TÉCNICO LISBOA

An Artificial Intelligence Approach for Atrial Fibrillation Detection in Single-lead Invisible ECG

Rafael Luís Soares da Costa e Silva

Thesis to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor(s): Prof. Ana Luísa Nobre Fred
Dr. Hugo Humberto Plácido da Silva

Examination Committee

Chairperson: Prof. João Miguel Raposo Sanches
Supervisor: Prof. Ana Luísa Nobre Fred
Member of the Committee: Prof. Maria Margarida Campos da Silveira

November 2021

”Sê todo em cada coisa.
Põe quanto és no mínimo que fazes.”
— Fernando Pessoa

Agradecimentos

Foram muitos os desafios que atravessei durante os 8 meses em que estive a realizar a tese de mestrado. Felizmente, tive o apoio da família e dos amigos para realizar este trabalho que foi tão importante para definir o quero fazer como Engenheiro Biomédico. Todos os que estiveram perto de mim ajudaram-me a chegar onde estou, e a finalmente apresentar o último projeto do curso. Por isso, gostaria de agradecer a todos os que sempre me apoiaram desde o início do curso, especialmente aos meus pais, à minha irmã e aos meus amigos mais próximos.

Gostaria também de expressar a minha gratidão pelo apoio e motivação que o professor Hugo Silva sempre me deu. Em tempos de pandemia, o professor foi incansável em seguir o meu progresso, mesmo com os constrangimentos pessoais que pudessem surgir, e tornou o meu trabalho mais significativo ao dar-me a oportunidade para trabalhar no projeto com o Instituto de Telecomunicações. Agradeço também a dedicação da professora Ana Fred que, apesar de todos os desafios, ajudou a tornar o meu trabalho mais rico e a ganhar sentido crítico no tema específico em que trabalhei. Foi um prazer tê-los como meus orientadores! O entusiasmo com que eles trabalham numa área que me fascina motiva-me a querer contribuir com o melhor que puder. Gostaria de também agradecer ao professor André Lourenço que esteve disposto a envolver-me num dos projetos da CardioID que, sem dúvida, entusiasmou-me muito para este trabalho.

Também manisfeto o meu agradecimento ao Instituto de Telecomunicações, que, além de acolher o tema desta tese, providenciou de forma prática todos os recursos necessários para poder realizar meu trabalho.

Preface

The work presented in this thesis was performed at Instituto de Telecomunicações (IT) in Lisbon, Portugal, during the period from March 2021 to October 2021, under the supervision of professor Hugo Humberto Plácido da Silva. The thesis was co-supervised at Instituto Superior Técnico by Prof. Ana Luísa Nobre Fred, affiliated with Departamento de Bioengenharia (DBE). The work also had the support from professor André Lourenço, CEO at CardioID Technologies.

The objective of this master's thesis is to obtain the Master of Science Degree in Biomedical Engineering at Instituto Superior Técnico (IST) and to obtain the Master in Technological Innovation in Health (MTiH) provided by the European Institute of Innovation and Technology (EIT) Health.

EIT Health[1] is a Knowledge and Innovation Community (KIC) of EIT focused on health that aims to strengthen the connections between industry, research centers, and universities across Europe. To engage and train students in this environment, EIT Health created the MTiH program for master students. It involves completing subjects related to Innovation and Entrepreneurship (I&E), working on innovative projects, and encouraging students to do an internship with a health-related company to stimulate innovation.

Motivated by the MTiH program, the opportunity to get involved in a project between IT and CardioID only seemed appropriate. CardioID Technologies[2], founded in 2014, is a spin-off company that started within IT. The company develops innovative solutions for Electrocardiography (ECG) recording, and it has successfully developed applications for automotive, sports, critical facilities, and IoT contexts. It is currently focused on heart biometrics, pathology detection, fatigue assessment, among others.

The project mentioned above between IT and CardioID involves the development of an ECG recording and monitoring system to use in sanitary facilities for OLI — Sistemas Sanitários[3]. OLI is a company that develops robust and efficient solutions for sanitary facilities, including the fabrication of cisterns, flush plates, and toilet mechanisms. It is the largest cistern producer in Southern Europe, and it has branded products in 80 countries.

[1] https://eithealth.eu/
[2] https://www.cardio-id.com/
[3] https://www.oli-world.com/

Resumo

As doenças cardiovasculares (DCVs) são atualmente a principal causa de mortalidade a nível mundial. Uma vez que algumas DCVs podem ocorrer sem sintomas e ainda serem prejudiciais, a sua deteção precoce e a monitorização de doentes de risco fora do ambiente hospitalar é crucial para evitar taxas de mortalidade ainda mais elevadas. Por essa razão, um paradigma recente de sistemas de monitorização contínua baseia-se na integração da monitorização fisiológica com a vida quotidiana do paciente, utilizando tecnologias *wearable* e *invisible*.

Uma das DCVs mais relevantes é a fibrilhação auricular (FA), associada a um risco acrescido de acidentes vasculares cerebrais. Devido à sua crescente prevalência e custos para os sistemas de saúde, várias abordagens para detetá-la foram desenvolvidas nos últimos anos. Utilizando registos eletrocardiográficos (ECG) de uma única derivação da *PhysioNet Computing in Cardiology Challenge 2017* (CinC2017), foi desenvolvido um algoritmo baseado em redes neuronais artificiais para distinguir a FA do Ritmo Sinusal Normal (RSN).

O modelo proposto envolve acoplar uma versão comprimida de segmentos de ECG gerados por um Autoencoder (AE) não supervisionado e um classificador de aprendizagem automática. Um Autoencoder Esparso (SpAE) e um Perceptrão Multicamada (MLP) obteve uma F1-score de 82.2%. Ao acrescentar uma característica relativa a alterações locais dos intervalos RR em torno de um pico R, a F1-score melhorou para 88.2%. Embora simples, esta abordagem comprova que os AEs podem superar os algoritmos que utilizam as mesmas características, e que estes podem ser melhorados para alcançar taxas de desempenho ainda mais elevadas.

Palavras-chave: Deteção de Fibrilhação Auricular, ECG de Derivação Única, Autoencoders, Aprendizagem Automática

Abstract

Cardiovascular diseases (CVDs) are currently the leading cause of mortality worldwide. Since some CVDs can occur without symptoms and still be harmful, early detection and monitoring of patients at risk outside the hospital environment is crucial to avoid even higher mortality rates. For this reason, a recent paradigm of continuous monitoring systems is based on integrating physiological monitoring with the patient's daily life, using wearable and invisible technologies.

One of the most relevant CVDs is Atrial Fibrillation (AF), associated with an increased risk of stroke. Due to its increasing prevalence and costs to healthcare systems, several approaches to detect it have been developed in recent years. Using single-lead electrocardiographic (ECG) recordings from PhysioNet Computing in Cardiology Challenge 2017 (CinC2017), an artificial neural network-based algorithm was developed to distinguish AF from Normal Sinus Rhythm (NSR).

The proposed model involves coupling a compressed version of ECG segments generated by an unsupervised Autoencoder (AE) and a Machine Learning (ML) classifier. A Sparse Autoencoder (SpAE) and a Multilayer Perceptron (MLP) obtained an F1-score of 82.2%. By adding a feature concerning local changes of RR intervals around an R peak, the F1-score improved to 88.2%. Although simple, this approach proves that AEs can outperform algorithms using the same features, and that these can be improved to achieve even higher performance rates.

Keywords: Atrial Fibrillation Detection, Single-lead ECG, Autoencoders, Machine Learning

Contents

List of Tables

List of Figures

List of Acronyms

AE Autoencoder. xv, xvii, xviii, 3–5, 15–17, 23, 29–39, 41–43, 45, 47–50, 54, 56–59, 75

AF Atrial Fibrillation. xv, xvii, xviii, 2–5, 11, 12, 19–24, 26, 29–32, 35–39, 42–47, 49, 50, 52–54, 56–59, 76–80

AFDB MIT-BIH Atrial Fibrillation Database. 24, 59

AHADB American Heart Association Database. 24

ANN Artificial Neural Network. 4, 5, 12–15, 19, 21, 23, 27, 33, 35, 37, 57, 59

AUC area under the curve. xviii, 18, 36, 47, 49–51, 53–56, 58

AV atrioventricular. 8, 11, 12, 24

CAE Contractive Autoencoder. xviii, xix, 33, 43, 44, 47–49, 58, 76, 78, 81

CART Classification and Regression Trees. 35, 47–49

CCDD Chinese Cardiovascular Disease Database. 24

CinC2017 Computing in Cardiology Challenge 2017. xv, xvii, 11, 24–26, 30, 31, 37, 42, 56, 58, 59

CNN Convolutional Neural Network. 23

CVD cardiovascular disease. 1, 7, 25, 26, 57

DAE Denoising Autoencoder. xviii, xix, 16, 33, 43, 44, 47–49, 58, 77, 79, 81

DT Decision Tree. 19, 35, 58

ECG Electrocardiography. xvii, xviii, 1–5, 7–12, 22–27, 29–33, 37–39, 41–45, 49, 50, 52, 56–59, 76, 77

EHR Electronic Health Record. 10

FIR Finite Impulse Response. 30

HR heart rate. 6, 38

HRV heart rate variability. 6, 38, 57–59

Hz Hertz. 6, 24, 30, 31

k-NN k-Nearest Neighbor. 19, 21, 35, 47, 48, 58

LCSD Local Change of Successive Differences. xv, xvii–xix, 3, 38, 39, 54–56, 58, 80

LSTM Long Short-Term Memory. 23, 59

LTAFDB Long Term Atrial Fibrillation Database. 24

MITDB MIT-BIH Arrhythmia Database. 24, 58, 59

MLP Multilayer Perceptron. xvii, xviii, 13, 15, 35–37, 47–56, 58

MSE Mean Squared Error. xv, 13, 15, 33, 36, 41–43, 50, 52

NSR Normal Sinus Rhythm. xvii, xviii, 10–12, 24, 26, 31, 32, 35–39, 42–47, 49, 50, 52–54, 56, 58, 76–80

PPG Photoplethysmography. xvii, 1, 3–7, 19, 57

RAE Robust Autoencoder. xviii, xix, 33, 43, 44, 47–49, 58, 76, 78, 81

ReLU Rectified Linear Unit. 35, 36, 41, 42, 49, 50, 80

RMSSD Root Mean Square of Successive Differences. 38, 39

RNN Recurrent Neural Network. 23

ROC Receiver Operating Characteristic. xvii, xviii, 18, 36, 47, 50, 51, 53–55, 59

SAE Standard Autoencoder. xviii, xix, 33, 36, 41, 43, 45–49, 58, 76, 78, 81

SDG Stochastic Gradient Descent. 14

SpAE Sparse Autoencoder. xv, xviii, xix, 16, 23, 33, 43, 47–56, 58, 77, 79–81

SVM Support Vector Machine. 19, 21, 35, 47–49, 52, 54, 58

t-SNE t-Distributed Stochastic Neighbor Embedding. xviii, xix, 45, 46, 52, 53, 81

TanH hyperbolic tangent. 36, 42, 49, 50

VAE Variational Autoencoder. xviii, xix, 16, 23, 34, 43, 45–49, 56, 58, 59, 77, 79, 81

WHO World Health Organization. 1

Chapter 1

Introduction

The present introductory chapter contains a topic overview that briefly presents current statistics and trends around cardiovascular monitoring, and a section that provides the main context motivating this work. The established objectives, the main achievements and an outline of the thesis structure are also provided.

1.1 Topic Overview

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the leading cause of mortality worldwide, with 17.9 million deaths estimated in 2019, representing 32% of the world's total deaths [1]. CVDs include several health conditions involving blood vessels and heart function, such as heart failure, coronary heart disease, stroke, myocardial infarction, and arrhythmia.

Since some heart conditions can occur with mild or no symptoms and still be harmful, early detection of heart disease and monitoring risk patients out of the hospital environment is crucial to avoid even higher mortality rates. For instance, silent myocardial infarction is associated with an increased risk of heart failure [2]; also, atrial fibrillation, responsible for 15% of all strokes, can appear with no symptoms [3]. Studies have equally pointed out that CVDs could be better diagnosed and prevented by continuous monitoring [4]. For this reason, home and remote monitoring of cardiac functions have gained more relevance over the years [4], becoming increasingly pervasive through the integration of cardiovascular assessment sensors in smartwatches and other wearable devices. Especially in wearables, this type of monitoring can be achieved by recording the heart's activity, using techniques such as the Photoplethysmography (PPG) or Electrocardiography (ECG).

Because of improvements in sensor technology, communications, and software tools, many monitoring devices arose in the market, making them available to clinicians and the general population. However, healthcare professionals are not expected to analyze the overwhelming amount of data generated by these devices. Instead, algorithms embedded in the device systems are responsible for analyzing and extracting useful information from the records, to detect and evaluate the presence of a heart condition. Nevertheless, after a potential diagnosis, clinical evaluation by a physician is of utmost importance.

Various types of algorithms for cardiac anomaly detection have been developed over the years, mainly focusing on detecting premature contractions, atrial fibrillation, myocardial infarction, and congestive heart failure [5]. Because of the nature and complexity of the problem, many approaches have been explored, but there is still no gold standard approach for mainstream use. A topical research area both in academia and industry is therefore the development of more pervasive and robust approaches that, more recently, build upon machine learning and artificial intelligence techniques.

1.2 Motivation

Cardiac monitoring devices such as smartwatches and other wearable sensors, *i.e.*, on-the-person [6], are now present in home, hospital, ambulatory, and remote settings [4], becoming an affordable way of monitoring health status and fitness. However, despite reaching mainstream market, recent surveys have shown that within 6 months of use 30% of users abandon their wearable trackers [7].

Novel approaches that integrate ECG sensors with everyday use objects are a clear alternative that doesn't require a conscious effort from the user to wear the device and make acquisitions. These approaches are known as *invisible* or off-the-person, and examples of such applications are sensors integrated into chairs, phone cases, steering wheels and keyboards [6, 8]. Other approaches based on image analysis and radio frequency sensing systems don't require direct contact with the user [9].

The development of such devices is not as straightforward as one would expect, since the electrodes and other hardware components must be adapted to their acquisition environment. For instance, in the case that motivates this thesis workflow, an invisible ECG approach for sanitary facilities, the electrodes are placed on a toilet seat, where environmental constraints are especially relevant. Since the electric contact between the electrodes and the body can be affected by the presence of hair and humidity, the material and texture choices of the electrodes were crucial to obtaining measurable ECG signals [10].

Since the context of acquisition can greatly affect signal quality, algorithms should consider the expected input signal properties. Also, when evaluating real-world deployment, processing and storage platforms (*e.g.*, IoT, edge computing or mobile computing) can be an important factor in choosing the methodologies to be used. For this reason, and since the use of algorithms for ECG anomaly detection in such implementations is not well established, this thesis aims to provide insight on suitable algorithmic approaches for cardiovascular diagnosis based on invisible ECG.

One of the most relevant cardiovascular diseases is Atrial Fibrillation (AF). Because of its high prevalence and costs to the healthcare systems [11, 12], the detection of Atrial Fibrillation (AF) was chosen to be the main focus of the developed algorithms. Also, in recent years, there has been an increased interest in developing algorithms for its detection [13], making it possible to compare different approaches and evaluate their performance.

1.3 Objectives

The work of this thesis aims to develop an artificial intelligence algorithm to detect AF. Besides that main goal, several other objectives can be stated, namely:

- Describe the main physiological processes that are targeted in cardiovascular monitoring and of AF mechanisms.
- Explore the particular challenges in the context of wearable and *invisible* monitoring modalities, and contrasting them with the conventional methods.
- Provide insight on the current algorithmic approaches to detect AF, and compare their performances.
- Develop an algorithmic approach based on single-lead ECG to detect AF that matches or surpasses the state-of-the-art.
- Explore various types of configurations and discuss their differences.
- Propose a final approach that presents the best results and discuss its viability for real-world use.
- Characterize possible enhancements to the approach and propose future work.

1.4 Contributions

This thesis made some contributions that, hopefully, are appreciated by the scientific community, namely:

- The development of PPG segmentation algorithm that is especially adapted to reflective mode acquisitions, now available at the BioSPPy Python toolbox[1].
- An article that studies the impact of sampling rate and interpolation methods on the quality of reflective PPG signals, which awaits approval to be published.
- The development of a novel technique based on Autoencoders (AEs) for AF detection using ECG segments.
- The proposal of a metric that evaluates local changes of RR-intervals around an R-peak, called Local Change of Successive Differences (LCSD).
- An R-peak median-based algorithm to correct inverted ECG records.

[1] `https://github.com/PIA-Group/BioSPPy`

1.5 Thesis Outline

Besides the current chapter, which provides the context, motivation and objectives of this thesis, *Chapter 2 - Background* describes the main working principles and physiological mechanisms around PPG and ECG recordings, and AF. It also provides theoretical notions of Artificial Neural Networks (ANNs), AEs, and classification metrics.

Chapter 3 - State of the Art summarizes the current trends in AF detection algorithms, characterizes the most relevant ECG databases to this work, and provides a brief insight on how ECG monitoring systems are structured.

In *Chapter 4 - Proposed Approach* the process of designing and choosing a classification model to detect AF based on AEs is presented, including the description of the preprocessing steps and how the algorithms will be trained and evaluated.

Chapter 5 - Results aims to report the results of the proposed approach, compare the various experiments, and justify the choices that lead to the best model, which is then compared to the state-of-the-art.

Finally, *Chapter 6 - Conclusions* provides an overview of the main findings and achievements, and it also provides future work ideas stemming from this work.

Chapter 2

Background

This section aims to introduce and describe basic physiological principles around Photoplethysmography (PPG) and Electrocardiography (ECG), which are two main non-intrusive physiological modalities to study the cardiovascular system and extract important features to detect disease and abnormalities. To integrate cardiovascular monitoring in the remote context, the advantages and disadvantages of using PPG or single-lead ECGs are briefly discussed, and an overview of atrial fibrillation pathophysiology is also presented. In addition, this chapter provides insight into the approach to detect AF used in this work, the AE. The fundamentals of the basic structure and functioning of an ANN, and standard classification metrics are also described.

2.1 Photoplethysmography

Numerous technological advancements over the last decades have led to the development of techniques that can monitor vital signs in the human body non-intrusively. One of the simplest techniques to study the cardiovascular system is Photoplethysmography (PPG), first described by Alrick Hertzman in 1938 as a non-invisible optical technique to measure blood volume changes in peripheral tissues, such as the toes and fingers [14]. The basis of the technique relies on light interacting phenomena with the tissues and blood vessels. The signal recorded by coupling a light source and a photodetector can be used to observe the light intensity levels along time (Figures 2.1 and 2.2), which is modeled by several factors, including blood volume, blood vessel wall movements, and the orientation of the red blood cells [15].

Another factor that plays an important role when recording the PPG signal is the wavelength of the light source, which can affect signal strength and quality: the water present in the tissues and blood absorbs specific wavelengths more than others (e.g., ultraviolet and longer infrared are strongly absorbed); the differences between the absorption spectra of oxyhemoglobin and deoxyhemoglobin can affect the signal; different light wavelengths have different skin penetration lengths. Red and infra-red light sources have been preferred and widely adopted in PPG devices, since these wavelengths are within the water's optical window and have an adequate skin penetration length [16].

Moreover, different acquisition modes are possible when recording a PPG signal: transmissive, where the tissue is placed in between the light source and the photodetector (e.g., finger clip); and reflective, where the light source and photodetector, in contact with the skin, are placed side by side as in Figure 2.1 (e.g., fitness tracker, smartwatches) [15]. Generally, commercial devices for clinical use the transmissive configuration with red and infra-red light sources, and everyday use sensors, often integrated with wearable technologies, use the reflective configuration with a green light source. Green light, with shorter skin penetration length, has been reported to be more resistant to motion artifacts [16, 17], thus more suitable for wearable applications [16, 15].

Because PPG records can be used to monitor heart rate, blood pressure, blood oxygenation levels, and others [16], they can be a rich source of information regarding the state of the cardiovascular system [18]. Also, current literature around reflective PPG signal processing is not as developed as the literature around the standard transmissive PPG signal. Motivated by such relevance in cardiac monitoring and potential contribution to the literature, a brief study of this signal was first conducted upon the start of the thesis work.

A core practical question regarding PPG acquisition is the signal quality obtained as a function of the sampling frequency of the Analog-to-Digital Converter (ADC), which is essential, for instance, to accurately compute heart rate (HR) and heart rate variability (HRV) metrics. This question is crucial for obtaining signals from limited-resource systems, such as many wearable devices, which have seen increased interest in recent years, or devices with shared computational resources to record other signals simultaneously.

To study the sampling frequency impact of such acquisition systems in PPG signal quality, several sampling frequencies were tested, and a proposal for a minimal high-quality acquisition was made. To achieve this, a PPG signal dataset originally acquired at 1 kHz was resampled into various downsampling frequencies (20, 50, 100, 200, 500 Hertz (Hz)), and, by computing peak differences with the original signal, time and amplitude errors were obtained. Since a common practice to enhance signal quality is using interpolation methods, their time and amplitude enhancements were also evaluated. Moreover, the distortion caused by PPG resampling and interpolation was assessed using the Pearson correlation coefficient, by computing the linear correlation between the original PPG segments and the generated ones.

By extracting statistical metrics from the time and amplitude errors, and the Pearson correlation coefficient values, the study showed that a 50 Hz sampling frequency with a quadratic or cubic spline interpolation achieved temporal and amplitude resolution identical to a 1 kHz sampled signal. These results are further detailed in an article that has been submitted and is currently waiting decision. This work also enriches the current literature regarding wearable PPG acquisitions since the signal was acquired using the reflective mode of operation with green light, an approach that is not well developed in the current literature.

Because it only requires a single contact point with the body and uses simpler instrumentation, PPG monitoring is easily achieved and mainstreamed. Also, since the pulse rate variability is highly correlated with HRV, the PPG signal is a reliable source of HR and HRV metrics [19, 20, 21]. Several features from

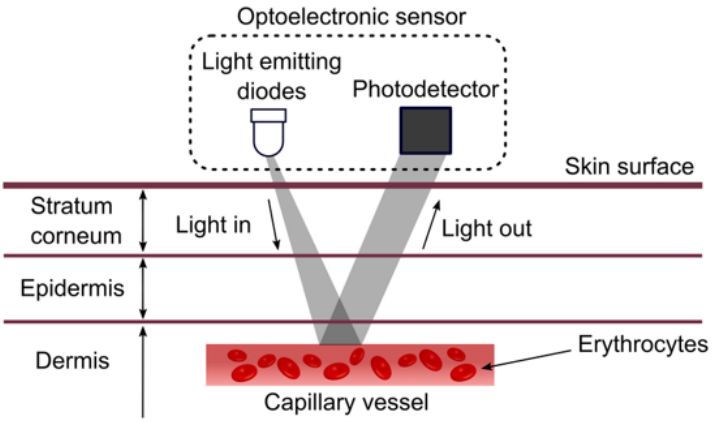


Figure 2.1: Representation of the PPG signal's acquisition principle using a reflective mode of operation, where the light source and photodetector are placed side by side. Extracted from [23].

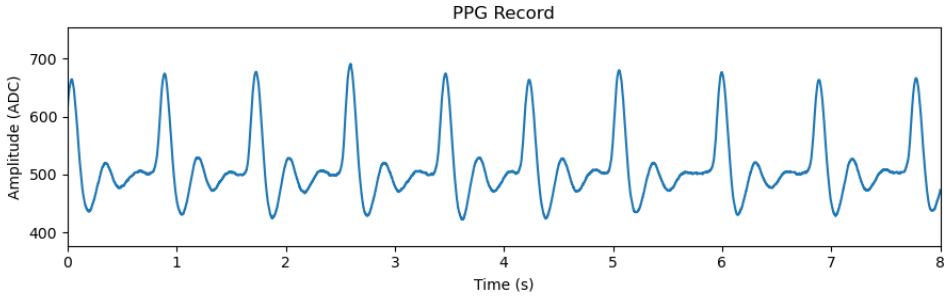


Figure 2.2: Example of a PPG record using the reflective mode of operation.

the PPG waveform and derivatives can even be extracted to assess the risk of CVDs [16, 15]. However, in the pursuit of developing an algorithm to monitor cardiac function and detect heart disease, the PPG signal can be somewhat limited since it is not able to capture relevant cardiac dynamics involving, for instance, the condition of the heart's electrical system based on ECG fiducials. Also, to be able to perform accurate diagnoses, the PPG waveform has to be of high-quality, with little noise or artifacts, for which there are several concerns [22]. For this reason, PPG has been mainly used for fitness and wellness purposes and not for medical ends. Therefore, since the ECG can be a richer source of information, this signal was chosen to be the target of the subsequent studies.

