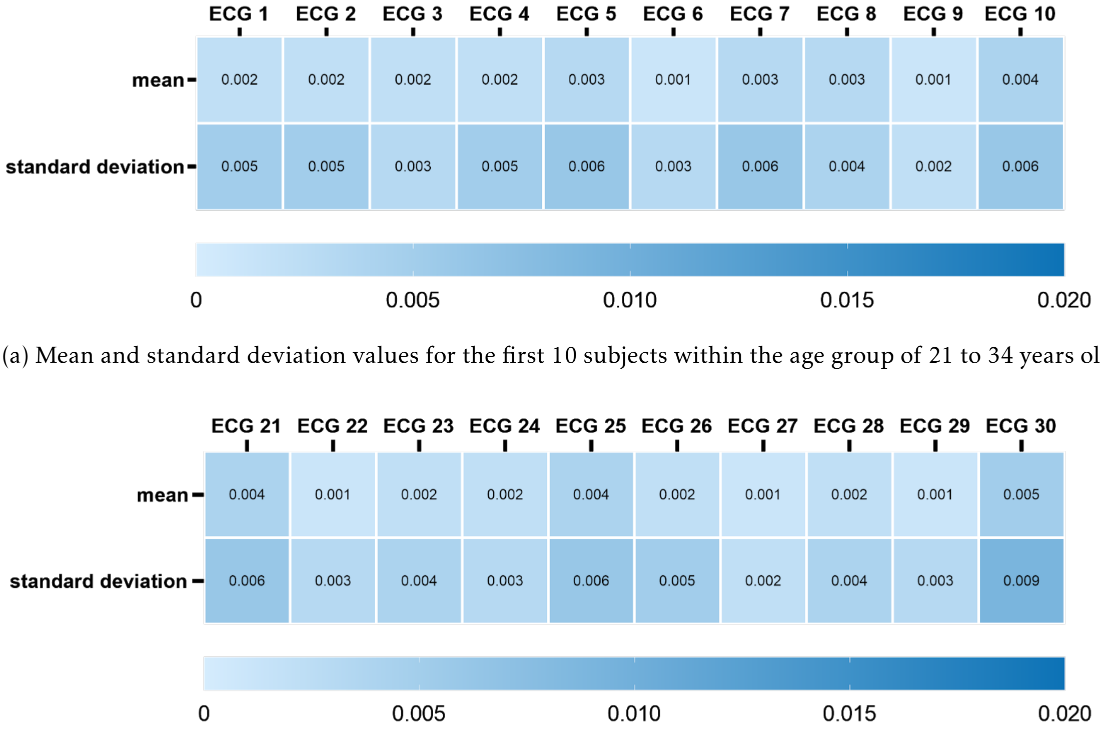
The model was able to learn and reconstruct the main characteristics of the signal, with similar amplitudes. In addition, the algorithm was also able to create a signal with the same frequency, baseline values and fluctuations as the original signal, as can be seen in Figure 5.3 and Figure 5.4, which represent the original and reconstructed signals for ECG 9 and 30, respectively.

Upon analyzing the results, it is considered that the encoder developed for the autoencoder can be applied for the following classification models. Hereupon, the trained autoencoder with its variables will be implemented within the Noise and Arrhythmia neural networks.

5.2 Noise detection neural network

In this task, two algorithms for the detection of noise corrupted segments of ECG were developed, as explained in chapter 4. At first, the binary noise detection model was developed, in order to perform classifications that either determined if the given signal window was contaminated by different types of noise or if the signal didn't present a low Signal-to-Noise Ratio (SNR).

Then, another model was developed in order to detect the source of the noise, by classifying between NS, the presence of BW, of MA or EM.



(a) Mean and standard deviation values for the first 10 subjects within the age group of 21 to 34 years old.

(b) Mean and standard deviation values for the first 10 subjects within the age group of 68 to 81 years old.

Figure 5.2: Mean and standard deviation of the mean squared error, calculated for each subject of the Fantasia Database.

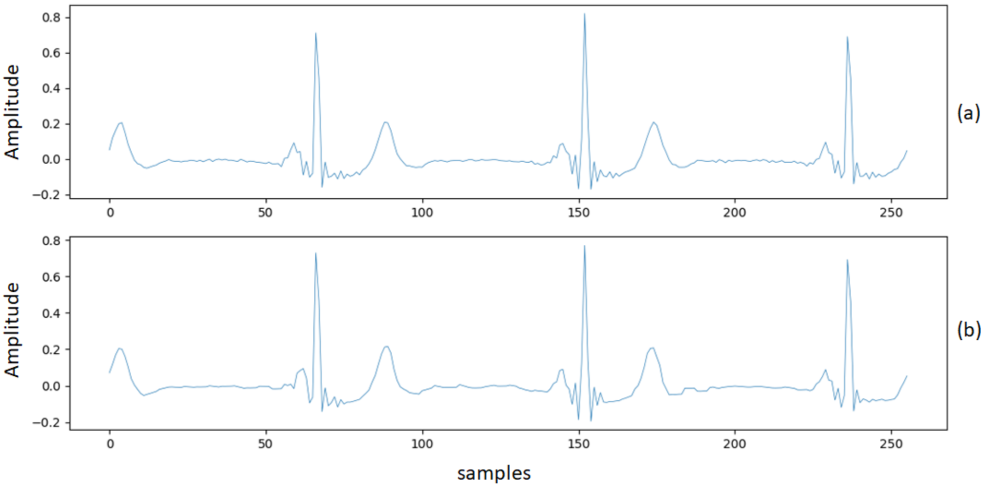


Figure 5.3: Portion of the signal ECG 9, from the Fantasia Database (a) and reconstruction of the same signal (b).