2.2 Electrocardiogram

Electrocardiography (ECG) is the recording of the electrical activity of the heart, which has a conduction system responsible for propagating the electrical "pacemaker" pulses generated at the sinoatrial node (or sinus node) throughout the muscle tissue, i.e., the myocardium. These action potentials reach atrial and ventricular myocardium and provoke muscle contraction, enabling the heart to pump blood through the cardiovascular system. Although the heart's electrical activity originates from its conduction system, the ECG results from changes in polarization of the cardiac muscle since the former does not reach the surface of the body, where the ECG is recorded [24]. However, changes and patterns in the

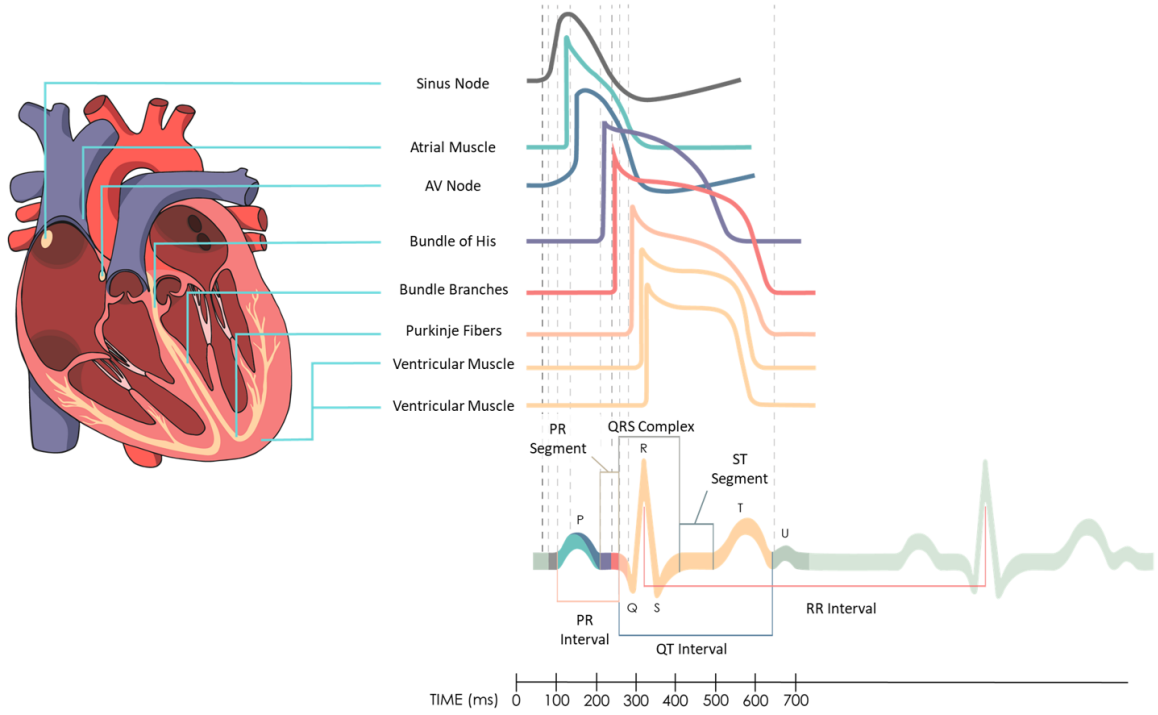


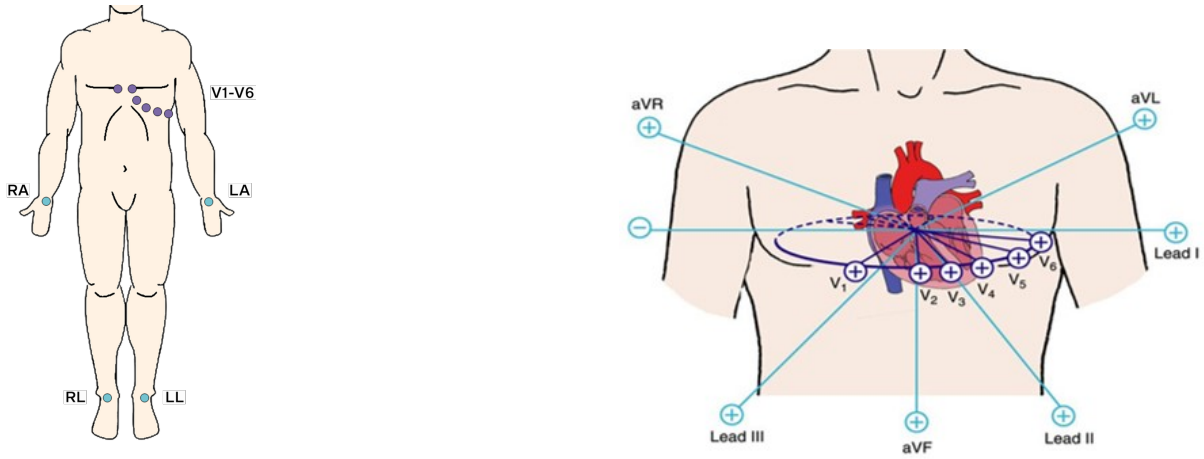
Figure 2.3: Heart's conduction system, action potentials and description of the ECG waveform. Adapted from [25].

ECG can reveal the condition of the conduction system.

The ECG waveform can be described by how the different parts of the heart's muscles and conduction system are activated (Figure 2.3). When the sinus node, located in the right atrium and part of the conduction system, generates action potentials, these start to be propagated through the atria myocardium, and a P-wave appears. Because the depolarization wave is oriented to the left and downwards, the P-wave will have a positive deflection on voltage measurements, or leads, matching this direction.

When electrical pulses reach the atrioventricular (AV) node, an isoelectric delay called the PR segment appears. Reaching the His-Purkinje system, with origin in the septal portion that separates both ventricles, a Q-wave of muscle depolarization is formed and directed towards the right, thus negatively deflected. Then, through the right and left bundles and Purkinje fibers, a depolarization wave is propagated in both ventricles, and the R peak is recorded, where the ECG reaches its maximum amplitude. Because the left myocardium has greater muscle thickness than the right one, the depolarization wave mean vector is oriented towards the left. Since depolarization continues in the direction of the upper left portion of the left ventricle, a negative deflection appears (S-wave). After a brief period of no electrical activity (ST segment), ventricular repolarization occurs, and a T-wave is generated. Following the T-wave, a U-wave can be noticed in some ECG recordings, attributed to the repolarization of the His-Purkinje system [24].

In a clinical environment, ECG recordings are often made with several electrodes placed on predefined locations of the body to obtain 12 lead records, *i.e.*, the 12-lead ECG. In this system, 6 leads cover



(a) (b)

Figure 2.4: (a) Standard electrode placement for the 12-lead system. (b) Electrocardiographic views of the heart, reproduced with permission from [28]. Copyright 2021 Wolters Kluwer Health, Inc.

the frontal plane – bipolar leads I, II, and III and unipolar leads aVR, aVL, and aVF – and 6 leads cover the horizontal plane – the unipolar precordial leads V1, V2, V3, V4, V5, and V6 (Figure 2.4). This allows the observation of different projections of the heart's electrical activity (Figure 2.5) and to perform differential diagnoses in patients [24]. Other lead positioning systems are also possible and with fewer leads, such as the Mason-Likar's, Frank's and EASI lead systems[1]. These alternative configurations may be useful when there is need to better understand certain regions of the heart, or to perform ambulatory acquisitions such as the Holter. The Holter monitor is a small device attached to the patient that makes continuous ECG acquisitions for 24 to 48 hours, generally using 3 to 8 leads.

Aside from the hospital setting, ECG acquisitions can be easily done in home, ambulatory and remote environments. These can be very useful for health practitioners and convenient for patients, whom would benefit more with these kind of ECG recording modalities. Because of the unpredictable nature of heart anomaly episodes, pervasive ECG monitoring out of the hospital can provide more insight on the patient's condition and can avoid the high costs of hospitalization [4, 26, 27]. For these and for commodity reasons, simple and portable ECG monitoring devices have been developed, with special emphasis on single-lead ECG data acquisition.

2.3 Single-lead Electrocardiogram

As the name suggests, single-lead ECG consists of only one lead ECG recordings, in an approach that requires only three or two (if a virtual reference is used [30]) electrodes on the body surface (Figure 2.6).

Although the single-lead ECG does not convey as much information as the 12-lead ECG, several ECG anomalies can be detected, such as atrial fibrillation, atrial flutter, atrioventricular blocks, premature atrial and ventricular contractions, left and right bundle branch blocks, among others [31]. A typical usage of single-lead ECG is to detect atrial fibrillation, which is the most common type of heart arrhythmia in

[1] For further details visit `https://ecgwaves.com/course/the-ecg-book/`

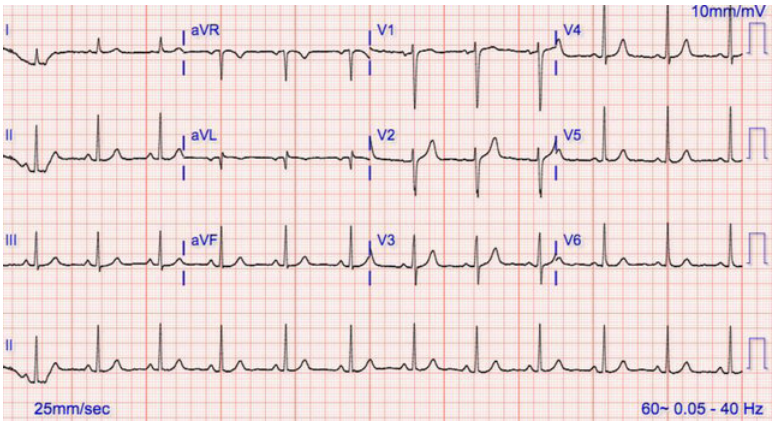


Figure 2.5: Normal Sinus Rhythm (NSR) 12-lead ECG record. Extracted from [29].

the Western World and one of the major causes of stroke, as described in Section 1.2.

Since only one lead is used, detecting these anomalies has some degree of uncertainty. For example, atrial fibrillation and premature atrial contractions can be difficult to distinguish, atrial flutter can be challenging to be detected, and bundle branch blocks can be mistaken for intraventricular conduction delays [31]. Also, single-lead ECGs have limitations in fully diagnosing acute coronary syndrome, myocardial infarction, and left ventricular hypertrophy [31].

Nevertheless, single-lead ECG devices can be a valuable first-line tool to promptly detect potential heart conditions and monitor the state of patients. ECG records can be adapted for diverse purposes such as hospital, ambulatory, home, and fitness monitoring. Also, information can be instantly directed to the patient's Electronic Health Record (EHR), smartphone applications, or cloud services, that more quickly connect patients with their physicians.

A key aspect of single-lead ECG monitoring is that it can be incorporated into the user's daily life, since the devices are usually compact and easy to use. It can even be incorporated with everyday use objects such as smartwatches, phone cases, and even invisibly objects, where the electrodes are merged with objects that the user is in frequent contact with (*e.g.*, chairs, keyboards, game controllers, car steering wheels) [9, 32]. More recently, invisible ECG has even been successfully integrated in a standard household toilet seat [10].

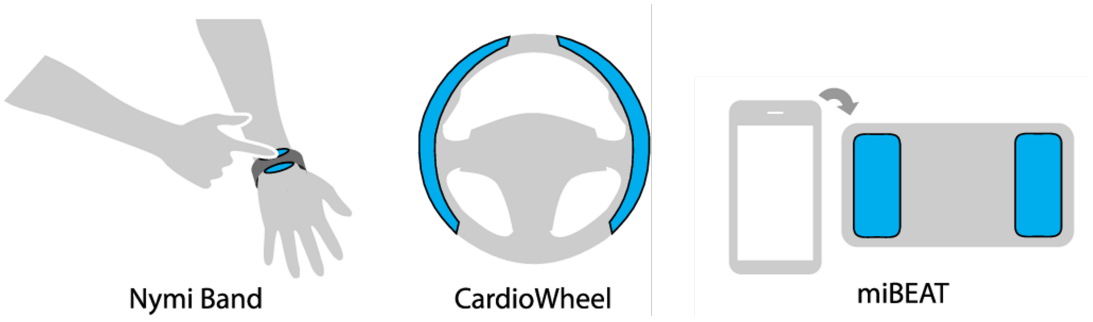


Figure 2.6: Different acquisition approaches to record single-lead ECG available in the market. Extracted from [32].

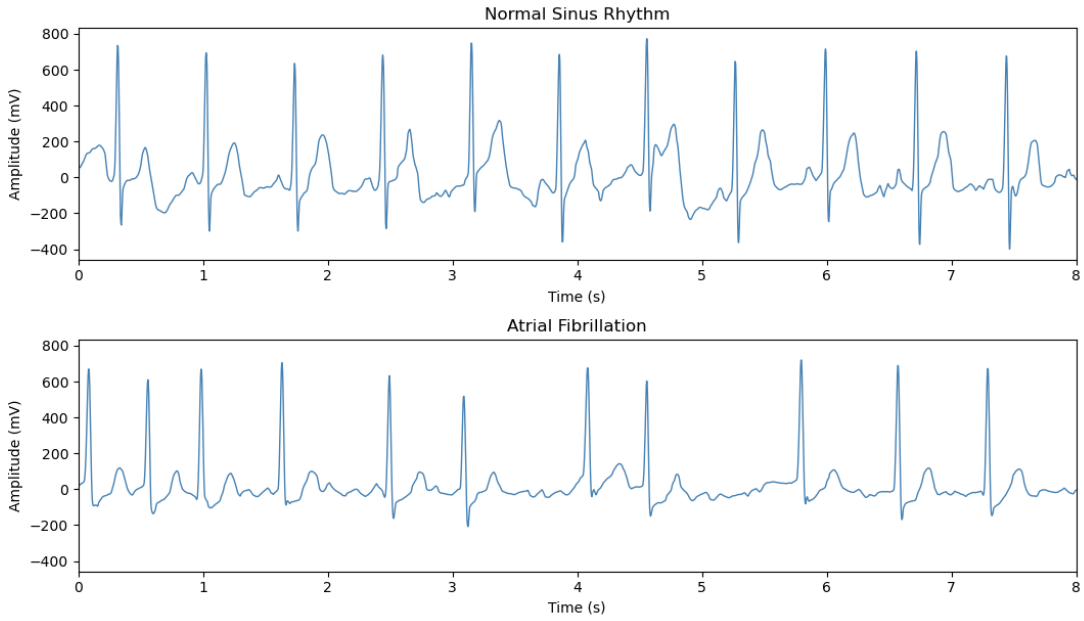


Figure 2.7: Comparison between Normal Sinus Rhythm and Atrial Fibrillation ECG waves. Data from the Computing in Cardiology Challenge 2017 database[1].

2.4 Atrial Fibrillation

Atrial Fibrillation (AF) is a type of arrhythmia, an uncoordinated activity of the heart chambers, in which the heart presents an irregular and high-rate electrical activity of the atria. It is the most common type of heart arrhythmia in the Western World with an estimated prevalence of 46.3 million people around the globe [33, 34], and it is one of the main causes of stroke since it increases the risk of having one by 4 to 5 fold [35]. Also, around 35% of patients that recover from cardiac surgery have an AF episode [36], and the risk of having AF is higher in men, increasing with age [34]. Other risk factors include heart failure, hypertension, ischemic heart disease, diabetes, hyperthyroidism, sleep apnea, among others [37, 34].

When observing the ECG, AF is characterized by having ”irregularly irregular” heart rhythms, absence of P-waves, nonexistence of an isoelectric baseline and variable ventricular rate [38], as depicted in Figure 2.7. Also, P-wave activity can be replaced by fibrillatory waves (f-waves), which can be either fine (amplitude $<0.05\,\mathrm{mV}$) or coarse (amplitude $>0.05\,\mathrm{mV}$) [39].

AF occurs because of a lack of synchrony in atrial contraction, which is regulated by the electrical activity of the sinoatrial node. The consequent deficient contractility of the atria can form blood pools in the atria, leading to the formation of blood clots. AF can be described by two mainly accepted mechanisms: the ectopic foci and the circuit re-entry [39]. The former happens when ectopic pacemaker foci appear near the pulmonary veins, and propagate disorganized electrical impulses throughout both atria, leading to their contraction. When reaching the AV node, the electrical impulses can also lead to contraction of the ventricles. The circuit re-entry mechanism describes the deterioration of the structural and biochemical conditions of the atria. Low conductivity areas can delay regular electrical pulses from the

[1] https://physionet.org/content/challenge-2017/

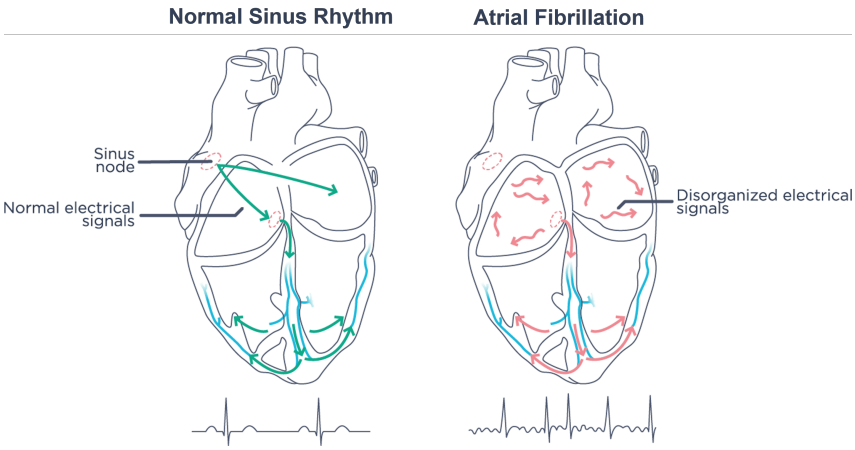


Figure 2.8: Physiological mechanisms in Normal Sinus Rhythm and in Atrial Fibrillation. Extracted from Withings website[1].

sinus node and, by reaching the AV node off time, the atrial myocardium is re-excited. The difference in propagation of the electrical impulses in Normal Sinus Rhythm (NSR) and in AF is shown in Figure 2.8.

AF can be classified by its persistence and medication effectiveness: paroxysmal, when there is a sudden occurrence that disappears within a week; persistent, when it stays for longer than a week and when medication is still effective; and chronic or permanent, when there are structural and biochemical changes that are irreversible and medication is poorly effective. Early detection of AF is of high importance, since the atria's condition can worsen over time and evolve from a paroxysmal state to a permanent one, in a process called atrial fibrillation remodelling.

Since a 12-lead ECG in clinical environment can only offer a brief view of the heart's conduction system (typically around 10 seconds), pervasive monitoring techniques can play a crucial role in patients with rare paroxysmal AF events [40]. Continuous monitoring of such patients can be achieved by using implantable cardiac monitors, adhesive patch monitors, and smartphone-based ECG systems. Several studies report high sensitivity and specificity of such approaches when detecting AF [40].

2.5 Artificial Neural Networks

Artificial neural networks, from now on just called ANNs, are computational models inspired by the biological structure of the central nervous system. This allows ANNs to solve very complex problems of non-linear, multivariate, and/or stochastic nature [41]. The basic unit of a ANN is an artificial neuron, that receives a set of input values $\bar{x} \in \mathbb{R}^n$ and outputs a real value $\hat{y} \in \mathbb{R}$. The neuron first weights each input value through a linear operation:

$$\bar{z} = \bar{w}^T \bar{x} + b \tag{2.1}$$

where $\bar{w} \in \mathbb{R}^n$ is a real-valued vector called weight-vector, and $b \in \mathbb{R}$ is a scalar called bias, and then, using an activation function f_a, $\bar{z}$ is mapped into another subset to limit the value range of the output:

[1] https://www.withings.com/us/en/health-insights/about-afib

$$\hat{y} = f_a(\bar{z}) \tag{2.2}$$

Using the artificial neurons as the building blocks of a ANN, multiple architectures are possible. The simplest one, called the perceptron, corresponds to a single layer of linear input neurons that is followed by a block that applies the unit step function u to the sum of its inputs (Figure 2.9), that is:

$$\hat{y_i} = \mathrm{u}\left(\bar{w}^T \bar{x}_i + b\right) \tag{2.3}$$

where $\bar{w}, \bar{x}_i \in \mathbb{R}^n$ and $\hat{y_i} \in \{0, 1\}$. To be able to solve problems, the perceptron must be trained with data for which one already knows the outputs y_i. Using a random initialization for $\bar{w}$ and b, the perceptron outputs a prediction. If the prediction is wrong, an update to the weights in $\bar{w}$ must be done. To achieve this, a loss function L quantifies the error of the sample, and the respective partial derivatives with respect to w_j are computed to minimize the loss [42]. Using the Mean Squared Error (MSE) as the loss function:

$$L\left(\bar{x}_i, y_i; \bar{w}, b\right) = \frac{1}{2}\left(\bar{w}^T \bar{x}_i + b - y_i\right)^2 = \frac{1}{2}\left(w_1 x_i^{(1)} + w_2 x_i^{(2)} + \ldots + w_n x_i^{(n)} + b - y_i\right) \tag{2.4}$$

the partial derivatives then become:

$$\frac{\partial L}{\partial w_j} = \left(w_j x_i^{(j)} - y_i\right) x_i^{(j)} \tag{2.5}$$

The weight-updating learning process at the $t+1$ iteration can be summarized in:

$$w_i^{(t+1)} = \begin{cases} w_i^{(t)} - \eta\left(w_j x_i^{(j)} - y_i\right) x_i^{(j)} & \text{if } \mathrm{u}\left(\bar{w}^T \bar{x}_i\right) \neq y_i \\ w_i^{(t)} \text{ otherwise} \end{cases} \tag{2.6}$$

where a learning rate η is defined to control the weight updates. This process is repeated until the desired performance is achieved.

However, the perceptron model can only deal with linearly separable problems [41]. To map non-linear dependencies, a more complex ANN is needed using, for instance, the Multilayer Perceptron (MLP), under the condition that it uses at least one non-linear activation function. The MLP consists of a perceptron with multiple stacked layers, called hidden layers, which are fully connected, meaning that each node of the input and hidden layers is connected to each node of the following, as in Figure 2.10. The corresponding weight-vectors can be organized using a matrix formulation for each layer. Considering an input layer with n nodes, a single hidden layer with m nodes, and an output layer with k nodes, the weights can be organized in matrix $W \in \mathbb{R}^{n \times m}$ for the input layer, and $H \in \mathbb{R}^{m \times k}$ for the hidden layer. The MLP then applies the following transformations:

$$\begin{cases} \bar{z} = f_h\left(W^T \bar{x} + \bar{b}\right) \\ \bar{y} = f_a\left(H^T \bar{z} + \bar{c}\right) \end{cases} \tag{2.7}$$

where $\bar{b} \in \mathbb{R}^{m \times 1}$ and $\bar{c} \in \mathbb{R}^{k \times 1}$ are the biases of the input and hidden layers, respectively, and f_h and f_a

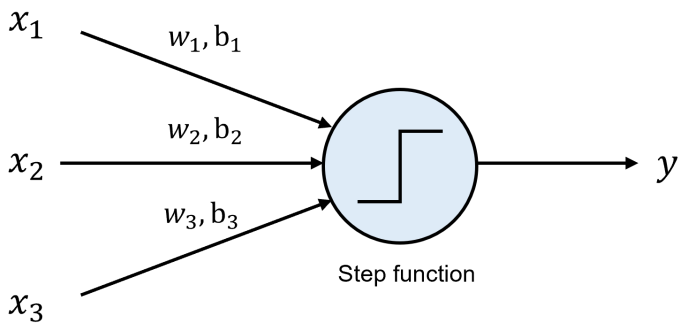


Figure 2.9: Diagram representing the perceptron model with three inputs.

the activation functions.