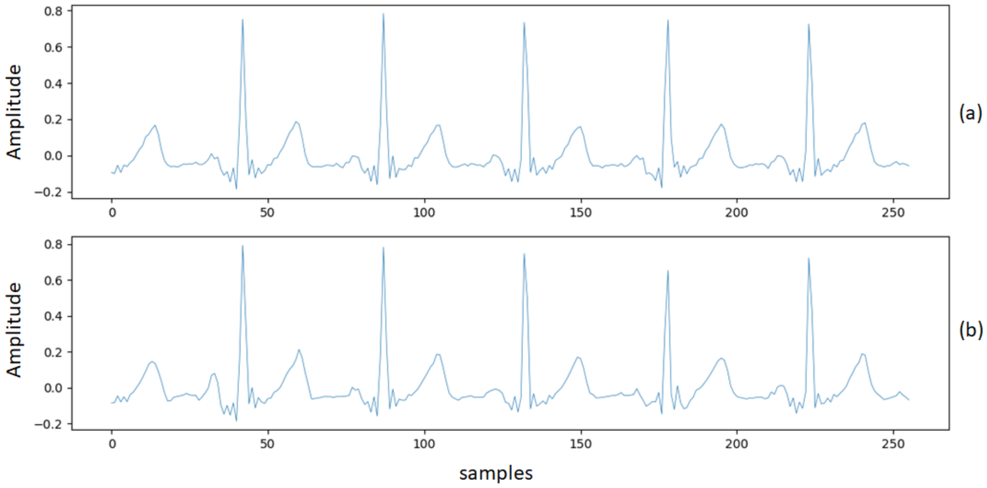


Figure 5.4: Portion of the signal ECG 30, from the Fantasia Database (a) and reconstruction of the same signal (b).

5.2.1 Binary noise detection model

The proposed algorithm classifies ECG segments according to two classes: NS or NAS. For the training phase, it is important to collect equal amounts of data for each class, in order to prevent overfitting, while collecting said data from signals that are from different individuals so that the model is able to provide better generalization.

A portion of 4 different signals from the Fantasia Database were selected in order to construct noise affected signals by contaminating them with BW noise, MA noise and EM noise. This generated a total of 10936 signal windows for each type of noise, totaling in 32808 noise affected segments. Furthermore, a portion of 12 different signals was collected from the same database without the addition of noise, creating 32808 segments of normal signal. This totals 65616 of ECG segments for the training phase.

Even though data selecting measures were implemented, overfitting of the data would occur after 600 epochs of training with an abrupt increase of error, therefore, this was selected as the maximum amount of epochs for the training phase, reaching a minimum cross entropy error of 0.51, reaching a training accuracy of 98,56%.

For the testing phase, a total of 28104 signal windows were selected, which amounts 14052 for each class. This data was collected from the same signals that were used for the training phase, without collecting the same samples.

The accuracy, sensitivity (or TPR) and specificity calculated in the testing phase are presented in Table 5.1. These results prove that the model was able to successfully perform classification of the presence of noise in the ECG. For this model, TP relates to correct identification of NS, being that TN represents the correct identification of noise in the ECG. So, in this case, the sensitivity reflects the DNN's ability to detect normal ECG signal, whereas the specificity evaluates the model's performance on the detection of noise in the signal. Furthermore, Figure 5.5 shows the resulting normalized confusion matrix.

Table 5.1: Binary noise detection model: classification performance (%)

Accuracy	Sensitivity	Specificity
98,18	98,21	98,15

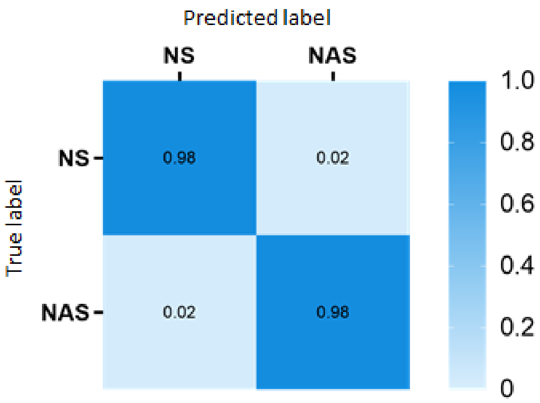


Figure 5.5: Normalized confusion matrix of the dataset used for the training phase, where NS is the positive label and NAS is the negative label.

T-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique that converts high-dimensional Euclidean distances between data samples into conditional probabilities to convey similarities, using a t-Student distribution to compute the similarities between two points in a low-dimensional space [61]. Equal amounts of NS and NAS encoded data was then processed using this technique to visualize the data, as can be seen on Figure 5.6. Each point in the graph represents the tensor created by the encoder, related to a single window signal. It is observed that there is a clear separation between the NS and NAS data points, represented in red and green, respectively. This confirms the results presented before for the binary noise model.

For validation purposes, a signal containing segments of BW, EM and MA noise, intercalated with normal segments was fed into the neural network, producing the classifications seen in Figure 5.7.

An ECG signal was acquired using Open Signals software, containing segments with normal ECG with considerable SNR and segments with low SNR, in order to test if the model could be applied for real life application. Figure 5.8 shows the acquired ECG signal and the classifications for each segment of the ECG, proving to be effective.

5.2.2 Multi-class noise detection model

For this model, ECG segments can be classified according to the following classes: NS, BW, MA and EM. Since there are four different classes, the data selected for the training

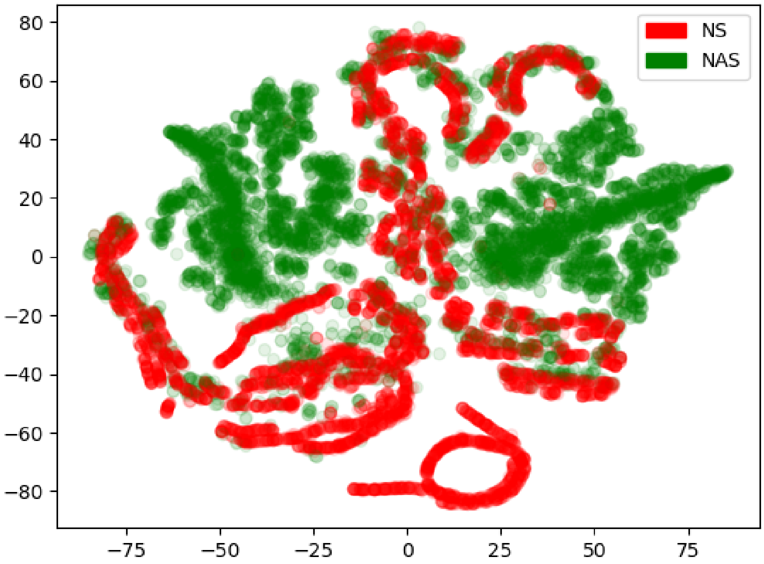


Figure 5.6: t-SNE representation of NS (red) and NAS (green) encoded data windows.

phase contains equal amounts of signals for each category, as to avoid overfitting.