To train such architecture, the backpropagation algorithm is used. Considering a parametrized network described by $\bar{\theta}$, the goal is to find the optimal set of parameters that minimize the cost function[1] C, that is:

$$\bar{\theta}_{\text{opt}} = \text{argmin}_{\bar{\theta}} C(\bar{\theta}) \tag{2.8}$$

To achieve this, by computing the negative gradient of the loss function L associated with one sample, the weights will be updated in the direction of the closest minimum [42, 41]. Starting from the output layer with weights $\bar{h}_{ij}$, the gradient of the loss function associated with the output $\hat{y}$ can be written using the chain rule:

$$\frac{\partial L}{\partial h_{ij}} = \frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial q_1}\frac{\partial q_1}{\partial h_{ij}} = \frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial q_1}\bar{z}_j = \delta_i \bar{z}_j \tag{2.9}$$

where q_1 is a general argument of the output activation function. Since $\bar{z}_j$ depends on the weights w_{pj} of the matrix W (Eq. 2.7), the gradient of the loss function can be fully defined as:

$$\frac{\partial L}{\partial w_{pj}} = \frac{\partial L}{\partial \hat{y}_i}\frac{\partial \hat{y}_i}{\partial q_1}\frac{\partial q_1}{\partial \bar{z}_j}\frac{\partial \bar{z}_j}{\partial q_2}\frac{\partial q_2}{\partial w_{pj}} = \delta_i h_{ji}\frac{\partial \bar{z}_j}{\partial t_2}\bar{x}_p \tag{2.10}$$

where q_2 is a general argument of the hidden activation function.

After computing the gradients, the cost function can be minimized towards its minimum at the step defined by the learning rate. However, to achieve faster convergence, the weights are often updated after the ANN has seen multiple samples or a batch. This method is called the Stochastic Gradient Descent (SDG) [42].

Because, in some cases, this method can lead to the learning process being stuck with sub-optimal parameters, various enhancements or optimizations have been developed to achieve better results, including the SDG with momentum, Adam, AdaGrad, AdaDelta, among other optimizers [42].

[1] The cost function is the average loss of the training dataset.

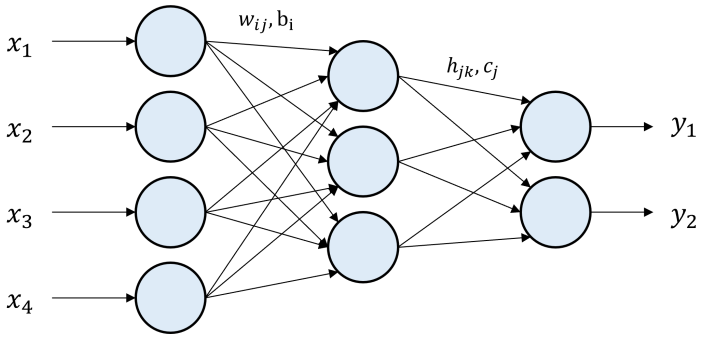


Figure 2.10: Diagram representing the Multilayer Perceptron model with 4 inputs, a single hidden layer and 2 output nodes.

2.6 Autoencoders

An AE is an unsupervised learning model based on ANNs, which is divided into two elements: an encoder and a decoder. The encoder is responsible for generating a feature vector $\bar{z}_i$ (also called code) from the input $\bar{x}_i$, generally by compression, while the decoder is responsible for reconstructing the model input from the feature vector (Figure 2.11). The standard (undercomplete) AE model achieves compression and reconstruction by, respectively, reducing and increasing the number of nodes layer by layer, often symmetrically. Following the formulation of Bonaccorso [42], the encoder can be described as a parametrized function $e(\cdot)$ with parameters $\bar{\theta}_e$:

$$\bar{z}_i = e\left(\bar{x}_i; \bar{\theta}_e\right) \tag{2.11}$$

and the decoder can be described as a parametrized function $d(\cdot)$ of parameters $\bar{\theta}_d$:

$$\hat{x}_i = d\left(\bar{z}_i; \bar{\theta}_d\right) \tag{2.12}$$

where $\hat{x}_i$ is an estimation of the input $\bar{x}_i$.

As in any standard machine learning algorithm, the goal is to minimize a predefined cost function C. In the case of AEs, the cost function is usually proportional to the reconstruction error using, for instance, the MSE between the input $\bar{x}_i \in X$ and output $\hat{x}_i$:

$$C\left(X; \bar{\theta}_e, \bar{\theta}_d\right) = \frac{1}{M}\sum_{i=1}^{M}\left\|\bar{x}_i - \hat{x}_i\right\|^2 = \frac{1}{M}\sum_{i=1}^{M}\left\|\bar{x}_i - d\left(e\left(\bar{x}_i; \bar{\theta}_e\right); \bar{\theta}_d\right)\right\|^2 \tag{2.13}$$

where M is the cardinality of the dataset X.

In a probabilistic approach, considering the inputs samples x_i taken from a probabilistic process $p(X)$, then, the goal of an AE is to find a parametric distribution $q(\cdot)$ that minimizes the Kullback-Leibler divergence D_{KL} between $p(X)$ and $q(\cdot)$. That is:

$$D_{KL}(p\|q) = \sum_i p\left(\bar{x}_i\right)\log\frac{p\left(\bar{x}_i\right)}{q\left(d\left(e\left(\bar{x}_i; \bar{\theta}_e\right); \bar{\theta}_d\right) \mid \bar{x}_i\right)} \tag{2.14}$$

By enforcing the input to be compressed into a latent representation, and by using a cost function that

favors the output to be as close as possible to the input, the feature vector within the code should hold relevant information about the data's structure, provided that the model converged and that no overfitting occured.

Besides the standard AE, a number of other architectures are possible, including Denoising Autoencoders (DAEs), Sparse Autoencoders (SpAEs) and Variational Autoencoders (VAEs). DAEs aim to reconstruct the original version of an input from a corrupted version. This can be achieved by sampling Gaussian noise or randomly dropout input nodes, which encourage the AE to learn the relevant features of the data [43].

SpAEs take advantage of setting some nodes to zero, since this is not generally achieved by a standard AE, and it can have a positive effect when learning an internal representation. Instead of reducing the number of nodes to achieve an information bottleneck, sparse AEs try to enhance the generalization ability by applying a L1 penalty to the code layer, which means that the penalty equals to the absolute value of the weights [44]. In this case, for a single sample, the loss function $\hat{L}(\cdot)$ becomes:

$$\hat{L}\left(\bar{x}_i; \bar{\theta}_e, \bar{\theta}_d\right) = L\left(\bar{x}_i; \bar{\theta}_e, \bar{\theta}_d\right) + \alpha \left\|\bar{z}_i\right\|_1 \tag{2.15}$$

where the code is now $z_i = (0, 0, z_i^n, ..., 0, z_i^m, ...)$, and α is a hyperparameter that controls sparsity.

VAEs have a different internal representation of the latent code z. Instead of simply generating an encoded version of the input, the latent code is assumed to be a random variable of a distribution. The goal is then to obtain parameters $\bar{\theta_q}$ of a probabilistic encoder $q(\bar{z}|\bar{x}; \bar{\theta_q})$, assumed to be a Gaussian distribution, that minimize the Kullback-Leibler divergence D_{KL} with the posterior distribution $p(\bar{z}|\bar{x}; \bar{\theta})$ [42], that is, minimizing:

$$D_{KL}\left(q\left(\bar{z} \mid \bar{x}; \bar{\theta}_q\right) || p(\bar{z} \mid \bar{x}; \bar{\theta})\right) = \sum_z q\left(\bar{z} \mid \bar{x}; \bar{\theta}_q\right) \log \frac{q\left(\bar{z} \mid \bar{x}; \bar{\theta}_q\right)}{p(\bar{z} \mid \bar{x}; \bar{\theta})} \tag{2.16}$$

The covariance matrix and mean of the distributions are therefore regularized by the D_{KL}. However, the posterior distribution $p(\bar{z}|\bar{x}; \bar{\theta})$ is unknown. To overcome this, a mathematical manipulation of the previous equation leads to the definition of a quantity called the negative of the evidence lower bound (ELBO) that is going to be the target of the optimization, i.e., the loss function:

$$E_z[\log p(\bar{x} \mid \bar{z}; \bar{\theta})] - D_{KL}\left(q\left(\bar{z} \mid \bar{x}; \bar{\theta}_q\right) || p(\bar{z}; \bar{\theta})\right) \tag{2.17}$$

This approach makes VAEs a generative model, since the output is not in the dataset, but instead it is a generated version from a random sampling of the encoded feature distributions.

Other types of AE are possible such as deep, contractive and convolutional AEs. For further details, the reader is referred to [42, 45].

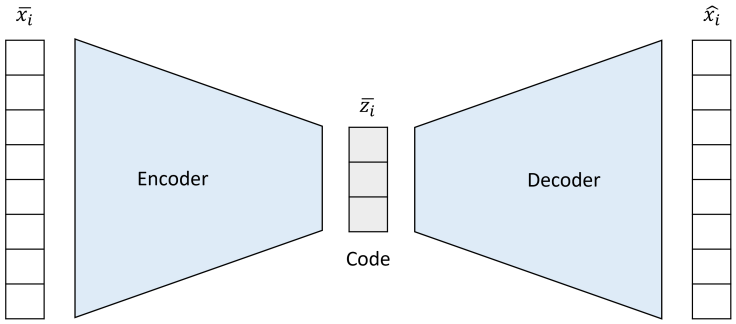


Figure 2.11: Schematic representation of an Autoencoder model. It is composed of an encoder, responsible for compressing the data into a latent representation called code, and a decoder, which aims to reconstruct the input from the code.

2.7 Classification Metrics

Classification metrics evaluate the performance of a specific characteristic of a classifier, and they are an essential tool to compare different classifiers with the same task. In the case of a binary classification problem, two distinct classes separate the data: one called positive and the other called negative. After training the classifier, a confusion matrix can be used to assess the number of correct and wrong predictions of the data used for testing, as in Figure 2.12. The True Positive (TP) and True Negative (TN) represent the number of positive and negative samples that were correctly classified, respectively. In contrast, the False Positive (FP) and False Negative (FN) represent the number of positive and negative samples wrongly classified, respectively [46].

Combining these values, several metrics, called threshold metrics, can be defined [46]. Accuracy measures the overall ability of the classifier to classify both positive and negative instances correctly (Eq. 2.18). Precision measures the ratio between the TP instances and the total number of predicted positive instances (Eq. 2.19). Recall (or sensitivity) measures the ratio between the TP instances and the total number of positive instances (Eq. 2.20). Specificity measures the ratio between TN and the total number of negative instances (Eq. 2.21); and the F1-score is the harmonic mean between precision and recall (Eq. 2.22), which is particularly useful when there is data imbalance.

	Actual Positive Class	Actual Negative Class
Predicted Positive Class	True Positive (TP)	False Negative (FN)
Predicted Negative Class	False Positive (FP)	True Negative (TN)

Figure 2.12: Confusion matrix of a binary classification problem. Adapted from [46].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.18}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.19}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.20}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{2.21}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.22}$$

Another tool to evaluate a classifier is the Receiver Operating Characteristic (ROC) curve, and the respective area under the curve (AUC), depicted in Figure 2.13. Most classifiers have parameters that can be tuned, thus giving different predictions of the data. Also, if the classifier's output consists of probability values, the threshold to separate the classes will affect its performance metrics. To evaluate such type of classifier, the ROC curve can be used to show the trade-off between the sensitivity and specificity by setting different parameters or threshold values, and the AUC, because it does not depend on a specific point of the curve, is capable of measuring the ability of the classifier to predict the classes accurately [47].

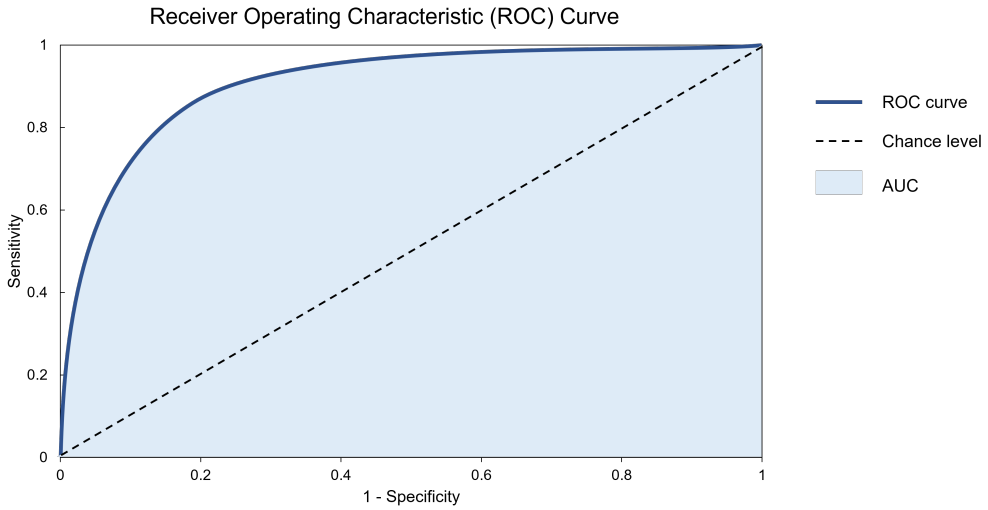


Figure 2.13: The ROC curve corresponds to the plot of sensitivity (or true positive rate) against 1 - specificity (or false positive rate), and the AUC to the corresponding area below the curve. The chance level defines the ROC curve corresponding to random chance classification.

Chapter 3

State of the Art

3.1 Atrial Fibrillation Detection Algorithms

Atrial fibrillation detection algorithms using artificial intelligence techniques have been developed since 1983. In 1992, Janet Slocum and colleagues [48] developed a decision tree approach based on power spectral analysis to detect AF that obtained an F1-score[1] of 91.3%. In 1994, Yang et al. [49] adopted a neural network approach using features based on P-wave and RR-intervals and obtained a 94.9% F1-score. Although limited by data availability and variability, these approaches opened the path for other algorithms to be developed and multiple types of features to be tested in the following decades.

Over the years, researchers built many approaches to detect AF, and, according to a 2021 systematic review from Wesselius et al. [13], they can be grouped into six main categories: rule-based classification, Decision Trees (DTs), regression analysis, k-Nearest Neighbors (k-NNs), Support Vector Machines (SVMs), and ANNs. Early techniques were more focused on rule-based and decision trees methodologies, whereas, between 2016 and 2020, SVMs and ANNs represent almost 50% of all developed algorithms [13] (Figure 3.1). Since these algorithms are more complex and considered black-box models, their accuracy and reliability can be questioned, because there is no clear interpretation of how the algorithms interpret and make use of the data. Thus, efforts to unveil how these algorithms work and perceive information are greatly appreciated by the scientific community and clinicians.

The aforementioned review article collected information regarding 130 AF detection algorithms out of 451 results from a PubMed[2] database search, conducted on the 14th of September 2020 [13]. Non-english articles, unclear algorithm descriptions, algorithms to distinguish different types of AF and algorithms using PPG were some of the excluding criteria. The list of the selected algorithms for the study are available in the review's supplementary material.

Algorithms greatly rely on what information is fed into them and, in the case of AF detection, different ECG feature inputs can be used. Atrial features to detect AF rely on ECG properties inherent to atrial activity, such as the absence of P-waves and/or appearance of f-waves. This analysis can be done in time and frequency domains with the help of statistical metrics. However, since these waves are of low

[1] F1-score is an accuracy measure defined as the harmonic mean between sensitivity and precision.
[2] https://pubmed.ncbi.nlm.nih.gov/

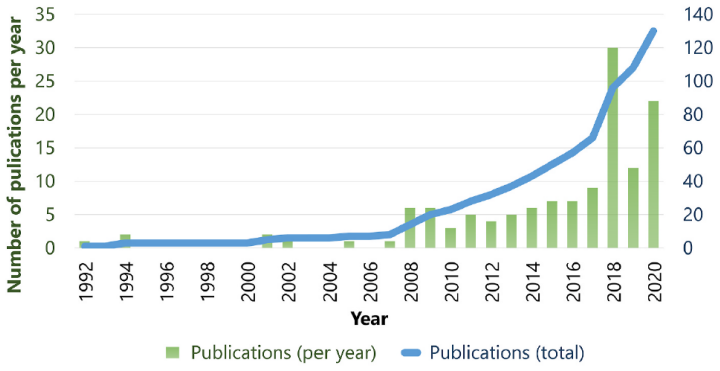
amplitude, they are more prone to be affected by noise and other artefacts, complicating the design of algorithms that use these features. For that reason, although AF reflects changes of atrial activity, atrial features are not widely used. The F1-score for an algorithm of such category developed by Christov et al. [50] was estimated at 83.8% [13].

Ventricular features are based on ventricular activity by obtaining information regarding the QRS complexes, which are generally more pronounced than P-waves and f-waves. The R-peaks are the main focus of such features because the presence of irregularities in RR-intervals are a key sign to diagnose AF. Examples of ventricular features are standard deviation (SD), coefficient of variance, root mean square of successive differences (RMSSD), Poincaré plot measurements, sample entropy, Shannon entropy, turning point ratio (TPR) and Lyapunov exponents. The review made by Wesselius et al. [13] reported a median F1-score of 96.9% from 38 algorithms that only use ventricular features and represented the highest score from the other types and combinations of features (Table 3.1). A brief description of some main ventricular features is provided in Table 3.2.

Another frequent type of features to be extracted are signal properties, which include a series of measures such as basic statistics, signal power, kurtosis and derivatives. More complex signal features involve performing power spectral analysis, phase space analysis, computing wavelet transform, and measuring signal quality (e.g., correlation with a template). The same review reports a median F1-score of 95.2% from a total of 34 algorithms that only rely on signal features [13]. From within this class, the rule-based algorithm proposed by Queiroz et al. [51] is estimated to have a 100% F1-score. The algorithm only uses kurtosis as the main statistical feature to distinguish normal from AF records in a windowed series of RR-intervals.

Table 3.1: Median F1-score by type of feature groups. It includes atrial, ventricular and signal features, as well as the combinations between them. Extracted from [13].

FEATURE GROUPS	NUMBER OF ALGORITHMS	MEDIAN F1-SCORE
Atrial features	1	83.8%
Ventricular features	38	96.9%
Signal features	34	95.2%
Atrial + Ventricular features	10	85.6%
Atrial + Signal features	1	88.9%
Ventricular + Signal features	6	91.1%
Atrial + Ventricular + Signal features	13	81.0%
Overall	**103**	**94.0%**



(a)

(b)

Figure 3.1: (a) Number of publications per year related to Atrial Fibrillation detection algorithms and (b) the corresponding distribution by type of algorithm. Extracted from [13].

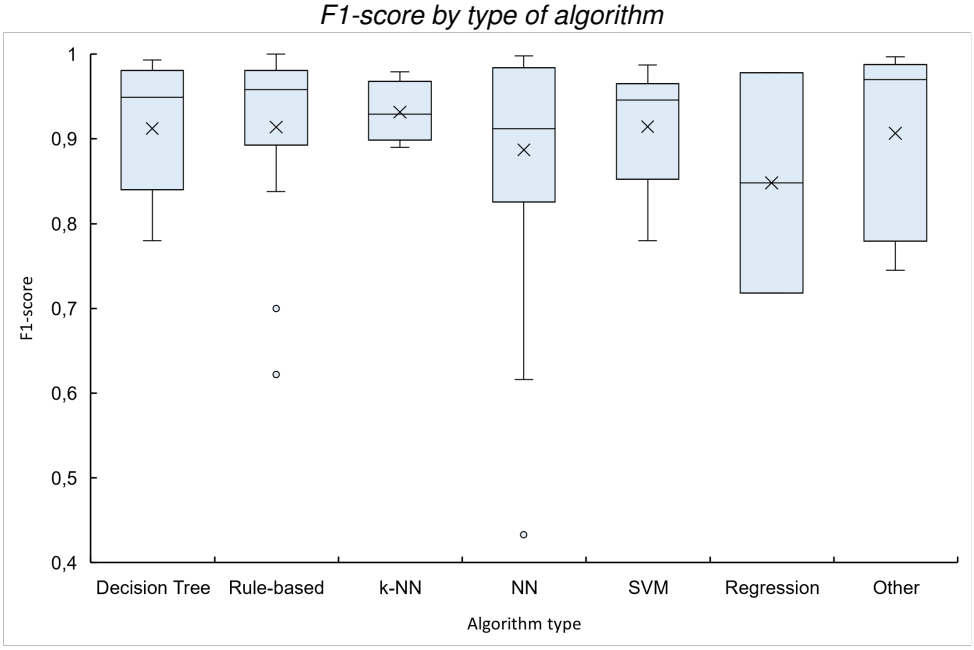


Figure 3.2: F1-score distributions of the algorithms reviewed by Wesselius et al. [13] sorted by algorithm type, which include decision tree (N=19), rule-based (N=16), k-Nearest Neighbor (k-NN) (N=4), Artificial Neural Network (ANN) (N=36), Support Vector Machine (SVM) (N=17), regression (N=2) and other (N=9). The data were obtained by the available F1-scores in the review's supplementary material of [13]. The $\times$ marks the average values.

Table 3.2: Brief description of relevant ventricular features used to detect AF from the ECG signal.

RR-INTERVAL feature	FORMULA	DESCRIPTION
Standard Deviation (SD)	$\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(RR_i - \overline{RR})^2}$	Measure of variation. However, higher HRs lead to lower variance in RR-intervals, thus the use of coefficient of variation is advised [52].
Coefficient of Variation (CV)	$SD/\overline{RR}$	Differences in HR are corrected by dividing SD by the mean HR [52].
Root Mean Square of Successive Differences (RMSSD)	$\sqrt{\frac{\sum_{i=1}^{N-1}(RR_i - RR_{i+1})^2}{N-1}}$	Reflects the average change in successive RR-intervals. It is the main measure of HRV.
Poincaré plot	Plot RR_{n+1} against RR_n	By fitting an ellipse, the area (S), width (SD1) and length (SD2) can be obtained. SD1/SD2 measures the unpredictability of the RR time series [53]. It is insensitive to changes in trends in the RR-intervals.
Sample Entropy (SampEn)	$-log\frac{\sum_{i=1}^{N-l} N_i(l+1,r)}{\sum_{i=1}^{N-l} N_i(l,r)}$	Measures the "probability that two matching RR-interval series will continue to match at the next RR-interval" [13]. It is a modified version of the approximate entropy (ApEn), adapted to physiological signals.
Shannon Entropy (ShEn)	$-\sum_{i=1}^{N_{bin}} p_i \frac{log(p_i)}{log(1/N_{bin})}$	Measures the uncertainty of RR-intervals [54]. It requires an estimation of the probability density function.
Turning Point Ratio (TPR)	$\#TP/N$	Non-parametric test to determine if a RR time series is random, by comparing each RR value with its neighbors [54]. If the series is random, it has the expected number of turning points (local maxima and minima).
Lyapunov Exponent (LE)	$\lambda_i = \lim_{t\to\infty} \frac{1}{t} log\left(\frac{d_i(t)}{d_i(0)}\right)$	It is a measure of the variation between the trajectories of two points along time [13, 55]. If the RR-intervals are irregular, the mean distances are greater as well as LEs.

RR = RR-interval
N = RR series length
$N_i(l,r)$ = number of l-length segments found at a distance smaller than the threshold r
N_{bin} = number of bins (histogram)
p_i = probability of event i
$d_i(t)$ = distance between trajectories at time t
TP = turning point
HR = heart rate
HRV = heart rate variability

3.2 Autoencoders for Atrial Fibrillation Detection

Many ANN-based algorithms to detect AF are built with deep architectures such as Convolutional Neural Networks (CNNs) [56, 57, 58] and Recurrent Neural Networks (RNNs) [59, 60], which are known for their ability to capture complex patterns in the data. However, very few approaches use simple architectures such as the AEs. The performances of two AE-based approaches for AF detection are detailed in Table 3.3.

In 2016, Yuan et al. [61] developed an approach for AF detection from ECG records using a stacked SpAE based on 84 selected features extracted from the RR-intervals and P-wave measurements of a 10-second window. The AE used to achieve data compression had 84 input nodes and 2 hidden layers with 300 nodes each[1]. AF detection was made by stacking a softmax classifier to the extracted features of the AE. Using ECG records from the MIT-BIH databases[2], the model first achieved a detection accuracy of 75.6%, and, after fine-tuning the model, a 98.3% accuracy was reported.

A similar approach was followed by Chen L. and Ying H. in 2019 [62], where a stacked SpAE receives 19 features extracted from the ECG records, including statistical measures, parameters from the Hilbert-Huang transform, and wavelet decomposition features. After training, the AE is then coupled to a softmax classifier to detect AF; a 96% accuracy was achieved.

Another AE-based algorithm to detect AF was developed by Cortez et al. [63]. This approach is significantly different from the previous ones because it uses a recurrent VAE, based on bidirectional Long Short-Term Memory (LSTM) networks[3], to extract features that allow not only to detect AF but also to evaluate its progression from a paroxysmal condition to a permanent one. The classifier consists of a single ANN layer and a softmax layer that are connected to the VAE's latent space. However, instead of first training the VAE and then using the resulting features to classify the records, the classifier is used to give feedback to the network in a semi-supervised manner. This way, the latent space is regularized (or shaped) by both the Kullback-Leibler divergence and the class labels. A graphic projection of the latent space is used to evaluate the progression of AF. Using synthetic data, an accuracy of 97.1% was obtained when assessing different stages of AF progression.

Table 3.3: Performance metrics of two Autoencoder-based models for Atrial Fibrillation detection.

Algorithm	AE type	Classifier	Database	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Yuan et al. [61] (2016)	SpAE	Softmax	MIT-BIH	98.3	96.6	97.7†	97.1†
Chen et al. [62] (2019)	SpAE	Softmax	N/A	96.0	93.8	90.0	91.8†

†Computed from the available information.
N/A = not available.

[1] An autoencoder with more nodes in a hidden layer than the input is called overcomplete.
[2] https://ecg.mit.edu/
[3] A Long Short-Term Memory network is a type of Recurrent Neural Networks.

3.3 ECG Datasets for Atrial Fibrillation Detection

The increased interest in AF detection in recent years led to the development of publicly available databases to train and test new models. The characteristics of the main databases for AF detection are presented in Table 3.4.

Some of the oldest databases date back to the 1980s, where three databases based on two-lead ECG records were made available. The MIT-BIH Arrhythmia Database (MITDB)[1], created by the Boston's Beth Israel Hospital laboratories and the Massachusetts Institute of Technology, was purposely made to test arrhythmia detectors and it consists of 30-minute-records of 48 subjects. It contains beat-by-beat annotations of normal and abnormal waveforms, rhythm annotations of several heart conditions, and signal quality remarks. The MIT-BIH Atrial Fibrillation Database (AFDB)[1], also made with the same aforementioned entities, has 10-hour-records of 25 participants with AF (mainly paroxysmal). Annotations of NSR, AF, atrial flutter and AV junctional rhythms are available. The American Heart Association Database (AHADB)[2] also contains several rhythm annotations for 154 3-hour records.

More recent databases include the Long Term Atrial Fibrillation Database (LTAFDB)[3] with 24-hour holter records of 84 subjects with parosysmal or persistent AF, released in 2007, and the Chinese Cardiovascular Disease Database (CCDD)[4], with 179 130 12-lead records.

The AF database from the Computing in Cardiology Challenge 2017 (CinC2017)[5] is considered to be one of the main sources of new publications related with AF detection algorithms (Figure 3.1(a)) [13]. The CinC2017 database has 12 186 short single-lead records, varying from 9 to 60 seconds. Record labels include NSR, AF, other rhythms, and noisy acquisitions (Figure 3.3). The records were obtained with a Left Arm – Right Arm lead configuration, equivalent to a Lead I, using AliveCor's[6] single channel ECG devices, including the AliveCor® KardiaMobile. The acquisitions were made using a 300 Hz sampling frequency with 16-bit resolution over a ± 5 mV dynamic range and a $0.5 - 40$ Hz bandwidth interval [64].

Taking into account the nature and quality of most wearable and invisible ECG signals and their acquisition methodologies, the CinC2017 database was considered the most appropriate for this work, taking into account the intended use case. It is the only one optimized for single-lead acquisition, recorded with a mobile-based device using current technology, and exclusively designed for AF detection. The AHADB, AFDB, and MITDB databases are acquired with hospital-grade systems and do not reflect the current state of the art in ECG recording technologies. The training set of the CinC2017 database was used to extract the ECG segments, and to train and evaluate the models. The set contains 8528 ECG records, being 5154 of NSR and 771 in AF. A detailed description of the training data is provided in Table 3.5.

[1] http://ecg.mit.edu/
[2] https://www.ecri.org/american-heart-association-ecg-database-usb
[3] https://physionet.org/content/ltafdb/
[4] http://ecgdb.com/
[5] https://physionet.org/content/challenge-2017/
[6] https://www.kardia.com/

Table 3.4: Description of publicly available databases for atrial fibrillation detection.

Database (Year)	Number of records	Lead system	Duration per record	Sampling rate (Hz)	ADC Resolution	Dynamic Range	Number of classes
CinC2017 (2017)	12,186	Single-lead	9 – 60 s	300	16-bit	$\pm$ 5 mV	4
CCDD (2010)	179,130	12-lead	20 min	500	N/A	N/A	N/A
LTAFDB (2007)	84	Two-lead	24 – 25 h	128	12-bit	$\pm$ 10 mV	9
AHADB (1985)	154	Two-lead	3 h	250	12-bit	$\pm$ 5 mV	8
AFDB (1983)	25	Two-lead	10 h	250	12-bit	$\pm$ 10 mV	4
MITDB (1980)	48	Two-lead	30 min	360	11-bit	$\pm$ 5 mV	15

CinC2017 = Computing in Cardiology Challenge 2017
CCDD = Chinese Cardiovascular Disease Database
LTAFDB = Long Term Atrial Fibrillation Database
AFDB = MIT-BIH Atrial Fibrillation Database
AHADB = American Heart Association Database
MITDB = MIT-BIH Arrhythmia Database
N/A = Not Available

Table 3.5: Description of the training set of Computing in Cardiology Challenge 2017.

Type	# Recordings	Time length (s)				
		Mean	SD	Max.	Median	Min.
Normal	5154	31.9	10.0	61.0	30	9.0
AF	771	31.6	12.5	60	30	10.0
Other rhythm	2557	34.1	11.8	60.9	30	9.1
Noisy	46	27.1	9.0	60	30	10.2
Total	**8528**	**32.5**	**10.9**	**61.0**	**30**	**9.0**

3.4 ECG Monitoring Systems

ECG monitoring systems became an essential supporting tool to assess CVDs, mainly because evidence shows that continuous monitoring of the heart's condition can significantly improve CVD management, either for prevention or for disease monitoring [65, 66, 4]. They are now available in hospital, home, ambulatory and remote settings, and in the most diversified ways in terms of acquisition platforms, pre-processing and processing algorithms, storage platforms, interface choices, and data modelling and analytics tools [4].

A 2020 review made by Serhani et al. [4] described the current ECG monitoring systems taking into account the great diversity of tools and platforms used across more than 600 publications of different cardiac monitoring systems.

A basic and complete ECG monitoring system can be described by its data flow, as depicted in

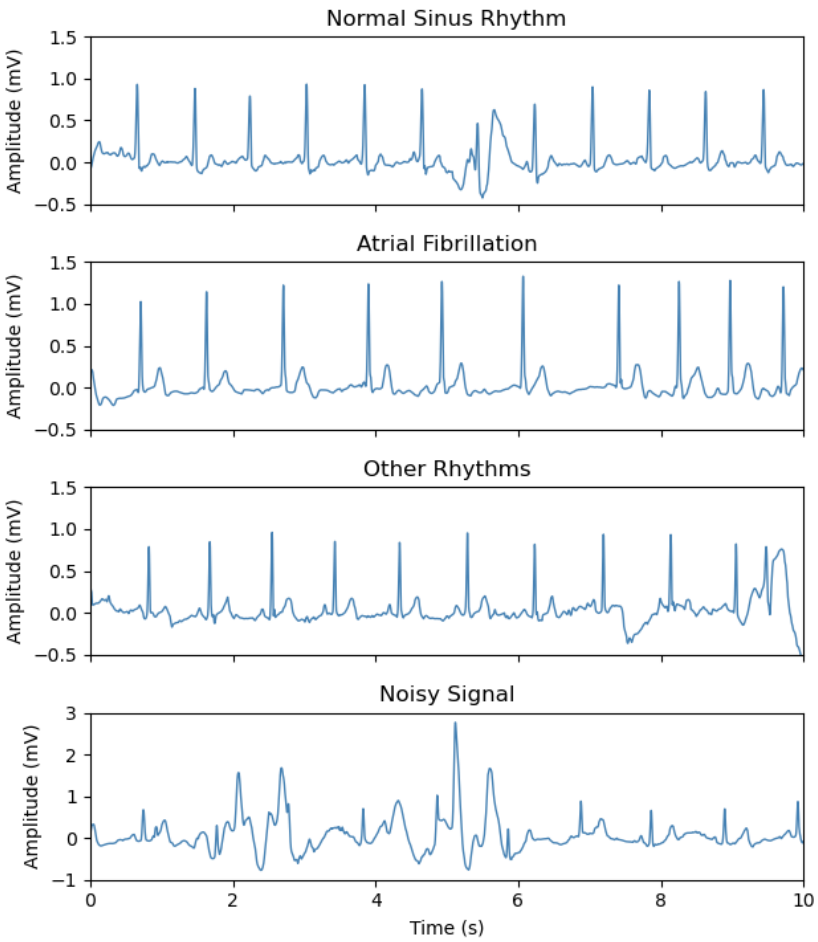


Figure 3.3: Examples of Normal Sinus Rhythm, Atrial Fibrillation, other rhythms and noisy acquisitions in the Computing in Cardiology Challenge 2017 dataset.

Figure 3.4. It starts by defining how the data will be acquired: the type of sensors, their placement, and the technology to acquire, store and share the data. For instance, choosing wireless over wired sensors will greatly define the hardware usability. ECG recordings can be made using standard ECG sensors or adapting electrodes into mobile and wearable designs (Figure 3.5). Data transmission between the sensors and the main processing units is typically performed using WiFi, Bluetooth and other radio frequency communication protocols (Figure 3.5). Also the analog front-end (AFE) component can be used to filter noise and amplify signals.

After defining the data acquisition platform, preprocessing steps and storage, the target features to be analysed have to be selected (Figure 3.4). As already presented in Section 3.1, atrial, ventricular and signal features can be extracted from ECG signals and used for several purposes. As categorized by [4], current monitoring systems can be directed towards diagnosis, activities and prognosis. For example, it is possible to develop algorithms that detect and assess CVDs such as arrhythmias [65, 67], AF [68] and other abnormalities. Evaluation of mood states [69], sleep apnea [66], and applications towards sports [70], driving [71, 72], and other daily activities are possible as well. More applications and references are available in [4].

Processing tools are then responsible for properly feeding algorithms with meaningful data to train

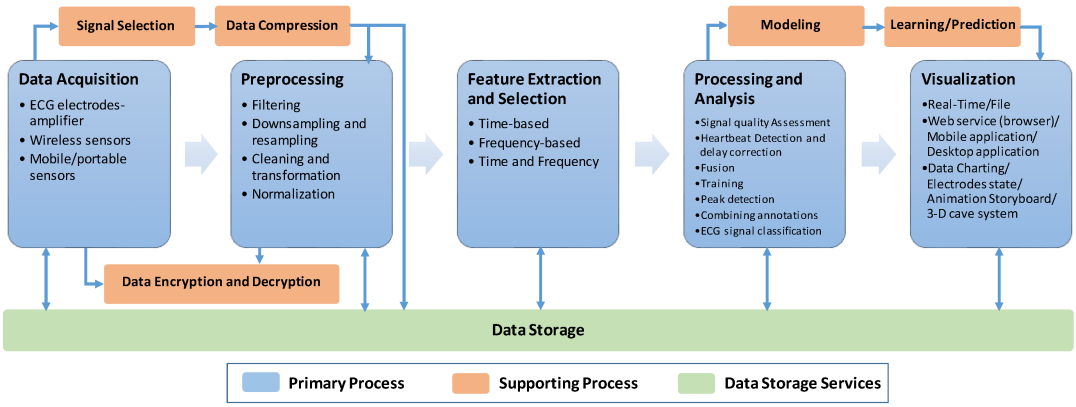


Figure 3.4: ECG monitoring systems key processes. Extracted from [4].

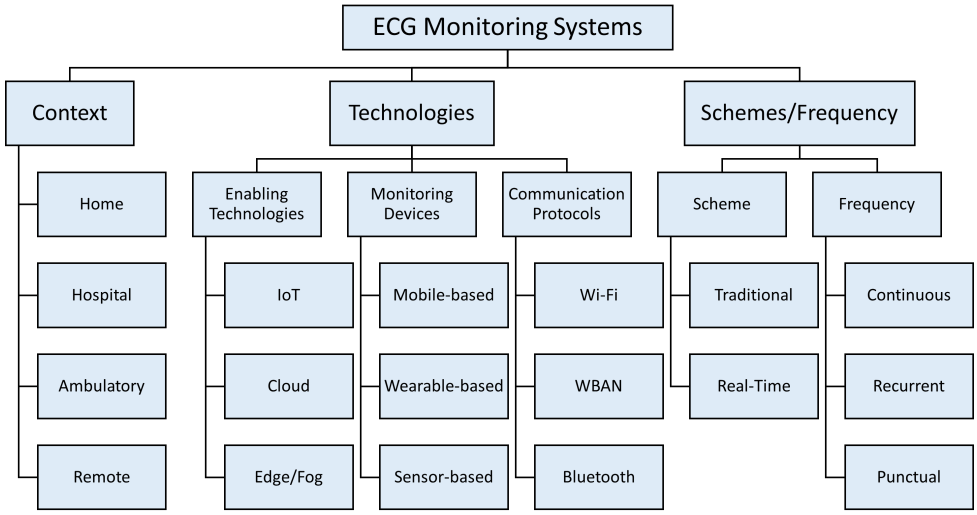


Figure 3.5: ECG monitoring systems clustering. Adapted from [4].

and test them using (Figure 3.4), for example, machine learning and ANN techniques. After validating the algorithm's ability to classify or provide metrics regarding the ECG recordings, an interface to report the results to the user has to be developed.

Two other important factors that shape the computational design of ECG monitoring systems are the type of physical setup and the rate at which the data is processed for evaluation. The screening setup can be stationary ("traditional") when a patient is directed towards the device (mainly for diagnosis and prognosis), or it can be deployed in real-time, when ECG monitoring is integrated with the patient's daily activities and a wide range of applications is possible [4]. Moreover, the data processing frequency can define if the system provides continuous, recurrent/scheduled or punctual monitoring (Figure 3.5) [4].

Chapter 4

Proposed Approach

In this chapter, the process of designing and choosing a classification model to detect AF based on AEs is thoroughly explained. The rationale behind the approach, the choice of the ECG dataset to be used, and the description of the proposed AEs and classifiers to adress the problem are presented. The results (Chapter 5) will follow a similar structure.

4.1 Rationale

As already discussed, AEs are an unsupervised learning model in which it is possible to retrieve a meaningful and compact representation of a signal. When training an AE neural network with ECG signal waveforms, it is expected that the network will be able to reconstruct unseen waveforms with excellent fidelity, since the physiological processes behind them are the same. However, the acquisition conditions of the data must be as similar as possible between acquisitions because different systematic errors or artifacts caused by different acquisition systems, or even different preprocessing steps, can affect the signal quality, waveform, and other relevant properties.

The interest in using AEs to detect AF arose because few approaches are using them; thus, its use is not well documented. Also, the great diversity of AE possible architectures allows exploring and creating innovative solutions to a real-world challenge, for which there is no current gold standard approach for mainstream use. Also, besides anomaly detection, AEs can be used for classification by feeding classifiers with their compact representations by dimensionality reduction.

In today's implementations of heart monitoring technologies and algorithms, responsiveness is highly valued, and systems with high computational speeds using simple algorithmic approaches are preferred. This is the case of AEs since there are reported real-time implementations with satisfactory performance. Also, when designing algorithms to detect AF and other anomalies, there is a great investment in choosing the features that allow the best classification performance. Many approaches use dozens of features and apply them in artificial intelligence algorithms without a proper rationale behind their choices. Many algorithms focus only on obtaining high classification scores without considering the real-world constraints of monitoring systems. Because of the nature of AEs, this is not an issue; they do not require

feature engineering, and are flexible in terms of complexity (e.g., number of nodes, layers, and connections), thus requiring fewer computation resources for possible real-time implementation.

In ECG monitoring systems, preprocessing steps often include beat extraction using segmentation techniques; thus ECG waveform fiducials and several metrics can be used to classify beats. However, most AF detection systems do not use segments, instead they rely on information extracted from ECG records containing several beats, namely because they focus on ventricular activity (e.g., RR-intervals). Also, although atrial activity algorithms do not have such high performance as the ventricular-based ones (see Section 3.1), a recent analysis by Tuboli et al. [73] pointed out that atrial activity metrics can increase the algorithms' specificity when detecting AF. This way, using ECG segments instead of long records, to detect AF was the first approach to be tested, turning the problem into a beat classification one. Since the AE receives the ECG waveform as its input, it can be called a morphological AE.

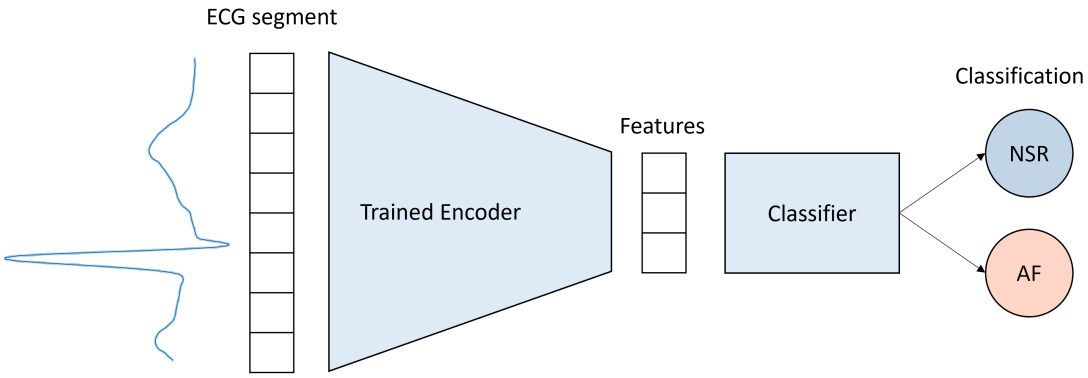


Figure 4.1: Diagram showing the proposed Atrial Fibrillation detector, by stacking a trained encoder with a classifier.

4.2 Data Preprocessing

To train the AEs with ECG segments, the records from the CinC2017 database were first preprocessed using the Python toolbox BioSPPy[1]. The data were filtered using a 90th-order high-pass Finite Impulse Response (FIR) filter with a cutoff frequency of 0.5 Hz, and the R-peak detector proposed by Hamilton [74] was used.

Because some of the records were inverted in amplitude, an approach to correct them was implemented. To avoid complex analyses of the signals, and because there is no standard approach to identify inverted ECG signals in the literature, the median of the R-peak amplitudes was chosen as the main criterion. The inverted records were identified by having a lower median R-peak amplitude than their correspondent inverted versions, that is, the correct ones. In fact, the peaks of the inverted signals are S-waves, with lower amplitude than the R-peaks. An example of such correction is shown in Figure 4.2.

After correcting the inverted ECG records, the signals were then clipped from -200 to 400 milliseconds around the R-peaks, as defined by the toolbox and used by a real-time implementation in [75].

[1] https://github.com/PIA-Group/BioSPPy

Since the sampling frequency of the records is 300 Hz, the segments have 180 samples each. However, because of the nature of AF, where the RR-intervals are irregular, second R-peaks could appear in the segments, if the RR-interval was smaller than 600 milliseconds. To prevent the AE from being affected by such phenomena, zero padding was applied 50 milliseconds before the second R-peak to cover the correspondent Q-wave as represented in Figure 4.3(a). This approach to segment the signals was preferred because the number of input nodes in an AE must be fixed from the beginning.

Another step to obtain representative segments of an ECG record is to eliminate outliers. To discard anomalous waveforms that do not follow the same morphology as the majority, the ECG outlier detection algorithm called DMEAN and described by Lourenço et al. [76] was used. This algorithm "is [an adaptive] distance-based approach which uses a single reference, the mean of the recording session" [76] to identify anomalous waveforms. The cosine distance was chosen to be distance metric. The ability to detect outliers is demonstrated in Figure 4.3(b).

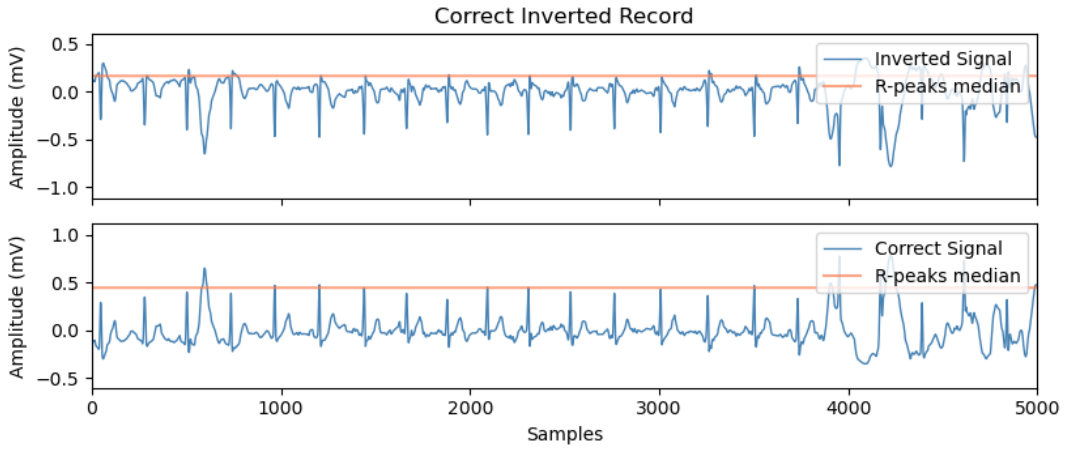
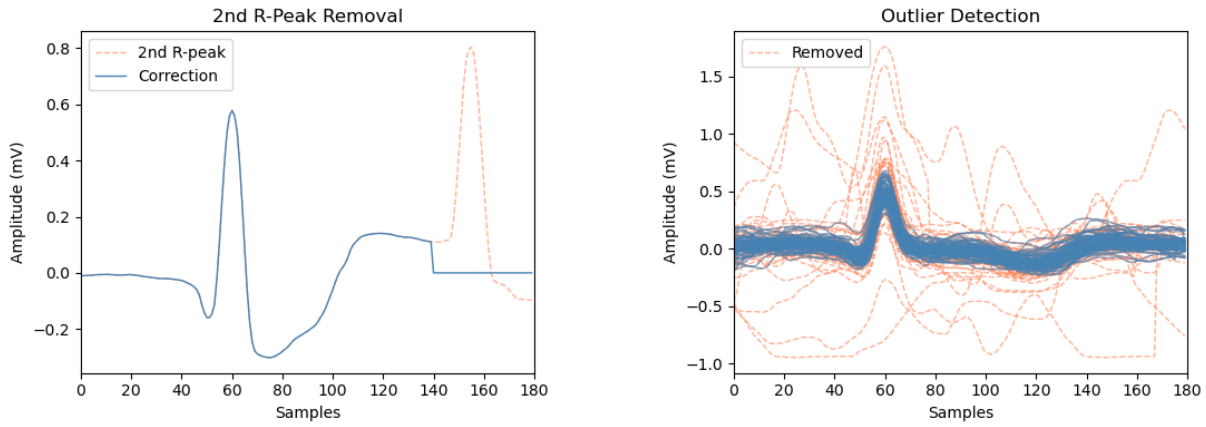


Figure 4.2: Example of an inverted ECG record that was identified by having a lower median of the correspondent peak amplitudes (top) than the inverted version (bottom), which is the correct one. The peaks of the inverted signal are S-waves, and the peaks of the correct signal are the real R-peaks.