The data was collected from nine different signals of the Fantasia Database, gathering 65616 signal windows in total. For each class, six different signals were used, completing for each one of the categories a total of 16404 signal windows. To guarantee better generalization, each class contained data from different signals, as seen on Table 5.2.

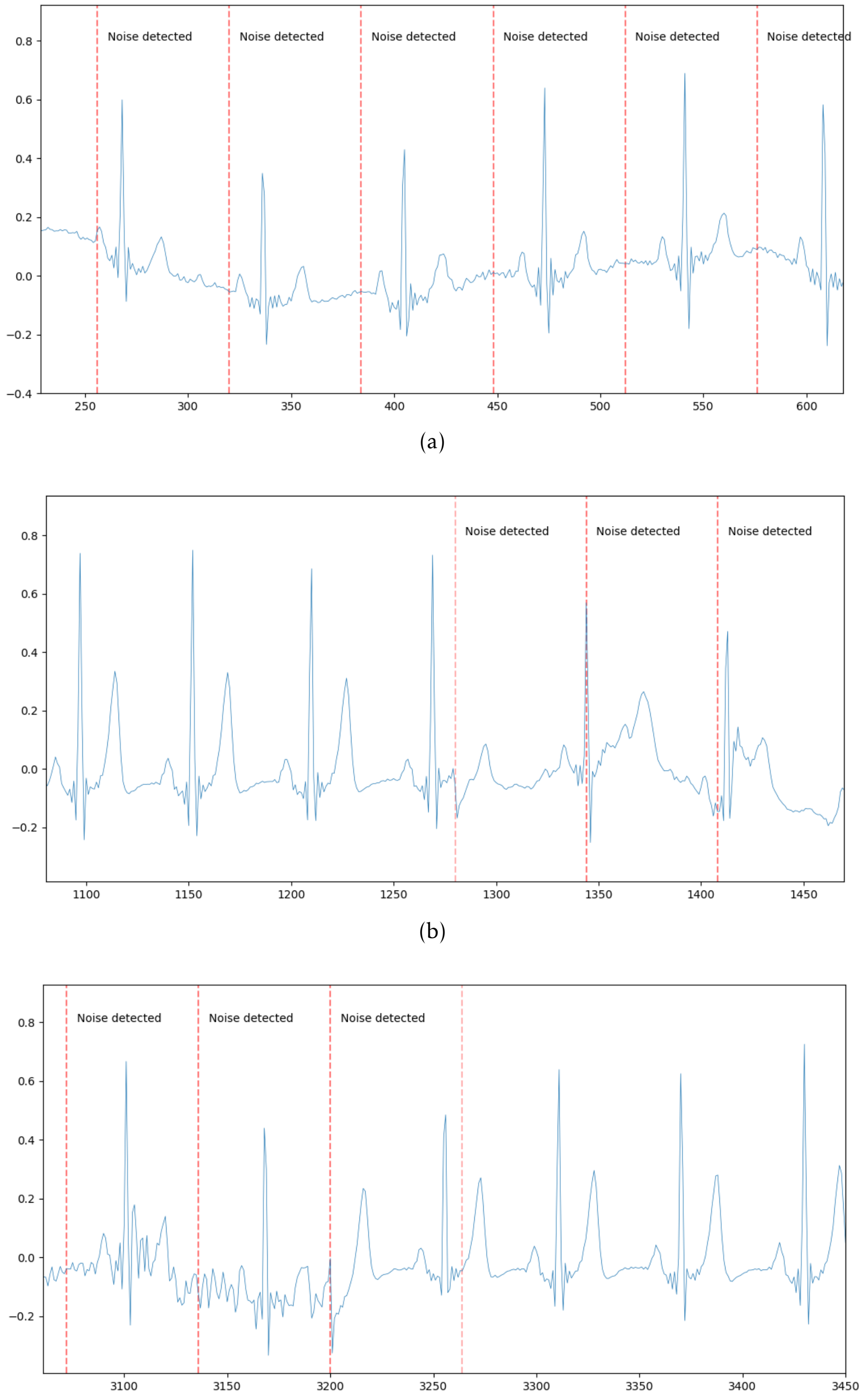
Table 5.2: Composition of the training data.

Class	**Number of windows**	**Signals used**
NS	16404	1 - 6
BW	16404	2 - 7
EM	16404	3 - 8
MA	16404	4 - 9

During the training phase, a minimum cross entropy error of 0.85 was reached after 600 epochs, resulting in a training accuracy of 86,17%. However, testing accuracy was only able to reach 70,74%.

The data selected for the testing phase uses six different signals in total, being that three of them were not used for the training phase, whereas different portions of another three of them were used for the training phase. Even though this model doesn't produce binary classifications, taking a given class as positive, and the other remaining classes as negative, the sensitivity and specificity for each class can be calculated. These performance values relating to each class can be seen in Table 5.3.

By inspecting the normalized confusion matrix in Figure 5.9, it is concluded that even though the algorithm is able to correctly identify normal signal most of the times,



(a)

(b)

(c)

Figure 5.7: Graphical representations of the signal used for validation, with six segments containing BW noise (a), three segments with NS and three windows with MA (b) and three segments containing EM noise, followed by three segments of normal signal (c).

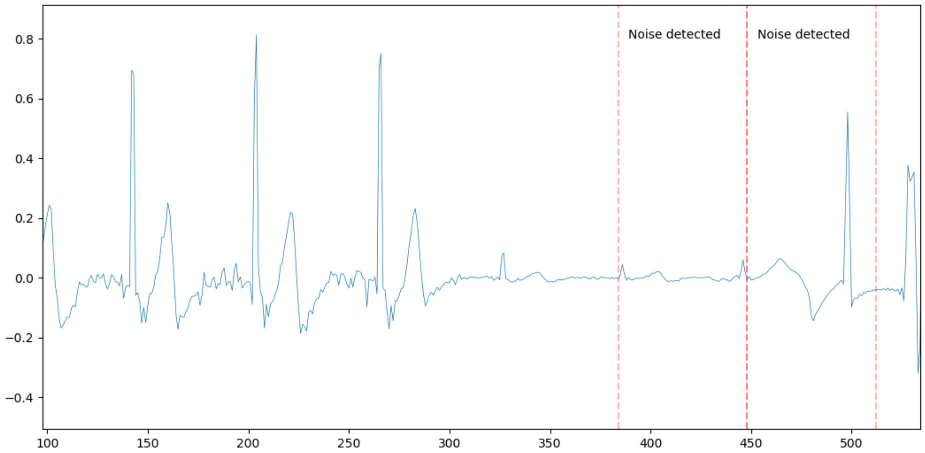


Figure 5.8: Graphical representation and noise detection of the signal acquired using Biosignalsplux wireless device and Open Signals software.

Table 5.3: Multi-class noise detection model: classification performance for each class (%)

Class	Sensitivity	Specificity
NS	89,77	96,63
BW	72,70	84,26
EM	65,19	91,35
MA	59,11	89,47

it is harder to correctly identify which kind of noise is present in an ECG. However, the model was able to learn some noise source characteristics, especially for signals with noise that comes from electrode motion and muscle artifact. Furthermore, the model underperformed in the detection of BW, which could be a consequence of the length chosen for the signal window, since it only presents one ECG cycle, and this type of noise is more noticeable in longer intervals.

The t-SNE representation for the encoded data of the NS, BW, EM and MA signals can be observed in Figure 5.10. The separation between different types of noise, represented by the green, blue and black points for BW, EM and MA, respectively, is not so evident, which confirms the classification errors that occur between these three classes. Despite this finding, the model was able to learn important parameters to differentiate between different types of noise.

Comparing these results with the previous DNN, we come to the conclusion that even though it is possible to detect noise corrupted ECG with high accuracy, differentiating between the several types of noise reveals to be a harder task, given that most of the times signals are affected by different types of noise sources at once.

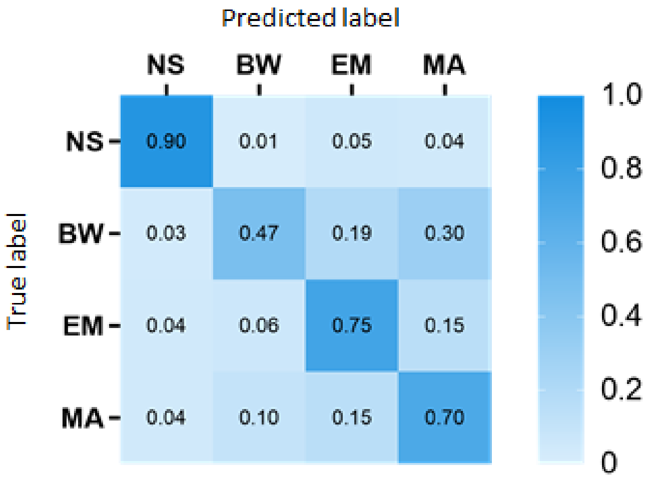


Figure 5.9: Normalized confusion matrix of the dataset used for the testing phase, with classes NS, BW, EM and MA.

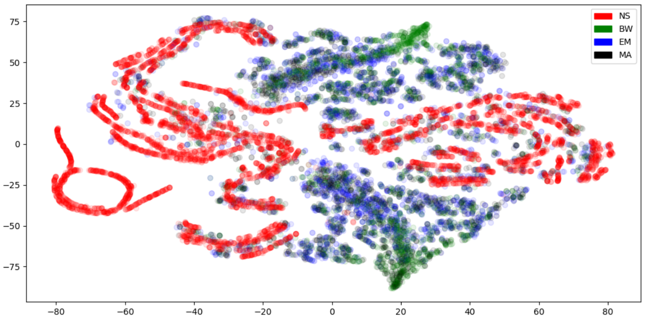


Figure 5.10: t-SNE representation of NS (red), BW (green), EM (blue) and MA (black) encoded data windows.

5.3 Arrhythmia detection model

The data available to train this model is smaller compared to the previous ones, being that the signals chosen for each class must have the same size in order to prevent overfitting, which leaves us with 506 windows for each class.

For the training phase, the segments of ECG signals were selected from the MIT BIH Arrhythmia Database, with a window overlap of 0.5, which generates a total of 806 windows for each arrhythmia class. On the other hand, the NSR class contains double this amount, so that the model could learn different variations of a healthy ECG signal. In total, 6656 signal windows were used for the training process of this DNN.

The minimum cross entropy error of 0.65 was reached after 1750 epochs, ending this training phase with a 90,74% accuracy. However, for the test data, the accuracy was only of 56,85%. This dataset included 103 signal windows from each class.

The normalized confusion matrix, calculated using the test dataset shows that the model was able to learn certain parameters about all arrhythmias, yet underperforming in some of the classifications. As we can observe in Figure 5.11, the DNN performed very well on the detection of paced rhythm (P), and performing very poorly on the detection of Atrial Fibrillation (AFIB), often confusing this class with Atrial Flutter (AFL). The misclassification between these two classes occurs due to the similar morphological characteristics, mainly in the baseline fluctuations. In order to accentuate the differences between AFIB and AFL, preprocessing techniques that suppress the different manifestations of these two arrhythmias should be reviewed.