Section 3.3 presented the chosen database from the CinC2017 challenge. However, there are 5154 records of NSR (87%) and only 771 of AF (13%), with similar time durations (Table 3.5). Data imbalance, such as this case, can make the models underperform by overlooking the minority class [77]. At the preprocessing level, a common and effective way to obtain a more balanced dataset is to resample the data, by undersampling the majority class instances, oversampling the minority ones, or combining both approaches [77]. Because the NSR and AF records contain several waveforms each, the NSR class was undersampled. Instead of removing records, which are an important source of morphological variability, the number of NSR waveforms was decreased. In practice, the NSR waveforms within a fold were first shuffled (to cover waveforms from different subjects) and then limited by the number of AF waveforms, resulting in a 1:1 proportion. The description of the data composition is presented in Table 4.1.

Because the lack of homogeneity can affect the ability of the models to capture patterns in data and can slow down the training process, another standard preprocessing step before training is to rescale



(a) (b)

Figure 4.3: (a) Example of 2nd R-peak removal on a segment using zero-padding. (b) Use of DMEAN to detect outliers in a set of ECG segments from a record – valid segments are depicted in blue, while outliers are displayed in dashed orange curves.

the inputs [44]. The Min-Max normalization[1] was applied, which rescales each data sample x into the 0–1 range according to the following linear transformation:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

Table 4.1: Description of the data distribution used to train and test the models.

	Train		Test	**Total**
	AE	Classifier		
NSR	10668	11295	6052	**28015**
AF	10668	11295	6052	**28015**
Sub-total	21336	22590	12104	**28015**
Total	**43926**		**12104**	**56030**

[1] Using Scikit's MinMaxScaler function

4.3 Autoencoder Models

Since the architecture of an AE is based on ANNs, its design is very versatile. The simplest AE structure contains a single hidden layer, where one needs to define the compression factor to be achieved, since the input and output layers have the same nodes as the data samples. Although there are no strict rules when choosing the number of nodes in the hidden layer, frequent compression factors are 25%, 50%, and 75% [78]. Adding more hidden layers is also a common choice for AEs. In a stacked AE, the number of nodes per layer decreases until the desired compression rate (*i.e.*, the number of features in the code), and it increases using a symmetrical structure. However, experiments have shown that adding more hidden layers does not necessarily increase the algorithm's performance [79, 78].

Aside from choosing an AE structure, inherent parameters of ANN training are still of utmost importance, since they can affect the overall ability of the AE to properly reconstruct the inputs through the choice of the networks' weights and biases. Therefore, the optimizer, the learning rate, the loss function of the network, and the activation function of each layer have to be set. Furthermore, the choice of the batch size (number of samples fed into the network in one training iteration) and the number of epochs (number of times that the whole dataset passes through the network) are also relevant parameters to achieve the best performance. Other ANN architecture choices that affect performance include adding dropout layers to avoid overfitting or even adding noise layers into the AE to promote meaningful compact representations [80].

To study the best approach to reconstruct ECG segments, different symmetric AE structures were tested, varying the compression factor and number of hidden layers. Under adequate learning parameters, different AEs were evaluated based on their capacity to reconstruct the inputs with a high compression factor. The MSE loss was monitored in each AE training, using validation data of subjects not seen by the AE.

The proposed AE models are briefly described in Table 4.2, indicating the AE type, structure, loss function and other details. The code implementations are available in Appendix A. In addition to input and output dense layers with $N\_segment$ nodes, the Standard Autoencoder (SAE) consists of a single hidden layer with $N\_bottleneck$ nodes. The SAE architecture is the basis for the other AE models, having a linear activation function for the output layer and a MSE loss function (i.e., the reconstruction error). The DAE adds a Gaussian noise layer to the input, defined by its standard deviation, and the SpAE model has a L1 regularization in the hidden layer that allows the model to become sparse, as detailed in Section 2.6.

The Robust Autoencoder (RAE) model is similar to a DAE in its ability to resist noise, but instead of adding noise to the inputs, the loss function aims to deal with data outliers. The chosen loss function is the one proposed by Liu et al. called correntropy [81], a similarity measure using a Gaussian kernel. This function has distinct characteristics from MSE that make it useful for non-linear signal processing. The Contractive Autoencoder (CAE) is also a model that aims to extract useful features by its loss function. In CAEs, a penalty term is added to the MSE, which is defined as the Frobenius norm of the Jacobian matrix of the hidden mapping of the AE [82].

The VAE model, already introduced in Section 2.6, has an intermediate layer with $N\_intermediate$ nodes before the bottleneck layer. The bottleneck consists of two steps, one involving two layers to encode the mean and the covariance matrix of the data as Gaussian distributions, and another involving a layer that allows backpropagation of the algorithm, called the reparametrization trick[1].

Table 4.2: Layer structure of the proposed Autoencoders and their loss functions.

AUTOENCODER	LAYER(#NODES, ·)	LOSS FUNCTION
SAE	Dense(N_segment) Dense(N_bottleneck) Dense(N_segment, linear activation)	MSE
DAE	Dense(N_segment) GaussianNoise(SD=0.05) Dense(N_bottleneck, sigmoid activation) Dense(N_segment, linear activation)	MSE
RAE	Dense(N_segment) Dense(N_bottleneck) Dense(N_segment, linear activation)	Correntropy loss
CAE	Dense(N_segment) Dense(N_bottleneck, sigmoid activation) Dense(N_segment, linear activation)	Contractive loss
SpAE	Dense(N_segment) Dense(N_bottleneck, L1 reg.=10e-5) Dense(N_segment, linear activation)	MSE
VAE	Dense(N_segment) Dense(N_intermediate) Dense(N_bottleneck), Dense(N_bottleneck) Lambda(sampling) Dense(N_intermediate) Dense(N_segment, linear activation)	MSE + D_{KL}

SAE = Standard autoencoder
DAE = Denoising autoencoder
RAE = Robust autoencoder
CAE = Contractive autoencoder
SpAE = Sparse autoencoder
VAE = Variational autoencoder
N_segment = number of samples in one segment
N_bottleneck = number of nodes in latent space
N_intermediate = number of nodes in intermediate hidden layer
SD = Noise standard deviation
MSE = Mean squared error
D_{KL} = Kullback-Leibler divergence

[1] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73. Accessed in 05/10/2021.

4.4 Classification Algorithms

After training the AE models, the encoders contain the mapping that leads to the latent representations of the signals. This reduced representation can be fed to standard classification algorithms, responsible for finding metrics or patterns in the data that separate the classes in a supervised manner.

Based on the state-of-the-art [78], five different classifiers were tested, described in more detail in Table 4.3. The first consists of a single node, a perceptron; the perceptron model takes the ensemble of the features captured by the encoder and the classification labels (NSR or AF), and it adjusts its weights in the training step to fit the binary data. The output of the node is a real value between 0 and 1, being 0 defined as NSR and 1 as AF. For this reason, the chosen activation function for the output node was the sigmoid function, and the loss function to train the model was the binary cross-entropy, often used in binary classification problems.

The second model is an MLP, which has a basic ANN structure, with hidden stacked layers and an output node or nodes to perform classification. The training and the data encoding are similar to the previous model, but the MLP model is advantageous in that it can map the features into a non-linear space. The hidden layers used the Rectified Linear Unit (ReLU) activation function and the output node the sigmoid function.

Other widely used classification models include SVMs, k-NNs and DTs, that were also tested. The SVM model used the radial basis function as its kernel, the k-NN model was defined with the number of neighbors k equal to 5, and the DT classifier used the Classification and Regression Trees (CART) algorithmic implementation, as recommended in [78, 79].

Table 4.3: Structure and parameters of the proposed classifiers.

CLASSIFER	STRUCTURE/PARAMETERS
Perceptron	Dense(1, sigmoid activation)
MLP	Dense(N_bottleneck, ReLU activation) Dense(N_bottleneck, ReLU activation) Dense(N_bottleneck, ReLU activation) Dense(1, sigmoid activation)
SVM	RBF kernel
k-NN	k = 5 neighbors
DT	CART implementation

MLP = Multilayer Perceptron
SVM = Support Vector Machine
k-NN = k-Nearest Neighbor
DT = Decision Tree
N_bottleneck = number of nodes in latent space
RBF = Radial Basis Function
CART = Classification And Regression Trees

4.5 Training and Evaluating the Models

To check the best combination of AE and classifier to distinguish NSR from AF beats, a series of experiments were conducted.

First, to understand how the number of hidden layers and the compression level in the AE models affect the quality of the reconstructed signal, the SAE model was trained with different configurations. Maintaining the same training parameters and the same training and validation data, the compression levels 25%, 50%, 75%, 90%, and 95% were tested, and, for each, the number of hidden layers varied between 1, 3, and 5. Following the bottleneck approach, in AEs with 3 or 5 hidden layers the number of nodes was equally spaced between the input and the bottleneck. The metric used to evaluate their performance was the MSE of the validation data. Because these AE parameters change the models' complexity, an early stopping approach was used instead of fixing the number of epochs, which could lead to underfitting of some models and overfitting of others. The early stopping consisted of fixing the minimum loss difference $delta$ between consecutive epochs to consider an improvement, and, if no improvements were seen for a fixed number of epochs (patience constraint), the training was stopped.

After understanding the optimal number of hidden layers and fixing the compression factor, another parameter to be set is the activation function of the hidden layers, essential to capture nonlinear features in the data while training the AE. To achieve this, the most common activation functions were tested, that is, linear, sigmoid, hyperbolic tangent (TanH), ReLU and LeakyReLU functions. Each one of these was tested using the same training parameters and the same training and validation data. By comparing the speeds of convergence and the final MSE loss of the validation data while training the model, an activation function is proposed for the AEs.

Finally, the different models are tested by extracting the features from the AEs and feeding them into the classifiers. To avoid the models being biased towards already seen data, a data split based on subjects was performed. Using cross-validation, the data were first split into training and testing sets. The k-Fold Cross-Validation consists of splitting the data into k parts (called splits), where $k-1$ splits are used to train the model, and the remaining one to test it. Each split tries to maintain the same proportion of the various labels. The different combinations of the splits to train and to test are used, and k performance results are obtained.

Then, from the training set, 50% of subjects were picked to train the AEs and the other 50% to train the classifiers, as depicted in Figure 4.4; the testing data was used in both. To evaluate the models, a 2$\times$5 cross-validation was made, which means that a 5-Fold Cross-Validation is performed twice, with different fold compositions [83]. This process is called a Repeated k-fold Cross-Validation, and it is useful to reduce the error when estimating a model's performance, since different cross-validations can lead to different performance distributions.[1] To evaluate the models, several metrics were extracted, including accuracy, precision, recall, and F1-score, already defined in Section 2.7. Also, since the perceptron and MLP classifiers give an output between 0 and 1, the ROC curve and AUC can be obtained. These two indicators are useful because the choice of a threshold to separate the classes can significantly

[1] `https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/`. Accessed in 14/10/2021.

influence the model's performance.

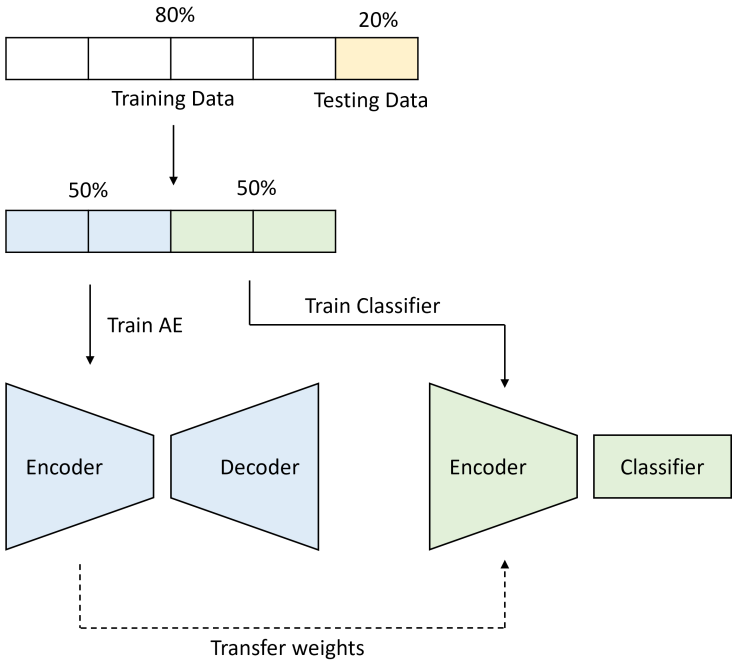


Figure 4.4: Diagram showing the data split approach to train the Autoencoders and the classifiers. The split avoids using the same data to train the Autoencoder and to train the classifier.

4.6 Alternative Approaches

This section describes two approaches to improve the algorithm and its viability in a real scenario. The first aims to evaluate the impact of using other rhythms and waveforms on the algorithm performance, and the second explores the use of information regarding RR-Intervals, as a way of incorporating short-term rhythm changes, which are characteristic of AF (see Section 2.4).

The approach presented so far is uniquely dedicated to distinguish ECG waveforms in NSR from ECG waveforms in AF. However, an algorithm suited for real-life applications must be able to deal with the variability and complexity of ECG records. For this reason, the CinC2017 challenge goal was for participants to develop an algorithm capable of distinguishing the four available dataset labels (NSR, AF, other rhythms and noisy acquisitions), as shown in Figure 3.3. Since the impact of other rhythms and waveforms in the proposed AF detection algorithm is unknown, these were also fed into an AE. Using the same preprocessing and training approaches previously described, tests were conducted to evaluate the ability of the AE to distinguish the three types of waves (excluding the noisy class). A diagram of this approach is depicted in Figure 4.5.

Because the problem is no longer of binary classification (two classes), the ANN-based classifiers (perceptron and MLP) had to be modified. Instead of having a single output node, the models were built with three output nodes, each corresponding to a label, and the cost function had to be changed from binary cross-entropy to categorical cross-entropy, which is suited for multiclass classification problems.

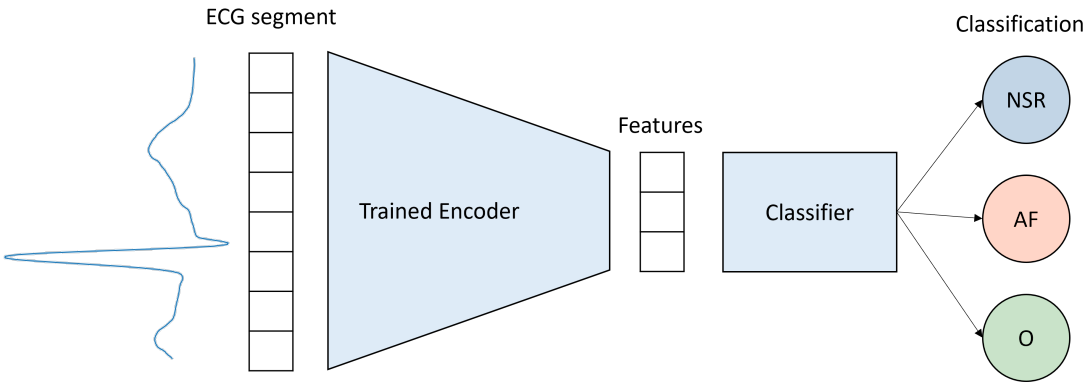


Figure 4.5: Proposed architecture for the three-class classification problem.

Also, to train these classifiers, the labels were changed from 0 (NSR), 1 (AF) and 2 (other), to the vector form using one-hot encoding, that is, [1,0,0], [0,1,0], and [0,0,1], respectively. Additionally, the problem was reduced again to a binary classification one, where the NSR and other beats were considered as one class, and AF beat as another, and tested for performance comparison.

Because the features extracted from the proposed AE models only rely on the waveform morphology, another alternative approach to enhance it is to provide some information regarding the RR-intervals of the ECG records, at least locally. As most state-of-the-art algorithms rely on metrics and features extracted from the RR-intervals, which are known to produce high performance rates (Section 3.1), the integration of RR-intervals into the classifiers is expected to increase their performances. To achieve this, a metric based on the differences between consecutive RR-intervals was created.

In ECG records, RR-intervals in AF are more irregular than in NSR. This means that, generally, in relation to a single R-peak R_i, the time difference between the next R-peak R_{i+1}, and the previous one R_{i-1}, is greater in AF than in NSR, that is:

$$|(R_{i+1} - R_i) - (R_i - R_{i-1})| \text{ in } AF > |(R_{i+1} - R_i) - (R_i - R_{i-1})| \text{ in } NSR \tag{4.2}$$

which is equivalent to:

$$|RR_i - RR_{i-1}| \text{ in } AF > |RR_i - RR_{i-1}| \text{ in } NSR \tag{4.3}$$

where RR_i is the RR-interval between R_i and R_{i-1}. Because higher HRs lead to smaller RR-intervals [84], the differences need to be normalized using the mean heart rate or the mean RR-intervals of the record. The proposed metric, hereinafter called Local Change of Successive Differences (LCSD), is defined for each R-peak R_i as:

$$\text{LCSD}\,(R_i) = \frac{|RR_i - RR_{i-1}|}{\frac{1}{N}\sum_{j=1}^{N} RR_j}, 1 < i < N \tag{4.4}$$

where N is the number of RR-intervals. Figure 4.6 describes the rationale behind the metric.

This measure is somehow similar to the HRV metric called the Root Mean Square of Successive Differences (RMSSD), which also aims to evaluate the variability of RR-intervals:

$$RMSSD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(RR_{i+1} - RR_i)^2} \tag{4.5}$$

However, since the RMSSD is normally computed from a set of ECG waveforms, it does not allow to pinpoint local changes of the RR-intervals as the LCSD. The proposed metric can be particularly relevant in the current context, where the individual waveforms are classified, allowing the classifiers to take into account not only the morphological features generated by the AE but also the local changes of the RR-intervals.

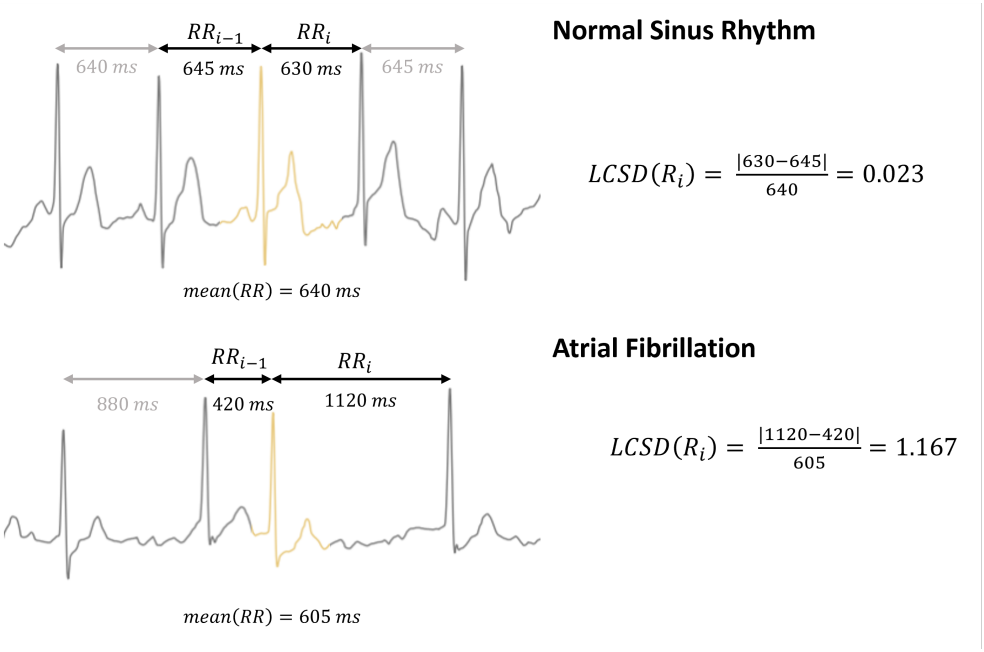


Figure 4.6: Illustrative example of the computation of the Local Change of Successive Differences (LCSD) and comparison between NSR and AF.

Chapter 5

Results

This chapter presents and discusses the results from all preliminary essays, including testing different number of hidden layers and activation functions, as well as the results from the core experiments, where the various combinations of AE and classifier were tested.

5.1 Compression Level and Hidden Layers

As stated in Section 4.5, the first preliminary experiment consisted in evaluating the impact of the compression level and the number of hidden layers in the ability of the SAE to reconstruct the ECG segments. All hidden layers used a linear activation function, the $delta$ parameter to perform early stopping was set to 1×10^{-6}, and the Adam optimizer [42] was used with a 0.001 learning rate. The train and validation data used 77% and 33% of the entire dataset, respectively. Figure 5.1 presents the results from the training process, namely the mean MSE loss of the validation data. Appendix B.1 contains the detailed results. Fixing the number of hidden layers, one can observe that, in general, there is no relevant information loss up to a 75% compression level, that corresponds to 45 nodes of the innermost AE layer. The validation MSE loss starts to steeply rise thereafter, reaching MSE errors close to 1×10^{-3} and higher. Interestingly, as already stated in [79, 78], the increase in the number of hidden layers appears to worsen the reconstruction capacity of the AEs, since, for each compression level, the MSE increases with the number of layers.

Using similar conditions as the previous experiment, the effectiveness of different activation functions to train the AEs was tested as well. The AEs used a 90% compression factor with a single hidden layer (18 nodes), and each training was performed with a batch size of 2000 samples and 144 epochs. Figure 5.2 shows the evolution of the validation loss with the number of epochs, and Table 5.1 presents the MSE validation losses by the end of the train. From all activation functions, the linear presents the best performance when reconstructing the input segments with the best convergence speed, followed by LeakyReLU and hyperbolic tangent. The mean MSE from the validation data were 4.77×10^{-4}, 5.50×10^{-4} and 5.78×10^{-4}, respectively. Sigmoid and ReLU presented the slowest convergence, with validation losses 8.87×10^{-4} and 1.57×10^{-3}.

As already mentioned in Sections 4.5 and 3.3, a Repeated K-Fold Cross-Validation[1] was chosen to train and test both the AEs and the classifiers, in a 2×5 approach, using the CinC2017 database. Since the data is organized by records, each one having a set of segmented ECG waveforms, the splits have 1185 records with nearly the same NSR:AF proportion.

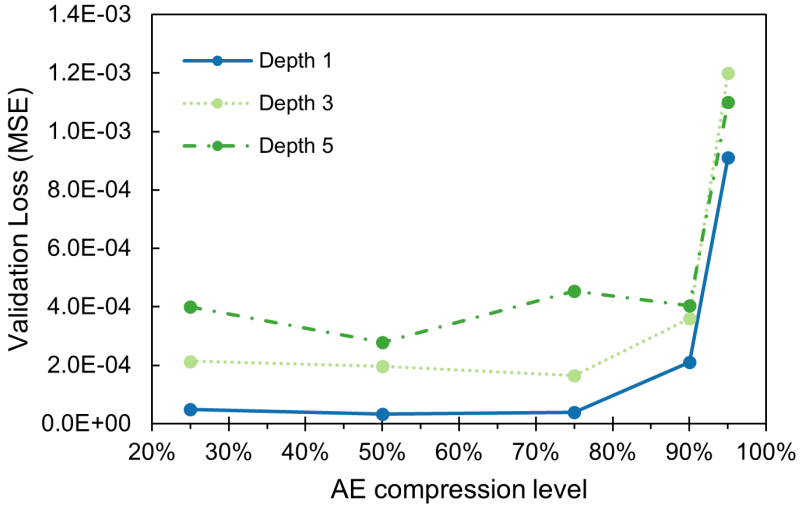


Figure 5.1: Validation losses varying the number of hidden layers (depth) and the compression level at the bottleneck layer. Appendix B.1 contains the detailed results.