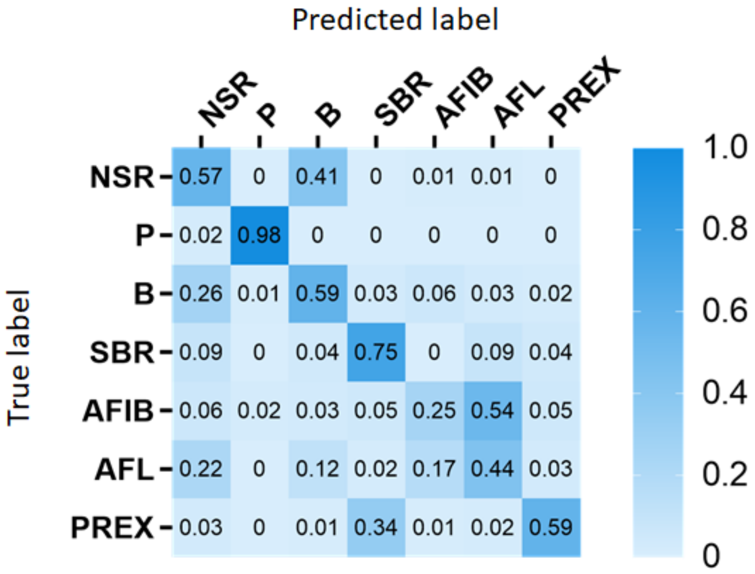


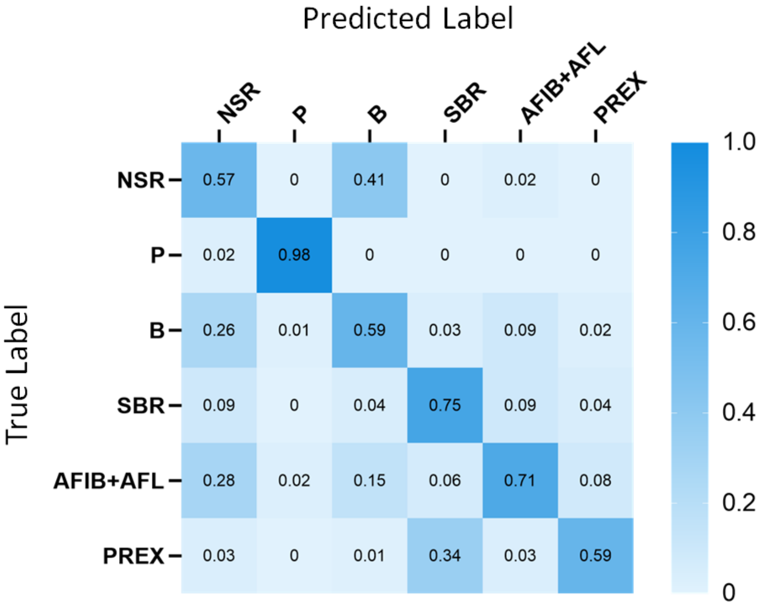
Figure 5.11: Normalized confusion matrix of the dataset used for the testing phase, with classes NSR, P, B, SBR, AFIB, AFL and PREX.

The application of this model in real life context is not possible yet, since the model often considers pathological ECG segments as NSR and vice versa. By analysing the classification performance of each class in Table 5.4, one can conclude that the algorithm often struggles to correctly identify ECG segments with Ventricular Bigemini (B), Atrial Flutter (AFL) and Atrial Fibrillation (AFIB). The average sensitivity and specificity for this model is of 61.13% and 93.10%, respectively.

Table 5.4: Arrhythmia detection model: classification performance for each class (%)

Class	Sensitivity	Specificity
NSR	63,35	85,49
P	97,12	99,73
B	36,31	93,67
SBR	62,60	96,33
AFIB	49,06	90,12
AFL	38,14	91,88
PREX	81,33	94,45

Hannun et al (2019) were able to reach an average sensitivity of 75.22% for a 12 class model, compared to the average sensitivity of 61.13% for the model proposed in this thesis. However, is it important to note that in the mentioned study, AFIB and AFL classes were combined into one unique class [40]. To obtain a better comparison, AFIB and AFL were merged, resulting in the confusion matrix represented in Figure 5.12, which shows better results compared to the confusion matrix represented in Figure 5.11. Also, for this new class, sensitivity values increased to 84,80% (Table 5.5, which in turn decreased the average sensitivity of the model to 70,91%, a value that only differs from the study mentioned by 4,31%.



	NSR	P	B	SBR	AFIB+AFL	PREX
NSR	0.57	0	0.41	0	0.02	0
P	0.02	0.98	0	0	0	0
B	0.26	0.01	0.59	0.03	0.09	0.02
SBR	0.09	0	0.04	0.75	0.09	0.04
AFIB+AFL	0.28	0.02	0.15	0.06	0.71	0.08
PREX	0.03	0	0.01	0.34	0.03	0.59

Figure 5.12: Normalized confusion matrix of the dataset used for the testing phase, with classes NSR, P, B, SBR, AFIB+AFL and PREX.

Table 5.5: Arrhythmia detection model with classes AFIB and AFL merged: classification performance for each class (%)

Class	Sensitivity	Specificity
NSR	63,35	85,49
P	97,12	99,73
B	36,31	93,67
SBR	62,60	96,33
AFIB+AFL	84,80	90,91
PREX	81,33	94,45

Chapter 6

Conclusions

6.1 Conclusions

The principal objective of this dissertation was to develop a framework that detects alterations in the normal pattern of the ECG signal, by creating a model that learns the key characteristics of a normal ECG cycle. This model is then used to create classification algorithms that detect these differences and categorize them in terms of presence of noise and the presence of pathological events.

An autoencoder was developed in order to implement in the architectures for noise and arrhythmia detection, more particularly its encoder, which extracts machine learned features. This algorithm was able to successfully reproduce ECG signals, by replicating the morphology of the signals in the test dataset, which shows the autoencoder was able to learn the key characteristics of a normal ECG cycle.

The task of classifying the ECG signal according to the presence of different types of noise such as BW, MA, EM, was completed using two approaches: a binary DNN, which detected if there was noise present in the signal, and a multi-class DNN that was able to not only detect the presence of noise, but also to detect the source of the noise, which could be from the movement of the thoracic cage muscles while breathing (BW), the activation of muscles in the body (MA) or from the faulty placement of the electrodes (EM).

The first binary algorithm reached an impressive accuracy of 98,18%, reaching state of the art performance. The second algorithm, while not reaching a higher accuracy like the previous one, was able to distinguish from different types of noise sources, while keeping a sensitivity of 89,77% for the detection of normal ECG signals, resulting in an accuracy of 70,74% for this DNN. Despite this difficulty in categorize which type of noise is affecting the ECG segment, the algorithm proposes that it is possible to perform such task with good results, which in itself reveals to be an improvement for the state of the art on noise detection DNNs.

The third DNN, which consisted in the classification of different pathological events, more specifically rhythmic arrhythmias, reached an accuracy of 56,85% accuracy and an average sensitivity of 61.13%. Even though the model did not perform with a state of the art accuracy, these results showed to be promising in the detection of these abnormalities, especially if trained with a higher number of arrhythmia data since the arrhythmia dataset available for training was not big enough.

The performance of the algorithms proves that it is possible to detect abnormal events in ECG using the proposed architectures, whereas the task of detecting the type of abnormalities that occur in the signals reveals to be more difficult due to the necessity of bigger amounts of training data. With that being said, the implementation of an encoder for classification algorithms proves to be effective in the detection of abnormal events by learning the signals key characteristics.

Even though the models presented didn't reach high performing results, this was expected since the algorithms developed sacrifice accuracy results in order to reach better generalizations for all ECG signals to be applicable in real life. The models proposed are then especially valuable to perform classifications of abnormalities in the ECG signal, without needing to be extremely specific to the classification that is set to perform.

6.2 Future Work

The developed work shows promising results, revealing to be applicable to perform good generalizations of the ECG signal, which is extremely beneficial for future use of this framework to detect abnormalities in ECG.

As for the noise detection algorithm, the binary class algorithm could be improved in order to be implemented in real time, by creating an application that allows the person that is acquiring ECG signal to receive at the same time information regarding the presence of noise in the signal. This is especially useful in order to control the sources of noise, guaranteeing that the signals that are obtained are clear, with all the necessary ECG characteristics visible. Furthermore, the multi-class algorithm could be improved in terms of accuracy, by either training the DNN with bigger amounts of data or by introducing ECG signals from wearable devices. These specific models, when applied in real time, would be notably useful for ECG wearables, allowing the user to have more control of the noise sources that affect the signal and for the device's developer to evaluate the wearable's quality.

The arrhythmia detection algorithm could also be improved, mainly by acquiring more data from different types of arrhythmia, opening the possibility to further add more classes to detect a wider number of pathological events. More preprocessing techniques could be explored for the preparation of the arrhythmia data, in order to preserve the key characteristics of the arrhythmias, while eliminating as much noise as possible.

Considering the performance of the noise and arrhythmia detection algorithms, an integration of both models into a single model could in the detection of different types of arrhythmias, by discarding noise affected segments in this task. Furthermore, the integration of both tasks into a single application would be useful for real life usability.

Given the high performance of the autoencoder for feature extraction of the key characteristics that determine the representation of a normal ECG, the identification of abnormalities in ECG as presented in this dissertation, could be applied in various tasks. The fact that the framework developed being able to perform great generalization then opens possibilities to apply ECG classification with active learning. Using the models proposed, an active learning approach could be especially helpful in the medical field through the detection of similar abnormalities in the ECG signal, of which the classifier actively creates new labels for patterns that diverge from the normal ECG signal and increasing new diagnostic possibilities. That said, this technique could not only aid in the diagnostic of arrhythmias, but also contribute to the study of these pathological anomalies, by finding correlations between the different expressions that the arrhythmias take on the ECG signal, and also by exploring these manifestations with different variables in mind such as the patient's gender, age, medications, among other factors.

Additionally, given the high generalization ability of the models, these algorithms could also be applied to different types of biological signals, such as EMG and respiratory signals. Particularly in the application on EMG signals, this framework could be used in the study of a prosthesis range of movements by detecting new movements. For respiratory signals, an abnormality detection model based on the algorithms proposed could be used to identify and classify pulmonary diseases for diagnosis, obstructive and restrictive patterns and to automatically detect coughing episodes or sudden stop of breathing, for patient monitoring.

Bibliography

[1] R. Silipo and C. Marchesi. “Artificial neural networks for automatic ECG analysis.” In: *IEEE Transactions on Signal Processing* 46.5 (1998), pp. 1417–1425. issn: 1053-587X. doi: 10.1109/78.668803.

[2] B. H. Brown, R. H. Smallwood, D. C. Barber, P. V. Lawford, D. R. Rose, and D. R. Shearer. *Medical Physics and Biomedical Engineering*. Medical Science Series. Bristol: Institute of Physics Publishing, 1999. isbn: 9781315275604.

[3] K. W. Johnson, J. T. Soto, B. S. Glicksberg, K. Shameer, R. Miotto, M. Ali, E. Ashley, and J. T. Dudley. “Artificial Intelligence in Cardiology.” In: *Journal of the American College of Cardiology* 71.23 (2018), pp. 2668 –2679. issn: 0735-1097. doi: https://doi.org/10.1016/j.jacc.2018.03.521. url: http://www.sciencedirect.com/science/article/pii/S0735109718344085.

[4] E. Coiera. *Guide to Health Informatics*. 2nd edition. London: Oxford University Press, 2003. isbn: 9780429192784. doi: https://doi.org/10.1201/b13618. url: http://www.coiera.com/.

[5] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya. “Deep learning for healthcare applications based on physiological signals: A review.” In: *Computer Methods and Programs in Biomedicine* 161 (2018), pp. 1 –13. issn: 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2018.04.005. url: http://www.sciencedirect.com/science/article/pii/S0169260718301226.

[6] H. Peter and W. Goodridge. “Integrating Two Artificial Intelligence Theories in a Medical Diagnosis Application.” In: *Artificial Intelligence Applications and Innovations: IFIP 18th World Computer Congress TC12 First International Conference on Artificial Intelligence Applications and Innovations (AIAI-2004) 22–27 August 2004 Toulouse, France*. Ed. by M. Bramer and V. Devedzic. Boston, MA: Springer US, 2004, pp. 11–23. isbn: 978-1-4020-8151-4. doi: 10.1007/1-4020-8151-0_2.

[7] D. Belo, J. Rodrigues, J. R. Vaz, P. Pezarat-Correia, and H. Gamboa. “Biosignals learning and synthesis using deep neural networks.” In: *Biomedical Engineering Online* 16.1 (2017), p. 115. issn: 1475-925X.