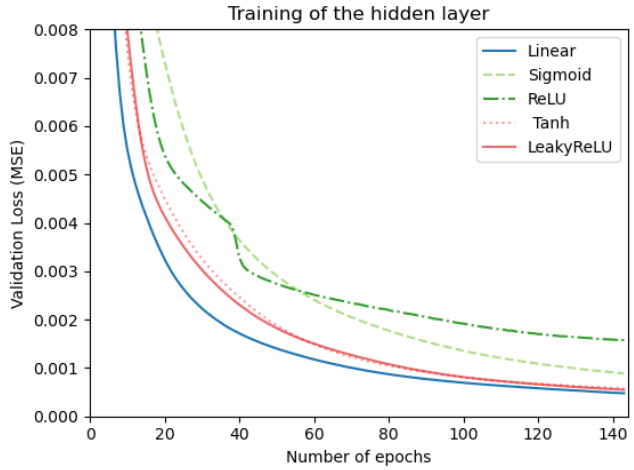


Figure 5.2: Training performance comparison of different activation functions applied to the AE's hidden layer, by monitoring the validation MSE loss with the number of epochs.

Table 5.1: Validation losses of different activation functions by the end of training.

ACTIVATION FUNCTION	VALIDATION LOSS (MSE)
Linear	4.77E-04
LeakyReLU	5.50E-04
TanH	5.78E-04
Sigmoid	8.87E-04
ReLU	1.57E-03

ReLU = Rectified Linear Unit
TanH = Hyperbolic tangent

[1] Using the function *RepeatedStratifiedKFold* from the Scikit-learn python library (`https://scikit-learn.org/`)

5.2 Autoencoder Configurations Evaluation

After establishing the basic structure for the SAE model, the different AE configurations were tested with the same training and test data, as described in Section 4.5. The following subsections present and discuss their reconstruction abilities, feature mapping and classification performances.

Reconstruction Ability

The MSE is the main metric to evaluate the waveform reconstruction ability of the different AEs. After 600 epochs using a 2000-sample batch size with the Adam optimizer (learning rate = 0.001), the resulting mean MSE values of all AEs are shown in Table 5.2.

Although all configurations resulted in similar MSE values, below 1×10^{-3}, some were able to capture the ECG waveforms with more accuracy. Namely, the DAE achieved the best reconstruction ability with mean MSE of 2.51×10^{-4} using the validation data, followed by the SpAE, RAE, SAE and VAE. The CAE led to the worst waveform reconstruction with a mean MSE of 8.08×10^{-4}. To exemplify the different performances, Figures 5.3 and 5.4 show the reconstruction of the two NSR and AF waveforms by the DAE and the CAE, respectively. Whereas the DAE is able to smoothly follow the ECG waveforms morphology, the CAE presents slight deviations, especially outside the QRS complex. All reconstruction examples are available in Appendix B.2.

In Figure 5.5, the same waveform examples are depicted but using the RAE, that achieved a mean MSE of 2.85×10^{-4}. Although very close to the DAE performance, this AE configuration presented reconstruction "noise", with sudden amplitude peaks, that suggest either lack of generalization capability or lack of training.

Table 5.2: Mean train and validation MSE values by AE type.

AE TYPE	TRAIN MSE	VALIDATION MSE
DAE	2.51E-4	2.51E-4
SpAE	2.54E-4	2.65E-4
RAE	2.73E-4	2.85E-4
SAE	2.85E-4	2.94E-4
VAE	4.70E-4	4.94E-4
CAE	7.49E-4	8.08E-4

Denoising Autoencoder (DAE)

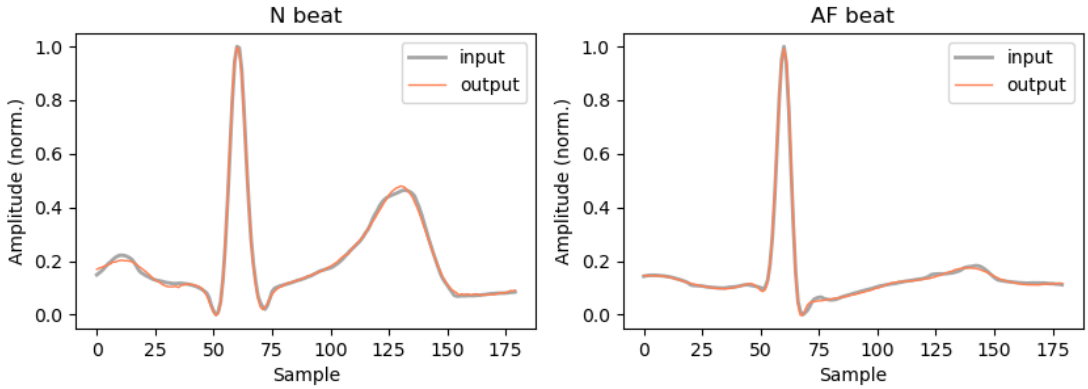


Figure 5.3: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Denoising Autoencoder.

Contractive Autoencoder (CAE)

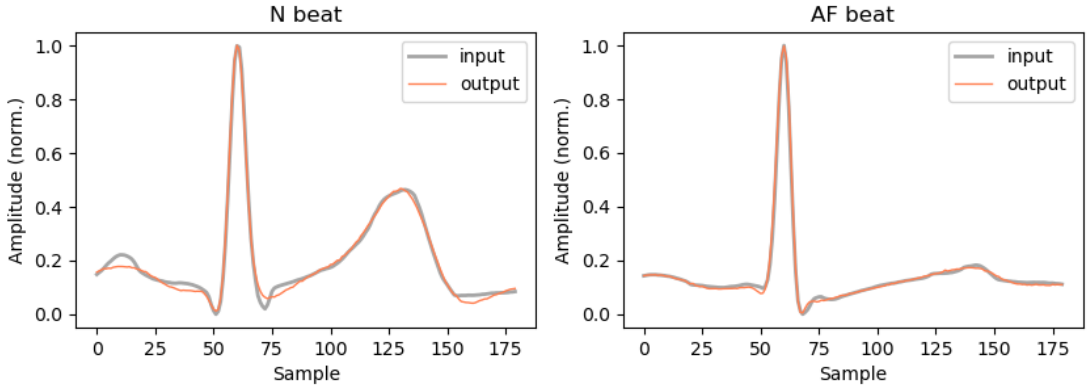


Figure 5.4: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Contractive Autoencoder.

Robust Autoencoder (RAE)

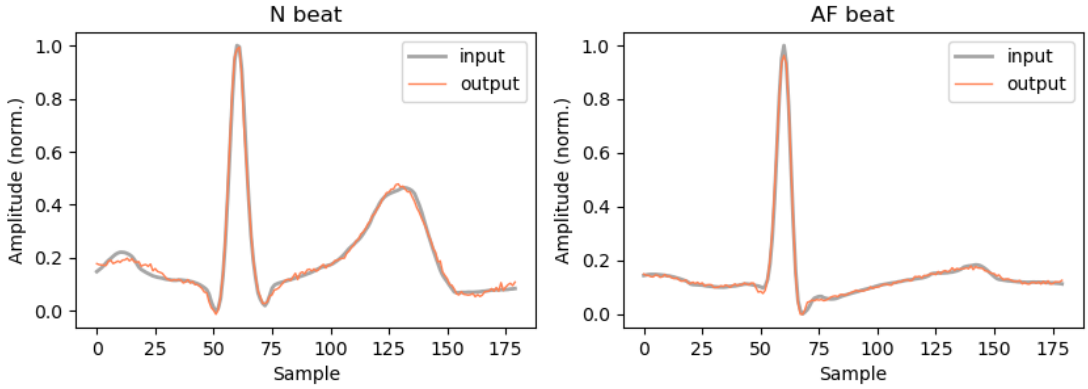


Figure 5.5: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Robust Autoencoder.

Feature Mapping

To provide insight on the features generated by the different AE types, a boxplot containing the distributions of the features from both NSR and AF waveforms can be used to compare the latent spaces. Since the AEs have 180 input nodes and a 90% compression level, the resulting latent space dimension is 18, meaning there are 18 different features.

Using the testing data, the distributions of the latent space mapped by the SAE are shown in Figure 5.6. The feature distributions take different values and ranges, and considerable overlap between the NSR and AF classes can be seen among all features. Because there is no obvious separation between them, this suggests that class separation can only be achieved by non-linear methods. The feature mapping of the VAE model, that regularizes the latent space by the Kullback-Leibler divergence, is shown in Figure 5.7. Although with significant overlap between classes, the features' values and ranges appear to be more regular than the SAE. The feature distributions generated by the remaining models are available in Appendix B.3.

In an attempt to check a non-linear relationship between the classes, a visualization technique called t-Distributed Stochastic Neighbor Embedding (t-SNE) was used. t-SNE is a nonlinear dimensionality reduction tool that tries to minimize the Kullback-Leibler divergence between the joint probabilities of the generated (low-dimensional) and the original (high-dimensional) data [85]. This way, the 18-feature-space is reduced to two components and a two-dimensional plot can be made.

Using a 4000-sample of both NSR and AF waveform rhythms, the t-SNE plots of the SAE and VAE configurations are depicted in Figure 5.8. In both plots, the encoded data shows a generalized overlap of both classes with two distinct clusters. Since these clusters do not reflect class separation, this may be due to signal quality differences that lead to different mapped features between standard ECG waves and anomalous ones. The t-SNE plots generated by the remaining models are available in Appendix B.4.

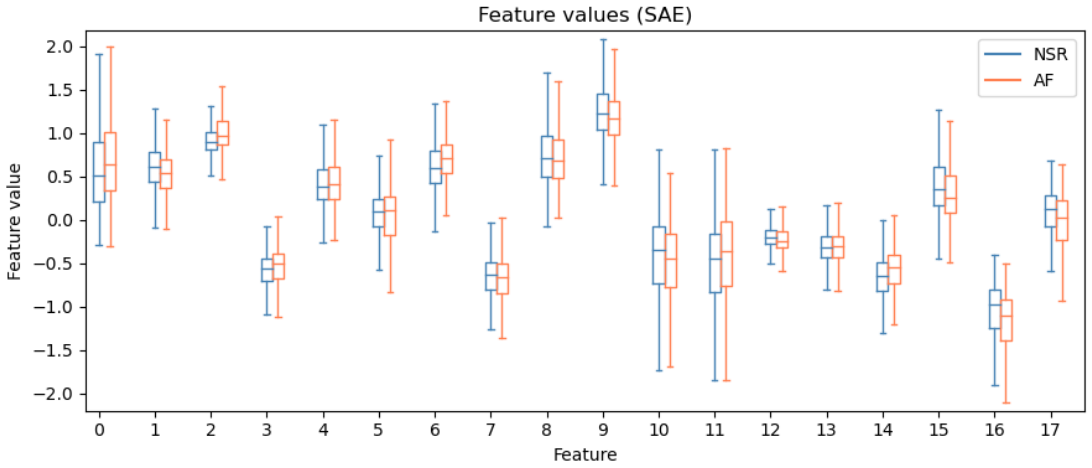


Figure 5.6: Feature value distributions of the hidden features generated by the SAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

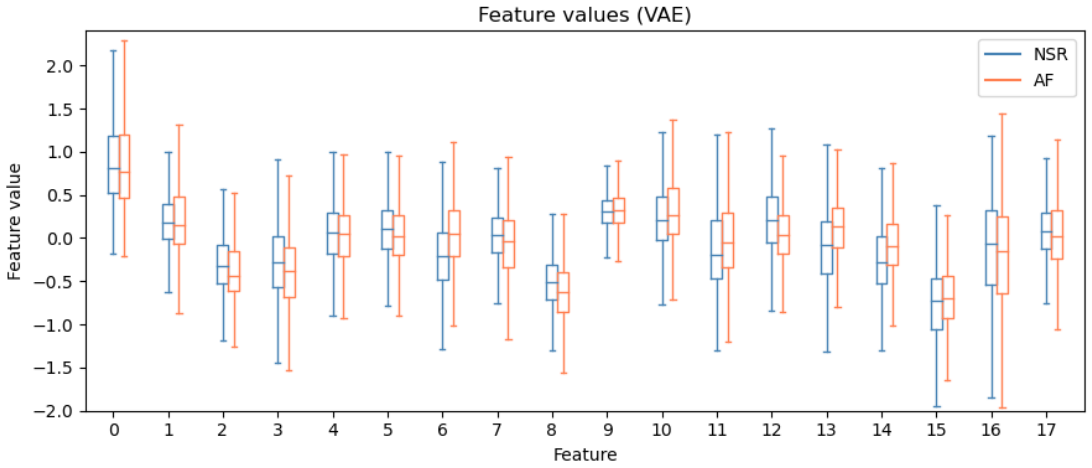
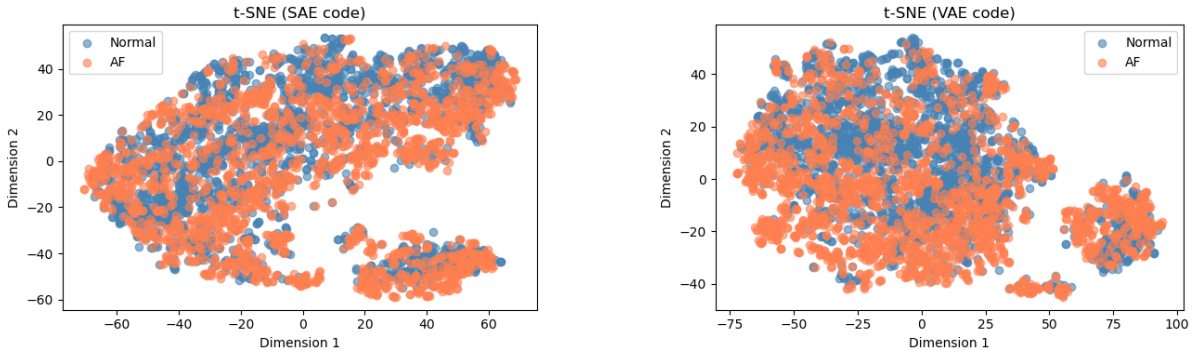


Figure 5.7: Feature value distributions of the hidden features generated by the VAE model for NSR and AF test instances.Considerable overlap between the classes' feature distributions is seen.



(a) (b)

Figure 5.8: 2-Dimensional t-SNE plot from a 4000-sample data generated by (a) the SAE and (b) the VAE. Although there are two distinct clusters in each plot, they do not correspond to class separation.

Classification Performance

After AE training, the resulting encoder was coupled with several classifiers, already described in Section 4.4. For each combination, the accuracy, precision, recall, F1-score and AUC are described in Table 5.3 using the test data. Because of the cross-validation process, the performance values correspond to the mean and standard deviation of the 10 iterations. The row in bold refers to the classifiers with the best F1-score.

The SpAE generated features in combination with the MLP classifier achieved the best performance among all model combinations, with a 82.2% F1-score and a AUC equal to 0.902. The ROC curve and the confusion matrix with the best score using this model are shown in Figures 5.9 and 5.10. Following the SpAE, the DAE-MLP (82.0%), the RAE-MLP (81.5%), the SAE-MLP (81.3%), the VAE-SVM (80.0%), and the CAE-SVM (77.2%) obtained the best F1-scores. The lack of performance from the CAE models reveals that the features extracted from it are not as meaningful as the other types. This can be correlated with the worse reconstruction results previously obtained (Table 5.2), suggesting that this model would probably benefit with more training epochs or more training data.

In general, the MLP and SVM classifiers present the best performances, and the k-NN, perceptron and CART the worst. This observation is supported by the high non-linear relationship between the features of NSR and AF classes (Figures 5.6 and 5.7). For instance, the perceptron model, based on the linear combination of its inputs, struggles to achieve the accuracy levels obtained by MLP and SVM, since all accuracy and F1 scores are below 70%. Although the k-NN and CART classifiers are non-linear, the needed complexity from them to match the MLP and SVM is elevated. Parameter optimization could enhance their ability to classify the instances. These differences in performance were also observed by [78].

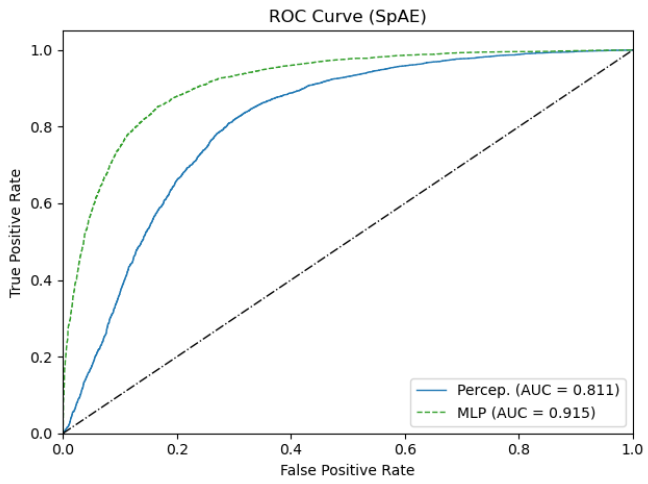


Figure 5.9: ROC curves for the perceptron and MLP classifiers with the best F1-score obtained by using the SpAE. Respective AUC in the legend.

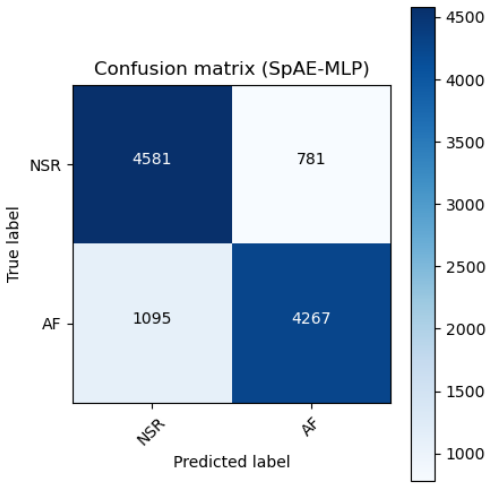


Figure 5.10: Confusion matrix of the SpAE-MLP classifier with the best F1-score.

Table 5.3: Performance metrics of different classifiers trained with the features generated by various AEs, using a 2×5 cross-validation procedure. The classifiers with the best F1-score are highlighted in bold.

AE Type	Classifier	Accuracy	Precision	Recall	F1-score	AUC
SpAE	Perceptron	0.727±0.015	0.731±0.013	0.717±0.024	0.724±0.018	0.785±0.018
	MLP	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**
	SVM	0.810±0.011	0.822±0.009	0.791±0.018	0.806±0.012	**-**
	k-NN	0.758±0.013	0.808±0.011	0.676±0.022	0.736±0.016	-
	CART	0.699±0.007	0.709±0.007	0.673±0.010	0.691±0.008	-
DAE	Perceptron	0.725±0.022	0.726±0.019	0.722±0.030	0.724±0.024	0.779±0.023
	MLP	**0.822±0.015**	**0.827±0.017**	**0.814±0.029**	**0.820±0.017**	**0.900±0.014**
	SVM	0.811±0.010	0.823±0.008	0.792±0.017	0.807±0.011	-
	k-NN	0.759±0.012	0.809±0.010	0.679±0.019	0.738±0.014	-
	CART	0.695±0.006	0.705±0.007	0.671±0.008	0.688±0.006	-
VAE	Perceptron	0.728±0.014	0.733±0.013	0.716±0.024	0.724±0.017	0.787±0.015
	MLP	0.799±0.010	0.812±0.010	0.778±0.017	0.794±0.012	0.876±0.011
	SVM	**0.805±0.008**	**0.822±0.008**	**0.772±0.017**	**0.800±0.010**	-
	k-NN	0.758±0.013	0.803±0.012	0.684±0.024	0.739±0.017	-
	CART	0.693±0.010	0.706±0.010	0.661±0.017	0.683±0.012	-
RAE	Perceptron	0.734±0.014	0.738±0.013	0.725±0.022	0.731±0.016	0.792±0.015
	MLP	**0.817±0.015**	**0.824±0.015**	**0.806±0.02**	**0.815±0.815**	**0.896±0.012**
	SVM	0.789±0.016	0.789±0.015	0.789±0.019	0.789±0.016	-
	k-NN	0.742±0.012	0.788±0.011	0.664±0.019	0.720±0.015	-
	CART	0.685±0.005	0.697±0.006	0.655±0.009	0.676±0.006	-
SAE	Perceptron	0.737±0.014	0.741±0.011	0.728±0.022	0.734±0.016	0.795±0.015
	MLP	**0.816±0.015**	**0.824±0.018**	**0.803±0.018**	**0.813±0.015**	**0.893±0.014**
	SVM	0.790±0.013	0.790±0.012	0.792±0.018	0.791±0.014	-
	k-NN	0.747±0.010	0.792±0.011	0.671±0.015	0.726±0.012	-
	CART	0.696±0.009	0.707±0.007	0.669±0.017	0.688±0.011	-
CAE	Perceptron	0.643±0.048	0.663±0.045	0.573±0.087	0.613±0.068	0.693±0.052
	MLP	0.777±0.013	0.793±0.013	0.752±0.028	0.771±0.016	0.853±0.012
	SVM	**0.771±0.017**	**0.769±0.018**	**0.776±0.022**	**0.772±0.018**	**-**
	k-NN	0.739±0.013	0.778±0.011	0.669±0.022	0.719±0.016	-
	CART	0.697±0.012	0.709±0.010	0.669±0.021	0.688±0.015	-

5.3 Discussion

From all previous experiments, several remarks can be made about the best AE model to distinguish AF from NSR beats:

- AEs with a single hidden layer are able to better represent an ECG waveform than AEs with multiple hidden layers (Figure 5.1).
- ECG waveform reconstruction is stable until a 75% compression level in the AE's bottleneck layer (Figure 5.1).
- AE training is optimized using either linear, LeakyReLU or TanH activation functions in the hidden layer (Figure 5.2).
- Different AE models lead to different latent representations of the ECG waveforms, and there seems to be no linear relationship to differentiate NSR and AF waveforms from them (Figures 5.6 and 5.7); this is supported by the low performance of the perceptron classifier that relies on the linear combination of its inputs.
- The DAE and SpAE models, that act directly on the AE's nodes, achieved the best waveform reconstructions and the best classification performances (Table 5.4); this suggests that better reconstructions also promote feature meaningfulness.
- The MLP and SVM classifiers led to the best classification performances across all AE models, and the CART and perceptron models led to the worst performances (Table 5.4).
- The best model from all AE-classifier combinations was the SpAE-MLP model, that achieved an F1-score of 82.2% and a AUC equal to 0.902 (Table 5.4).

Table 5.4: Summary of the best AE-classifier combinations, and the respective performance metrics. The model with the best F1-score is in bold.

AE TYPE	CLASSIFIER	ACCURACY	PRECISION	RECALL	F1-SCORE	AUC
SAE	MLP	0.816±0.015	0.824±0.018	0.803±0.018	0.813±0.015	0.893±0.014
DAE	MLP	0.822±0.015	0.827±0.017	0.814±0.029	0.82±0.017	0.9±0.014
RAE	MLP	0.817±0.015	0.824±0.015	0.806±0.02	0.815±0.815	0.896±0.012
CAE	SVM	0.771±0.017	0.769±0.018	0.776±0.022	0.772±0.018	-
SpAE	**MLP**	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**
VAE	SVM	0.805±0.008	0.822±0.008	0.772±0.017	0.8±0.01	-

5.4 Non-Linear Activation Function

As previously observed, the reconstruction of ECG waveform segments was favoured by using linear, LeakyReLU and TanH activation functions (Figure 5.2 and Table 5.1); the linear function, with best performance, was used to conduct the tests on the various types of AEs. This means that the models mapped linear relationships between the ECG segment samples and the AE's 18-feature-code. To test whether the choice of the linear activation function had an impact on classification performance, the same SpAE model was tested using the LeakyReLU function in the hidden layers.

Using the same training parameters, this SpAE model resulted in a mean MSE of 2.73×10^{-4} with train data and 2.96×10^{-4} with validation data. The ensemble of features extracted using the LeakyReLU function is compared across 4000 samples with the linear activation function in Figure 5.11. By visual inspection, one can observe that, when using the LeakyReLU, the code has a narrower range of feature values than using the linear function.

In terms of classification ability, as described in Table 5.5, negligible improvements are seen using the LeakyReLU function. The best classifier, similarly to previous experiments, was the MLP, with a mean F1-score of 82.3%, a 0.1% improvement, and a AUC of 0.902, the same achieved by using the linear function. These observations lead us to conclude that no classification improvements are seen when using a non-linear activation function.