[8] A. L. Goldberger, L. A. Amaral, L Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic

signals." In: *Circulation* 101.23 (2000), E215–20. ISSN: 1524-4539. URL: http://www.ncbi.nlm.nih.gov/pubmed/10851218.

[9] J. D. Enderle, S. M. Blanchard, and J. D. Bronzino. *Introduction to Biomedical Engineering*. San Diego, USA: Academic Press, 2000. ISBN: 0-12-238660-4.

[10] H. D. Hesar and M. Mohebbi. "ECG Denoising Using Marginalized Particle Extended Kalman Filter with an Automatic Particle Weighting Strategy." In: *IEEE Journal of Biomedical and Health Informatics* 21.3 (2017), pp. 635–644. ISSN: 21682194. DOI: 10.1109/JBHI.2016.2582340.

[11] U. R. Acharya, J. Suri, J. A. E. Spaan, and S M. Krishnan. *Advances in Cardiac Signal Processing*. Oct. 2014, pp. 1–53. DOI: 10.1007/978-3-540-36675-1.

[12] M. Homaeinezhad, M. ErfanianMoshiri-Nejad, and H. Naseri. "A correlation analysis-based detection and delineation of ECG characteristic events using template waveforms extracted by ensemble averaging of clustered heart cycles." In: *Computers in Biology and Medicine* 44 (2014), pp. 66–75. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2013.10.024. URL: http://www.sciencedirect.com/science/article/pii/S0010482513003132.

[13] U. R. Acharya, J. Suri, J. A. E. Spaan, and S M. Krishnan. *Advances in Cardiac Signal Processing*. Oct. 2014, pp. 55–81. DOI: 10.1007/978-3-540-36675-1.

[14] U. R. Acharya, J. Suri, J. A. E. Spaan, and S M. Krishnan. *Advances in Cardiac Signal Processing*. Oct. 2014, pp. 121–155. DOI: 10.1007/978-3-540-36675-1.

[15] *Arrhythmia | National Heart, Lung, and Blood Institute (NHLBI)*. URL: https://www.nhlbi.nih.gov/health-topics/arrhythmia{\#}Signs,-Symptoms,-and-Complications (visited on 01/30/2019).

[16] J. Malmivuo and R. Plonsey. "Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields." In: 1995. Chap. 19. ISBN: 9780195058239. DOI: 10.1093/acprof:oso/9780195058239.001.0001.

[17] *Premature Ventricular Complex (PVC) • LITFL • ECG Library Diagnosis*. URL: https://litfl.com/premature-ventricular-complex-pvc-ecg-library/ (visited on 09/12/2019).

[18] J. Rodrigues, D. Belo, and H. Gamboa. "Noise detection on ECG based on agglomerative clustering of morphological features." In: *Computers in Biology and Medicine* 87 (2017), pp. 322 –334. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2017.06.009. URL: http://www.sciencedirect.com/science/article/pii/S0010482517301737.

[19] F. Xiong, D. Chen, Z. Chen, and S. Dai. "Cancellation of motion artifacts in ambulatory ECG signals using TD-LMS adaptive filtering techniques." In: *Journal of Visual Communication and Image Representation* 58 (2019), pp. 606–618. ISSN: 1047-3203. DOI: 10.1016/J.JVCIR.2018.12.030. URL: https://www.sciencedirect.com/science/article/pii/S1047320318303560.

[20] J. Webster. *22. Webster, J. G. (ed.), Medical instrumentation: application and design, Fourth edition, John Wiley Sons, Hoboken, NJ, 2010*. Jan. 2010.

[21] J. C. Huhta and J. G. Webster. "60-Hz Interference in Electrocardiography." In: *IEEE Transactions on Biomedical Engineering* BME-20.2 (1973), pp. 91–101. ISSN: 0018-9294. DOI: 10.1109/TBME.1973.324169.

[22] J Kameenoff. "Signal Processing Techniques for Removing Noise from ECG Signals." In: *Biomedical Engineering and Research* 1.1 (2017), p. 1.

[23] J. D. Bronzino. *The Biomedical Engineering Handbook*. Boca Raton, US: CRC Press, 1995.

[24] "Understanding Sampling." In: *Digital Signal Processing*. London: Springer London, pp. 3–16. DOI: 10.1007/978-1-84800-119-0_1. URL: http://link.springer.com/10.1007/978-1-84800-119-0{_}1.

[25] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[26] S. Haykin. *Neural networks*. Upper Saddle River, US: Prentice Hall International, 1999.

[27] S. Savalia and V. Emamian. "Cardiac Arrhythmia Classification by Multi-Layer Perceptron and Convolution Neural Networks." In: *Bioengineering* 5.2 (2018), p. 35. ISSN: 2306-5354. DOI: 10.3390/bioengineering5020035. URL: http://www.ncbi.nlm.nih.gov/pubmed/29734666http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC6027502http://www.mdpi.com/2306-5354/5/2/35.

[28] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. "How to Construct Deep Recurrent Neural Networks." In: *CoRR* (2013).

[29] A. M. Ghimes, A. M. Avram, and A. V. Vladuta. "A Character Prediction Approach in a Security Context using a Recurrent Neural Network." In: *2018 International Symposium on Electronics and Telecommunications (ISETC)* (Nov. 2018). DOI: 10.1109/ISETC.2018.8584007.

[30] K. Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." In: *Biological Cybernetics* 36.4 (1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791.

[32] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli. "Applications of Deep Learning and Reinforcement Learning to Biological Data." In: *IEEE Transactions on Neural Networks and Learning Systems* 29.6 (2018), pp. 2063–2079. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2018.2790388.

[33] Z. Mao, W. X. Yao, and Y. Huang. "EEG-based biometric identification with deep learning." In: *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*. 2017, pp. 609–612. DOI: 10.1109/NER.2017.8008425.

[34] C. Cao, F. Liu, H. Tan, D. Song, W. Shu, W. Li, Y. Zhou, X. Bo, and Z. Xie. "Deep Learning and Its Applications in Biomedicine." In: *Genomics, Proteomics Bioinformatics* 16.1 (2018), pp. 17 –32. ISSN: 1672-0229. DOI: https://doi.org/10.1016/j.gpb.2017.07.003. URL: http://www.sciencedirect.com/science/article/pii/S1672022918300020.