The comparison between both approaches can be further explored with the corresponding ROC curves (Figure 5.12) and confusion matrices (Figure 5.13) of the cross validation iteration with the best F1-score.

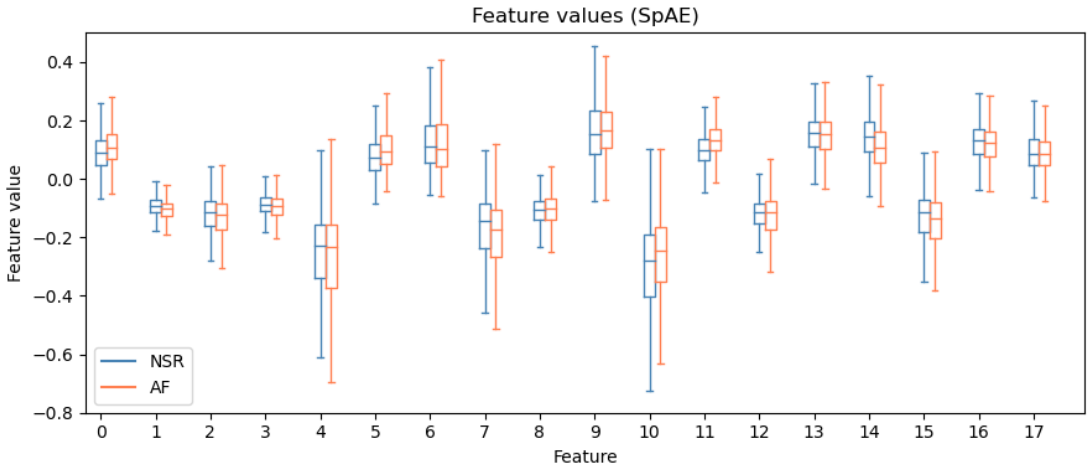
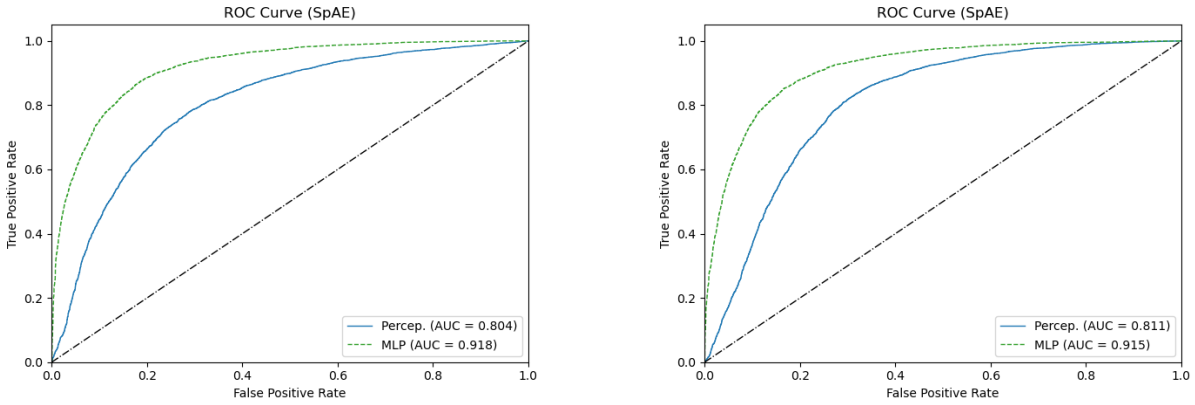


Figure 5.11: Feature value distributions of the hidden features generated by the SpAE model for NSR AF using the LeakyReLU hidden activation function.

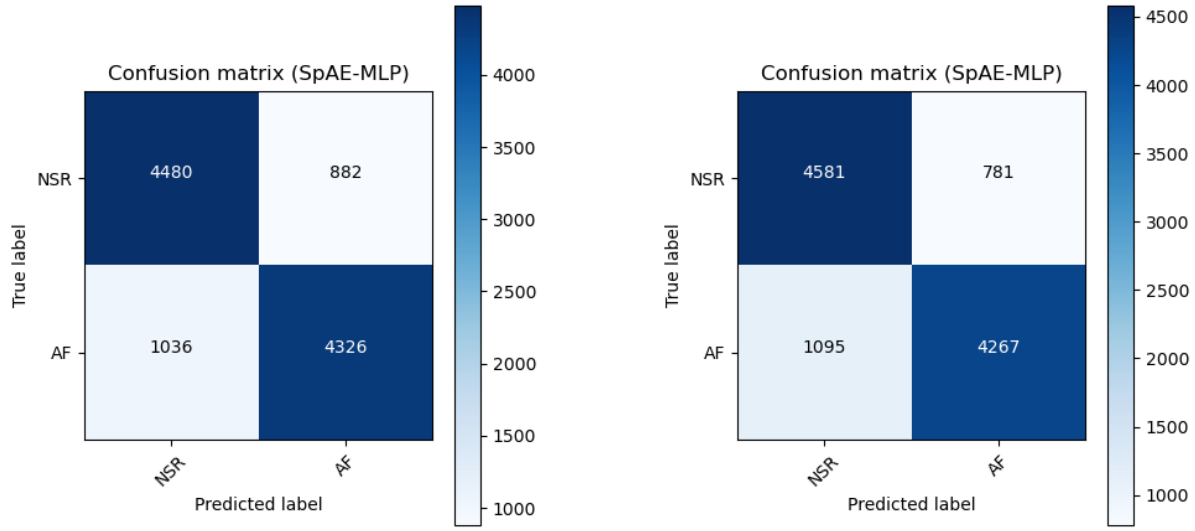
Table 5.5: Summary of the best SpAE-classifier combinations using the LeakyReLU activation function in the hidden layer and the respective performance metrics. The performance scores of the SpAE with the linear activation function previously obtained is also present. The best F1-score model is highlighted in bold.

AE Type	Classifier	Accuracy	Precision	Recall	F1-score	AUC
SpAE Leaky-ReLU	Perceptron	0.724±0.015	0.727±0.010	0.719±0.029	0.723±0.019	0.779±0.014
	SVM	0.796±0.010	0.797±0.010	0.795±0.017	0.796±0.010	-
	k-NN	0.747±0.011	0.795±0.012	0.667±0.019	0.725±0.014	-
	CART	0.685±0.007	0.694±0.007	0.660±0.013	0.677±0.008	-
	MLP	**0.825±0.012**	**0.830±0.007**	**0.817±0.029**	**0.823±0.015**	**0.902±0.010**
SpAE Linear	**MLP**	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**



(a) LeakyReLU (b) Linear

Figure 5.12: ROC curves of the SpAE using (a) the LeakyReLU and (b) the Linear activation functions in the hidden layer using both perceptron and MLP classifiers. The respective AUCs are in the legends.



(a) LeakyReLU (b) Linear

Figure 5.13: Confusion matrices of the SpAE using (a) the LeakyReLU and (b) the Linear activation functions in the hidden layer using the MLP classifier with best F1-score.

5.5 Testing with Other Rhythms

To test the impact of adding other ECG rhythms to the proposed classifier model, the respective class records were preprocessed with the same procedure as the NSR and AF classes. Using the SpAE as the main model, the training resulted on an average MSE loss of 2.72×10^{-4} and a loss of 2.73×10^{-4} with testing data. The testing split consisted of 6010 instances for each class. Because of their higher accuracy rates, only the SVM and MLP were tested. The MLP model, with three output nodes, used the softmax activation function. A classifier where the NSR and the other rhythm classes are joined was also tested.

The feature value distributions are shown in Figure 5.14, and the respective t-SNE plot is shown in Figure 5.15. In both plots, the features from other rhythms get mapped into the same overall range of values, suggesting that only a non-linear relationship between the features and the classes is possible.

Table 5.6 describes the performance metrics of the three classifier approaches. Since it is a multiclass problem, precision, accuracy, recall and F1-score were computed using the weighted method, where the macro criterion, meaning that the values equal to the mean of each label's metrics, is used. It is noticeable that the classification performance suffers when adding a third class. An F1-score of 58.1% was achieved, in contrast with the 82.2% where only AF and NSR were the classifier targets. From the analysis of the respective confusion matrix in Figure 5.17, 67.7% of NSR, 72.5% of AF and 43.5% of other rhythm labels were correctly classified. Also, 34.8% of the other rhythm labels were misclassified as NSR. This means that the classifier is having more trouble distinguishing these two classes, possibly meaning that their waveforms have great resemblance.

For that reason, the other test consisted of joining the NSR and other rhythm classes, assigning them the same label. When training in this condition, an improvement of the F1-score was seen, increasing from 58.1% to 64.3%. Still, 36.4% of AF waveforms were misclassified, whereas the classifier with no 'other' class misclassified 18.2% of AF instances (Figures 5.10 and 5.17), suggesting that some waveforms from other rhythms may be morphologically similar to AF. This also reflects that, because no RR-interval information is given to the classifiers, this approach is not sufficiently robust to perform accurate detection of AF.

Table 5.6: Summary of the SpAE-classifier combinations that also include other rhythms, and the respective performance metrics. The model which classifies NSR, AF and other rhythms separately, and the model which joins NSR and other rhythms classes are compared with the model that only classifies AF and NSR. The best F1-scores are highlighted in bold.

AE Type	Classifier	Accuracy	Precision	Recall	F1-score	AUC
SpAE NSR/AF/O	**MLP**	**0.586±0.014**	**0.581±0.014**	**0.586±0.014**	**0.581±0.013**	-
	SVM	0.586±0.009	0.582±0.009	0.586±0.009	0.579±0.008	-
SpAE (NSR+O)/AF	**MLP**	**0.776±0.010**	**0.685±0.017**	**0.605±0.026**	**0.643±0.019**	**0.829±0.015**
	SVM	0.771±0.010	0.709±0.017	0.528±0.023	0.606±0.021	-
SpAE NSR/AF	**MLP**	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**

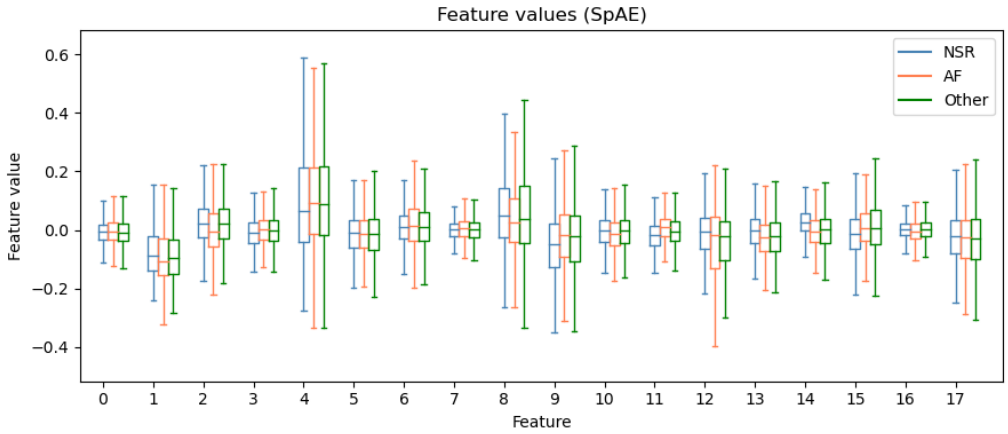


Figure 5.14: Feature value distributions of the hidden features generated by the SpAE-MLP model for NSR, AF, and other rhythms.

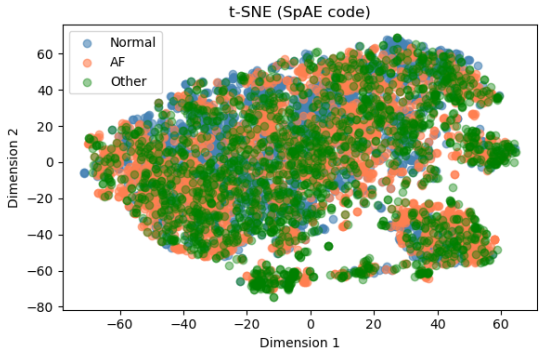


Figure 5.15: 2-Dimensional t-SNE plot from a 6000-sample data generated by the SpAE adding instances from the other rhythm class.

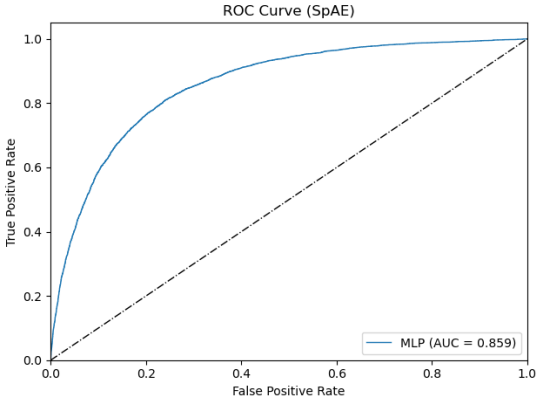
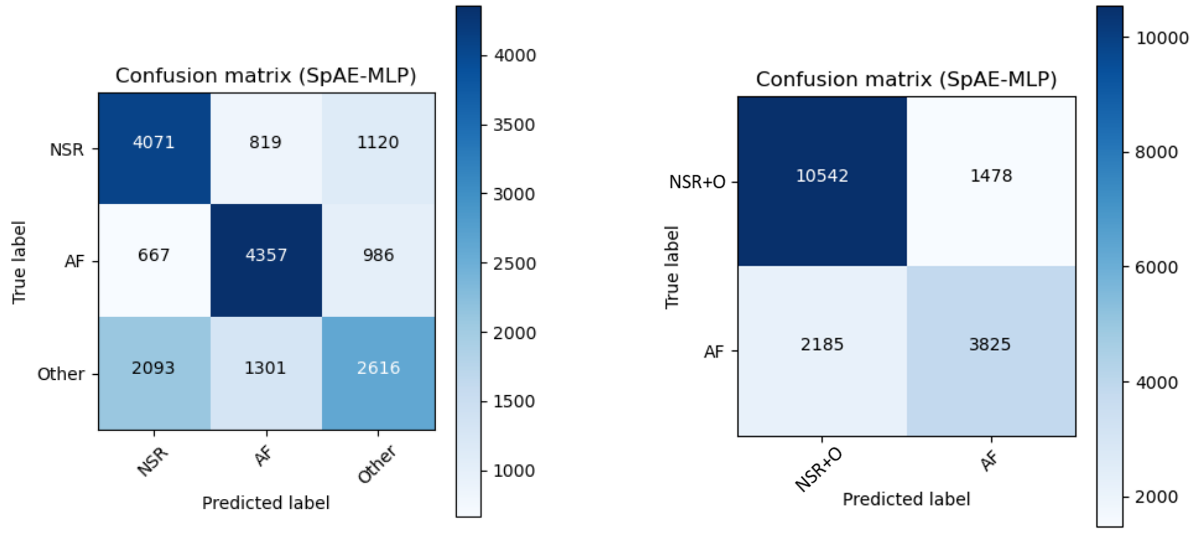


Figure 5.16: ROC curve and respective AUC of the SpAE binary classification model using NSR and other rhythms as one class.



(a) NSR / AF / O

(b) (NSR+O) / AF

Figure 5.17: Confusion matrices of the SpAE-MLP models which aim to classify between (a) NSR, other rhythms and AF and (b) NSR/other rhythms and AF.

5.6 Integration of RR-Interval Differences

As already described in Section 4.6, a strategy to feed the classifiers with information regarding the RR-intervals was defined using the LCSD proposed metric. All classifiers were tested using both the features generated by the SpAE model, and the LCSD of the R-peak associated with the waveform. It must be noted that not all AF beats present higher LCSD values than NSR ones, as noticeable in Figure 5.18 (feature number 18), however there is a significant difference in their distributions. The role of this feature is to reinforce AF detection when the extracted AE features wrongly lead to NSR classification. This observation is supported by the increase in Recall from 81.3% to 88.8% using the MLP classifier (Table 5.7), which indicates a significant reduction of false negatives. The equal increase in precision led to an F1-score of 88.2%, an improvement of 6%. Besides the SVM, the LCSD feature also increased the performance of the remaining classifiers (Table 5.7). The best AUC using the MLP was of 0.954.

Significant improvements can also be seen when comparing the ROC curves and the corresponding AUCs, as in Figure 5.19. By observing the confusion matrices in Figure 5.20, the percentage of correct AF labels raised from 79.6% to 84.8%.

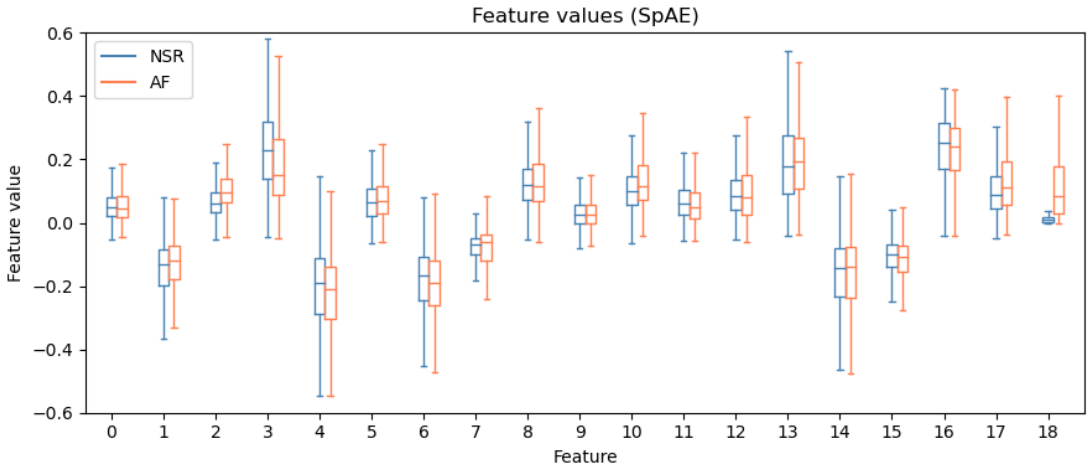
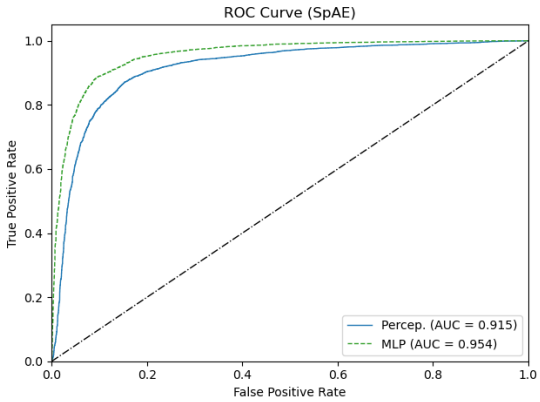


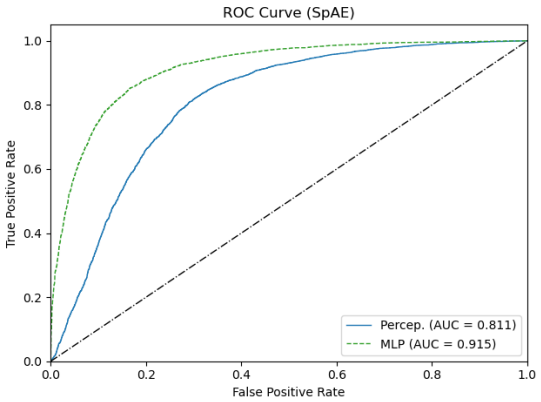
Figure 5.18: Feature value distributions of the hidden features generated by the SpAE/LCSD model for NSR and AF test instances. The feature number 18 corresponds to the LCSD distributions, which present significant mean and value range differences.

Table 5.7: Summary of the SpAE-classifier combinations that also include the LCSD, and the respective performance metrics. The previously tested model which doesn't use the LCSD is also present for comparison. The best F1-scores are highlighted in bold.

AE Type	Classifier	Accuracy	Precision	Recall	F1-score	AUC
SpAE Code + LCSD	Perceptron	0.825±0.020	0.841±0.028	0.806±0.048	0.821±0.023	0.898±0.016
	SVM	0.871±0.006	0.884±0.022	0.855±0.030	0.868±0.008	-
	k-NN	0.841±0.013	0.875±0.014	0.797±0.041	0.833±0.018	-
	CART	0.783±0.011	0.780±0.028	0.792±0.024	0.785±0.006	-
	MLP	**0.880±0.009**	**0.877±0.029**	**0.888±0.027**	**0.882±0.006**	**0.950±0.007**
SpAE Code	**MLP**	**0.824±0.012**	**0.832±0.014**	**0.813±0.019**	**0.822±0.013**	**0.902±0.011**

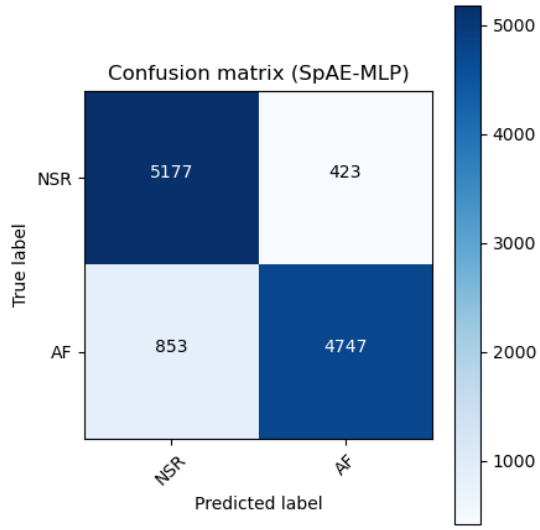


(a) Code+LCSD

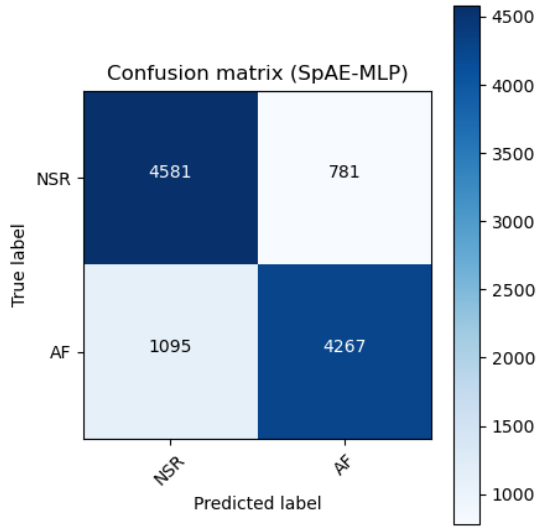


(b) Code only

Figure 5.19: ROC curves and respective AUCs of the SpAE models using the perceptron and MLP as classifiers. The model (a) which uses the SpAE generated features (code) in combination with LCSD is compared with (b) the model that does not use it.



(a) Code+LCSD



(b) Code only

Figure 5.20: Confusion matrices of (a) the SpAE-MLP model that uses the LCSD as a feature and (b) the SpAE-MLP model that does not use it.

5.7 Discussion of the Alternative Approaches

The alternative approaches in Sections 5.4 to 5.6, have given insight on the viability of using AEs to detect ECG waveforms in AF. The main findings were:

- Morphological AEs can extract meaningful information by only using linear activation functions, since the classifiers using features from an AE with a non-linear activation function had similar performance results as the features extracted from an AE with a linear activation function (Table 5.5).
- Adding ECG waveforms from other rhythms than NSR or AF leads to worse classification results (Table 5.6), possibly because the extracted features are not sufficiently robust to capture the inherent morphological differences between the waveforms. In this case, further testings should be done by either increasing the number of features (i.e., the number of hidden nodes in the AE), or by testing some form of supervised learning that can shape the latent space (e.g., a supervised VAE).
- Adding temporal information regarding the RR-intervals, using the proposed metric LCSD, greatly increases the ability of classifiers to distinguish AF from NSR waveforms (Table 5.7).
- The best tested approach to detect AF beats was by using an MLP classifier fed with unsupervised features generated by a SpAE and the local differences of the RR-intervals (LCSD). This model achieved an 88.2% F1-score and an AUC of 0.950 (Table 5.7), comparable to the current state-of-the-art algorithms.
- The aforementioned F1-score is above the median of current AF detection algorithms based on atrial and ventricular features (85.6%) but stays well below the overall F1-score median of the considered algorithms of Section 3.1 (94.0%). However because the methodology in this work was not developed to classify ECG records, but instead to classify ECG waveform segments, the method can be underrepresented.
- Comparing with the algorithms developed for the CinC2017 challenge [64], using invisible single-lead ECG recordings, the F1-score was superior to the best algorithms (83.0%). However, besides being record classification based, the CinC2017 challenge consisted of additionally classifying other rhythm and noisy records.