[35] U. R. Acharya, H. Fujita, O. S. Lih, Y. Hagiwara, J. H. Tan, and M. Adam. "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network." In: *Information Sciences* 405 (2017), pp. 81–90. ISSN: 0020-0255. DOI: 10.1016/J.INS.2017.04.012. URL: https://www.sciencedirect.com/science/article/pii/S0020025517306539.

[36] Y. LeCun and Y. Bengio. "The Handbook of Brain Theory and Neural Networks." In: ed. by M. A. Arbib. Cambridge, MA, USA: MIT Press, 1998. Chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. ISBN: 0-262-51102-9. URL: http://dl.acm.org/citation.cfm?id=303568.303704.

[37] Y. Baştanlar and M. Özuysal. "Introduction to Machine Learning." In: *miRNomics: MicroRNA Biology and Computational Analysis*. Ed. by M. Yousef and J. Allmer. Totowa, NJ: Humana Press, 2014, pp. 105–128. DOI: 10.1007/978-1-62703-748-8_7. URL: https://doi.org/10.1007/978-1-62703-748-8_7.

[38] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2004, p. 415. ISBN: 0-262-01211-1.

[39] I. H. Witten and E. Frank. *Data mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, 2005, p. 525. ISBN: 9780080477022.

[40] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng. "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network." In: *Nature Medicine* 25.1 (2019), pp. 65–69. ISSN: 1078-8956. DOI: 10.1038/s41591-018-0268-3. URL: http://www.ncbi.nlm.nih.gov/pubmed/30617320http://www.nature.com/articles/s41591-018-0268-3.

[41] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng. "Cardiologist-level arrhythmia detection with convolutional neural networks." In: *arXiv preprint arXiv:1707.01836* (2017).

[42] R. Rodrigues and P. Couto. "A Neural Network Approach to ECG Denoising." In: *CoRR* abs/1212.5217 (2012). URL: http://arxiv.org/abs/1212.5217.

[43] S. Ansari, J. Gryak, and K. Najarian. "Noise Detection in Electrocardiography Signal for Robust Heart Rate Variability Analysis: A Deep Learning Approach." In: *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 5632–5635.

[44] U. Satija, B. Ramkumar, and M. S. Manikandan. "Automated ECG Noise Detection and Classification System for Unsupervised Healthcare Monitoring." In: *IEEE Journal of Biomedical and Health Informatics* 22.3 (2018), pp. 722–732. DOI: 10.1109/JBHI.2017.2686436.

[45] J. N. John, C. Galloway, and A. Valys. "Deep Convolutional Neural Networks for Noise Detection in ECGs." In: *ArXiv* abs/1810.04122 (2018).

[46] R. J. Martis, U. R. Acharya, H. Prasad, C. K. Chua, C. M. Lim, and J. S. Suri. "Application of higher order statistics for atrial arrhythmia classification." In: *Biomedical Signal Processing and Control* 8.6 (2013), pp. 888–900. ISSN: 17468094. DOI: 10.1016/j.bspc.2013.08.008. URL: http://dx.doi.org/10.1016/j.bspc.2013.08.008.

[47] Y. Wang, Y. S. Zhu, N. V. Thakor, and Y. H. Xu. "A short-time multifractal approach for arrhythmia detection based on fuzzy neural network." In: *IEEE Transactions on Biomedical Engineering* 48.9 (2001), pp. 989–995. ISSN: 00189294. DOI: 10.1109/10.942588.

[48] F. Sufi and I. Khalil. "Diagnosis of Cardiovascular Abnormalities From Compressed ECG: A Data Mining-Based Approach." In: *IEEE Transactions on Information Technology in Biomedicine* 15.1 (2011), pp. 33–39. ISSN: 1089-7771. DOI: 10.1109/TITB.2010.2094197.

[49] R. S. Andersen, R. Peimankar, and S. Puthusserypady. "A Deep Learning Approach for Real-Time Detection of Atrial Fibrillation." In: *Expert Systems with Applications* 115 (Aug. 2018). DOI: 10.1016/j.eswa.2018.08.011.

[50] *biosignalsplux | wearable body sensing platform - biosignalsplux | wearable body sensing platform*. URL: https://www.biosignalsplux.com/en/ (visited on 01/30/2019).

[51] *biosignalsplux | wearable body sensing platform - Software*. URL: https://www.biosignalsplux.com/en/software (visited on 01/30/2019).

[52] *PyCharm: the Python IDE for Professional Developers by JetBrains*. URL: https://www.jetbrains.com/pycharm/ (visited on 01/30/2019).

[53] *TensorFlow*. URL: https://www.tensorflow.org/ (visited on 01/31/2019).

[54] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, I. Polat, Y. Feng, E. W. Moore, J. Vander Plas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt. "SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python." In: *arXiv e-prints* (2019). eprint: `1907.10121` (cs.MS).

[55] N. Iyengar, C. Peng, R. Morin, A. L. Goldberger, and L. A. Lipsitz. "Age-related alterations in the fractal scaling of cardiac interbeat interval dynamics." In: *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* 271.4 (1996), R1078–R1084.

[56] G. B. Moody, W. Muldrow, and R. G. Mark. "A noise stress test for arrhythmia detectors." In: *Computers in cardiology* 11.3 (1984), pp. 381–384.

[57] G. B. Moody and R. G. Mark. "The impact of the MIT-BIH arrhythmia database." In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.

[58] *scipy.signal.decimate — SciPy v1.2.1 Reference Guide*. URL: `https://docs.scipy.org/doc/scipy-1.2.1/reference/generated/scipy.signal.decimate.html` (visited on 09/12/2019).

[59] G. Montavon, G. Orr, and K.-R. Müller. *Neural Networks: Tricks of the Trade*. Vol. 7700. Springer, 2012.

[60] T. Tieleman and G. Hinton. *Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude*. COURSERA: Neural Networks for Machine Learning. 2012.

[61] L. v. d. Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.