Chapter 6

Conclusions

The work developed in this thesis allowed the development of a supervised approach for AF detection based on heartbeat waveform extraction from ECG records. The conclusions from this work are summarized in the outline and the possible future directions are in the following section.

6.1 Outline

Since the prevalence of CVDs is high and expected to increase around the globe, their detection is of high importance. For that purpose, pervasive monitoring systems based on physiological measurements such as PPG and ECG have been developed in the last decades. Moreover, besides professional diagnosis of CVDs, a recent and emergent paradigm of monitoring devices is based on invisible technologies, which aim to continuously track physiological signals by integrating them with the patient's daily activities. Multiple studies have presented convincing evidence that this type of approach can help deal with CVD management, either for prevention or for disease monitoring.

Since hardware requirements are especially relevant on wearable and invisible devices, a first study regarding reflective PPG signal was made. Its acquisition only needs a single interface region with the body, therefore the easiest to implement. The study revealed that a sampling rate as low as 50 Hz combined with either quadratic or cubic spline interpolation could achieve time and amplitude resolutions close to a 1 kHz sampled signal. Using this approach, high-quality PPG waveforms can be obtained, and, consequently, accurate HRV measurements can be extracted to monitor key physiological parameters.

However, in the pursuit of developing a method to monitor cardiac function and detect heart disease, the PPG has been found to contain limited information for diagnostics, reason for which the ECG signal was chosen to be the target of analysis. Being AF one of the most relevant CVDs, an algorithmic approach for its detection was developed. Taking into account the context of wearable and invisible ECG monitoring systems, a single-lead ECG database for AF detection was selected.

In agreement with the current state-of-the-art methodologies, the proposed method consisted of using a ANN-based approach for classification of ECG waveforms. To do so, a type of ANN called AE was

first used to achieve a compressed version of the ECG segments, and then the correspondent generated features were used to feed a classifier. Since there are many AE types and possible structures, some of them were tested, including the SAE, DAE, RAE, CAE, SpAE and VAE models. The perceptron, MLP, SVM, k-NN and DT were selected as the classifiers. After performing some preliminary tests to define the best set of parameters, each combination of AE-classifier was tested and evaluated with several standard performance metrics.

After data preprocessing, a 2×5 cross-validation approach was used to split the data into training and testing sets and to allow the models evaluation. From all experiments, the SpAE model in combination with a MLP achieved the best performance in distinguishing NSR and AF segments, with an F1-score of 82.2% and an AUC of 0.902. This AE model acts directly on the AE's hidden layer, by promoting sparsity, to achieve meaningful compact representations of its inputs.

Adding other ECG rhythm segments, available in the database, to the SpAE-MLP model had a negative impact on its performance, since it achieved an F1-score of 58.1%. By joining the NSR and the other rhythm classes, the metric improved to 64.3%. The difficulty of the algorithm to deal with other rhythm waveforms suggests that the features extracted from it are not sufficiently robust. An approach to cope with such challenge could involve the development of a supervised method capable of mapping the segments into a more partitioned subspace.

Because of the high-performance rates of algorithms that use HRV or related metrics, and not the segments themselves, an approach to integrate them into the model was explored. By defining a new metric focused on the local RR-interval differences around an R-peak (LCSD), significant improvements were observed in AF detection performance. The new SpAE-MLP model using this metric as one of its features, resulted in a 88.2% F1-score (+8.0%) and an AUC of 0.950.

In comparison with state-of-the-art algorithms, the aforementioned F1-score is above the median of current AF detection algorithms based on atrial and ventricular features (85.6%) but stays well below the overall F1-score median of the considered algorithms of (94.0%). However, comparing with the algorithms developed for the CinC2017 challenge, using invisible single-lead ECG recordings, the F1-score was superior to the best algorithms (83.0%). These comparisons, however, are not the most appropriate because the methodology in this work was not developed to classify ECG records, but instead it was developed to classify ECG waveform segments. Also, the CinC2017 challenge consisted of additionally classifying other rhythm and noisy records. Thus, the potential to diagnose AF using this approach could be further explored. Also, the algorithm was not tested with other databases such as the MITDB.

6.2 Future Work

Despite satisfactory results in distinguishing AF from NSR waveforms, future work opportunities have been identified throughout the execution of this work, namely:

- The AF records available from the CinC2017 challenge are not guaranteed to only have AF episodes, that is, it may be possible that some records contain brief transitions between NSR

and AF rhythms. As so, the performance of the proposed models may be underestimated regarding the ECG beats. Further studies should be conducted with a more detailed database, such as the AFDB or the MITDB, which have precise annotations regarding rhythm and waveform changes.

- Since it is mainly waveform-based, the proposed classification model is not yet suited to classify records. A model which uses beat-by-beat classification to evaluate ECG records would still be needed to compare with other approaches, namely the ones made by the CinC2017 participants. To be suited for a real-life implementation, such model would involve the study of a classification threshold based on the ROC curves to avoid false positives rates (Figure 5.19).
- Another approach to deal with the previous issue to classify records would be to test the AEs with time window containing several beats, or to test other ANN structures appropriate for time-series classification, such as LSTMs.
- Because of the great diversity of AE models and general ANN structures and training parameters, improvements to the proposed AEs can still be possible. Supervised or semi-supervised approaches could possibly enhance the performance rates (e.g., a supervised VAE model).
- A type of supervised learning called Multiple Instance Learning (MIL) could also be used to train the AF detection models. This method assigns a single class label to a bag of instances and, during training, if a bag contains a single positive instance (i.e., AF), the whole bag is considered positive, thus classifying a record as an AF rhythm.
- The proposed AEs are significantly attached to the ECG waveform morphologies. Since some of the state-of-the-art algorithms which use HRV and associated ventricular metrics (Tables 3.1 and 3.2) still have higher performance rates than the proposed approach, a revision of the method could be made.
- Because it is a ANN-based approach, where the meaning of the learned features is unknown (black-box), the proposed model would greatly benefit from a study to try unveil the meaning behind the classification decisions. Such process is part of a field called Explainable Machine Learning.

References

[1] World Health Organization. (Nov. 6, 2021). “Cardiovascular diseases (CVDs) fact sheet”, World Health Organization, [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

[2] W. T. Qureshi, Z.-M. Zhang, P. P. Chang, W. D. Rosamond, D. W. Kitzman, L. E. Wagenknecht, and E. Z. Soliman, “Silent myocardial infarction and long-term risk of heart failure: The ARIC study”, *Journal of the American College of Cardiology*, vol. 71, no. 1, pp. 1–8, Jan. 2, 2018, ISSN: 0735-1097. DOI: 10.1016/j.jacc.2017.10.071.

[3] L. F. Lyckhage, M. L. Hansen, K. Procida, and T. Wienecke, “Prehospital continuous ECG is valuable for very early detection of atrial fibrillation in patients with acute stroke”, *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 9, p. 105 014, Sep. 1, 2020, ISSN: 1052-3057. DOI: 10.1016/j.jstrokecerebrovasdis.2020.105014.

[4] M. A. Serhani, H. T. El Kassabi, H. Ismail, and A. Nujum Navaz, “ECG monitoring systems: Review, architecture, processes, and key challenges”, *Sensors*, vol. 20, no. 6, p. 1796, Jan. 2020, Number: 6 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/s20061796.

[5] S. Hong, Y. Zhou, J. Shang, C. Xiao, and J. Sun, “Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review”, *Computers in Biology and Medicine*, vol. 122, p. 103 801, Jul. 1, 2020, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2020.103801.

[6] H. P. da Silva, “The biosignal c.a.o.s.: Reflections on the usability of physiological sensing for human-computer interaction practitioners and researchers”, in *Converging Clinical and Engineering Research on Neurorehabilitation II*, J. Ibáñez, J. González-Vargas, J. M. Azorín, M. Akay, and J. L. Pons, Eds., ser. Biosystems & Biorobotics, Cham: Springer International Publishing, 2017, pp. 807–811, ISBN: 978-3-319-46669-9. DOI: 10.1007/978-3-319-46669-9_132.

[7] A. Fadhil, “Beyond technical motives: Perceived user behavior in abandoning wearable health & wellness trackers”, *arXiv:1904.07986 [cs]*, Apr. 16, 2019. arXiv: 1904.07986.

[8] H. P. da Silva, C. Carreiras, A. Lourenço, A. Fred, R. C. das Neves, and R. Ferreira, “Off-the-person electrocardiography: Performance assessment and clinical correlation”, *Health and Technology*, vol. 4, no. 4, pp. 309–318, Apr. 1, 2015, ISSN: 2190-7196. DOI: 10.1007/s12553-015-0098-y.

[9] H. Plácido da Silva, “Biomedical sensors as invisible doctors”, in *Regenerative Design in Digital Practice: A Handbook for the Built Environment*, Eurac Research, Jan. 1, 2019, pp. 322–329, ISBN: 978-3-9504607-3-5.

[10] A. Dos Santos Silva, H. Almeida, H. P. da Silva, and A. Oliveira, “Design and evaluation of a novel approach to invisible electrocardiography (ECG) in sanitary facilities using polymeric electrodes”, *Scientific Reports*, vol. 11, no. 1, p. 6222, Mar. 18, 2021, ISSN: 2045-2322. DOI: 10.1038/s41598-021-85697-2.

[11] G. Lippi, F. Sanchis-Gomar, and G. Cervellin, “Global epidemiology of atrial fibrillation: An increasing epidemic and public health challenge”, *International Journal of Stroke*, vol. 16, no. 2, pp. 217–221, Feb. 1, 2021, Publisher: SAGE Publications, ISSN: 1747-4930. DOI: 10.1177/1747493019897870.

[12] J. Zhang, S. P. Johnsen, Y. Guo, and G. Y. H. Lip, “Epidemiology of atrial fibrillation: Geographic and ecological risk factors, age, sex, genetics”, *Cardiac Electrophysiology Clinics*, Risk Factors in Atrial Fibrillation: Appraisal of AF Risk Stratification, vol. 13, no. 1, pp. 1–23, Mar. 1, 2021, ISSN: 1877-9182. DOI: 10.1016/j.ccep.2020.10.010.

[13] F. J. Wesselius, M. S. van Schie, N. M. S. De Groot, and R. C. Hendriks, “Digital biomarkers and algorithms for detection of atrial fibrillation using surface electrocardiograms: A systematic review”, *Computers in Biology and Medicine*, vol. 133, p. 104 404, Jun. 1, 2021, ISSN: 0010-4825. DOI: 10.1016/j.compbiomed.2021.104404.

[14] A. B. Hertzman, “Photoelectric plethysmography of the fingers and toes in man”, *Proceedings of the Society for Experimental Biology and Medicine*, vol. 37, no. 3, pp. 529–534, Dec. 1, 1937, Publisher: SAGE Publications, ISSN: 0037-9727. DOI: 10.3181/00379727-37-9630.

[15] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran, “A review on wearable photoplethysmography sensors and their potential future applications in health care”, *International journal of biosensors & bioelectronics*, vol. 4, no. 4, pp. 195–202, 2018, ISSN: 2573-2838. DOI: 10.15406/ijbsbe.2018.04.00125.

[16] J. Allen, “Photoplethysmography and its application in clinical physiological measurement”, *Physiological Measurement*, vol. 28, R1–39, Apr. 1, 2007. DOI: 10.1088/0967-3334/28/3/R01.

[17] J. Spigulis, L. Gailite, A. Lihachev, and R. Erts, “Simultaneous recording of skin blood pulsations at different vascular depths by multiwavelength photoplethysmography”, *Applied Optics*, vol. 46, no. 10, pp. 1754–1759, Apr. 1, 2007, Publisher: Optical Society of America, ISSN: 2155-3165. DOI: 10.1364/AO.46.001754.

[18] J. G. Webster, *Design of pulse oximeters*, ser. Medical science series. Bristol ; Philadelphia: Institute of Physics Pub, 1997, 244 pp., ISBN: 978-0-7503-0467-2.

[19] H.-Y. Jan, M.-F. Chen, T.-C. Fu, W.-C. Lin, C.-L. Tsai, and K.-P. Lin, “Evaluation of coherence between ECG and PPG derived parameters on heart rate variability and respiration in healthy volunteers with/without controlled breathing”, *Journal of Medical and Biological Engineering*, vol. 39, no. 5, pp. 783–795, Oct. 1, 2019, ISSN: 2199-4757. DOI: 10.1007/s40846-019-00468-9.

[20] A. H. Khandoker, C. K. Karmakar, and M. Palaniswami, “Comparison of pulse rate variability with heart rate variability during obstructive sleep apnea”, *Medical Engineering & Physics*, vol. 33, no. 2, pp. 204–209, Mar. 2011, ISSN: 1873-4030. DOI: 10.1016/j.medengphy.2010.09.020.

[21] A. Schäfer and J. Vagedes, “How accurate is pulse rate variability as an estimate of heart rate variability? a review on studies comparing photoplethysmographic technology with an electrocardiogram”, *International Journal of Cardiology*, vol. 166, no. 1, pp. 15–29, Jun. 5, 2013, ISSN: 1874-1754. DOI: 10.1016/j.ijcard.2012.03.119.

[22] J. Fine, K. L. Branan, A. J. Rodriguez, T. Boonya-ananta, Ajmal, J. C. Ramella-Roman, M. J. McShane, and G. L. Coté, “Sources of inaccuracy in photoplethysmography for continuous cardiovascular monitoring”, *Biosensors*, vol. 11, no. 4, p. 126, Apr. 16, 2021, ISSN: 2079-6374. DOI: 10.3390/bios11040126.

[23] J. de Moraes, M. Id, Rocha, G. Vasconcelos, J. E. Vasconcelos Filho, V. Hugo, V. Albuquerque, and A. Alexandria, “Advances in photopletysmography signal analysis for biomedical applications”, *Sensors*, vol. 18, Jun. 1, 2018. DOI: 10.3390/s18061894.

[24] P. Podrid, R. Malhotra, R. Kakkar, and P. Noseworthy, *Podrid’s Real-World ECGs: Volume 1, The Basics: A Master’s Approach to the Art and Practice of Clinical ECG Interpretation*, ser. Podrid’s Real-World ECGs:A Master’s Approach to the Art and Practice of Clinical ECG Interpretation. Cardiotext Publishing, 2012, ISBN: 978-1-935395-76-8.

[25] I. Pinto, A. Fred, C. Rodrigues, and H. Plácido da Silva, *Electrophysiology of the Heart and the Electrocardiogram: Visual Depictions*. Dec. 10, 2020.

[26] S. R. Steinhubl, A. M. Edwards, J. Waalen, R. Zambon, R. Mehta, L. Ariniello, G. Ebner, K. Baca-Motes, C. Carter, E. Felicione, T. Sarich, and E. Topol, “Healthcare resource utilization associated with electrocardiograph (ECG) sensor patch screening for atrial fibrillation (AF): Results from the mHealth screening to prevent strokes (mSToPS) trial”, *Journal of the American College of Cardiology*, ACC.19: The American College of Cardiology 68th Annual Scientific Sessions, vol. 73, no. 9, p. 296, Mar. 12, 2019, ISSN: 0735-1097. DOI: 10.1016/S0735-1097(19)30904-0.

[27] J. H. E. Yong, K. Thavorn, J. S. Hoch, M. Mamdani, K. E. Thorpe, P. Dorian, M. Sharma, A. Laupacis, and D. J. Gladstone, “Potential cost-effectiveness of ambulatory cardiac rhythm monitoring after cryptogenic stroke”, *Stroke*, vol. 47, no. 9, pp. 2380–2385, Sep. 1, 2016, Publisher: American Heart Association. DOI: 10.1161/STROKEAHA.115.011979.

[28] P. Morton and D. Fontaine, *Critical care nursing: A holistic approach*. Lippincott Williams & Wilkins, Jul. 19, 2013, 1 p., Journal Abbreviation: Critical Care Nursing: A Holistic Approach: 10th Ed.

Pages: 1321 Publication Title: Critical Care Nursing: A Holistic Approach: 10th Ed., ISBN: 978-1-60913-749-6.

[29] S. Sareen, M. Nayyar, B. Wheeler, M. Skelton, and R. Khouzam, "Electrocardiographic artifact potentially misleading to the wrong management", *Annals of Translational Medicine*, vol. 6, pp. 17–17, Jan. 1, 2018. DOI: 10.21037/atm.2017.11.33.

[30] H. Silva, A. Lourenço, R. Lourenço, P. Leite, D. Coutinho, and A. Fred, "Study and evaluation of a single differential sensor design based on electro-textile electrodes for ECG biometrics applications", in *2011 IEEE SENSORS*, ISSN: 1930-0395, Oct. 2011, pp. 1764–1767. DOI: 10.1109/ICSENS.2011.6127028.

[31] M. P. Witvliet, E. P. M. Karregat, J. C. L. Himmelreich, J. S. S. G. d. Jong, W. A. M. Lucassen, and R. E. Harskamp, "Usefulness, pitfalls and interpretation of handheld single-lead electrocardiograms", *Journal of Electrocardiology*, vol. 66, pp. 33–37, 2021, ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2021.02.011.

[32] J. Ribeiro Pinto, J. S. Cardoso, and A. Lourenço, "Evolution, current challenges, and future possibilities in ECG biometrics", *IEEE Access*, vol. 6, pp. 34 746–34 776, 2018, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2849870.

[33] E. J. Benjamin, P. Muntner, A. Alonso, M. S. Bittencourt, C. W. Callaway, A. P. Carson, A. M. Chamberlain, A. R. Chang, S. Cheng, S. R. Das, F. N. Delling, L. Djousse, M. S. Elkind, J. F. Ferguson, M. Fornage, L. C. Jordan, S. S. Khan, B. M. Kissela, K. L. Knutson, T. W. Kwan, D. T. Lackland, T. T. Lewis, J. H. Lichtman, C. T. Longenecker, M. S. Loop, P. L. Lutsey, S. S. Martin, K. Matsushita, A. E. Moran, M. E. Mussolino, M. O'Flaherty, A. Pandey, A. M. Perak, W. D. Rosamond, G. A. Roth, U. K. Sampson, G. M. Satou, E. B. Schroeder, S. H. Shah, N. L. Spartano, A. Stokes, D. L. Tirschwell, C. W. Tsao, M. P. Turakhia, L. B. VanWagner, J. T. Wilkins, S. S. Wong, S. S. Virani, and n. null null, "Heart disease and stroke statistics—2019 update: A report from the american heart association", *Circulation*, vol. 139, no. 10, e56–e528, Mar. 5, 2019, Publisher: American Heart Association. DOI: 10.1161/CIR.0000000000000659.

[34] J. Kornej, C. S. Börschel, E. J. Benjamin, and R. B. Schnabel, "Epidemiology of atrial fibrillation in the 21st century", *Circulation Research*, vol. 127, no. 1, pp. 4–20, Jun. 19, 2020, Publisher: American Heart Association. DOI: 10.1161/CIRCRESAHA.120.316340.

[35] J. S. Healey, S. J. Connolly, M. R. Gold, C. W. Israel, I. C. Van Gelder, A. Capucci, C. P. Lau, E. Fain, S. Yang, C. Bailleul, C. A. Morillo, M. Carlson, E. Themeles, E. S. Kaufman, S. H. Hohnloser, and ASSERT Investigators, "Subclinical atrial fibrillation and the risk of stroke", *The New England Journal of Medicine*, vol. 366, no. 2, pp. 120–129, Jan. 12, 2012, ISSN: 1533-4406. DOI: 10.1056/NEJMoa1105575.

[36] J. W. Greenberg, T. S. Lancaster, R. B. Schuessler, and S. J. Melby, "Postoperative atrial fibrillation following cardiac surgery: A persistent complication", *European Journal of Cardio-Thoracic Surgery*, vol. 52, no. 4, pp. 665–672, Oct. 1, 2017, ISSN: 1010-7940. DOI: 10.1093/ejcts/ezx039.

[37] P. Kirchhof, S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, M. Castella, H.-C. Diener, H. Heidbuchel, J. Hendriks, G. Hindricks, A. S. Manolis, J. Oldgren, B. A. Popescu, U. Schotten, B. Van Putte, P. Vardas, S. Agewall, J. Camm, G. Baron Esquivias, W. Budts, S. Carerj, F. Casselman, A. Coca, R. De Caterina, S. Deftereos, D. Dobrev, J. M. Ferro, G. Filippatos, D. Fitzsimons, B. Gorenek, M. Guenoun, S. H. Hohnloser, P. Kolh, G. Y. H. Lip, A. Manolis, J. McMurray, P. Ponikowski, R. Rosenhek, F. Ruschitzka, I. Savelieva, S. Sharma, P. Suwalski, J. L. Tamargo, C. J. Taylor, I. C. Van Gelder, A. A. Voors, S. Windecker, J. L. Zamorano, and K. Zeppenfeld, "2016 ESC guidelines for the management of atrial fibrillation developed in collaboration with EACTS", *European Journal of Cardio-Thoracic Surgery: Official Journal of the European Association for Cardio-Thoracic Surgery*, vol. 50, no. 5, e1–e88, Nov. 2016, ISSN: 1873-734X. DOI: 10.1093/ejcts/ezw313.

[38] P. Podrid, R. Malhotra, R. Kakkar, and P. Noseworthy, *Podrid's Real-World ECGs: Volume 4, Arrhythmias: A Master's Approach to the Art and Practice of Clinical ECG Interpretation*, ser. Podrid's Real-World ECGs:A Master's Approach to the Art and Practice of Clinical ECG Interpretation. Cardiotext Publishing, 2015, ISBN: 1-935395-02-5.

[39] N. Sperelakis, Y. Kurachi, A. Terzic, and M. Cohen, *Heart Physiology and Pathophysiology*, th$. Elsevier Science, 2000, ISBN: 978-0-08-053388-9.

[40] K. T. Hickey, T. C. Riga, S. A. Mitha, and M. J. Reading, "Detection and management of atrial fibrillation using remote monitoring", *The Nurse practitioner*, vol. 43, no. 3, pp. 24–30, Mar. 12, 2018, ISSN: 0361-1817. DOI: 10.1097/01.NPR.0000530214.17031.45.

[41] D. Graupe, *Principles of artificial neural networks*, 3rd edition, ser. Advanced series on circuits and systems volume 7. New Jersey: World Scientific, 2013, 363 pp., ISBN: 978-981-4522-73-1.

[42] G. Bonaccorso, *Mastering Machine Learning Algorithms: Expert techniques for implementing popular machine learning algorithms, fine-tuning your models, and understanding how they work, 2nd Edition*. Packt Publishing Ltd, Jan. 31, 2020, 799 pp., Google-Books-ID: M0vODwAAQBAJ, ISBN: 978-1-83882-191-3.

[43] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, *Extracting and composing robust features with denoising autoencoders*. Jan. 1, 2008, 1096 pp., Journal Abbreviation: Proceedings of the 25th International Conference on Machine Learning Pages: 1103 Publication Title: Proceedings of the 25th International Conference on Machine Learning. DOI: 10.1145/1390156.1390294.

[44] H. Saleh, *Machine Learning Fundamentals*. Packt Publishing, Nov. 29, 2018, 240 pp., ISBN: 978-1-78980-355-6.

[45] G. Dong, G. Liao, H. Liu, and G. Kuang, "A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images", *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 3, pp. 44–68, Sep. 2018, ISSN: 2168-6831, 2473-2397, 2373-7468. DOI: 10.1109/MGRS.2018.2853555.

[46] M. Hossin and S. M.N, “A review on evaluation metrics for data classification evaluations”, *International Journal of Data Mining & Knowledge Management Process*, vol. 5, pp. 01–11, Mar. 31, 2015. DOI: 10.5121/ijdkp.2015.5201.

[47] K. Hajian-Tilaki, “Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation”, *Caspian Journal of Internal Medicine*, vol. 4, no. 2, pp. 627–635, 2013, ISSN: 2008-6164.

[48] J. Slocum, A. Sahakian, and S. Swiryn, “Diagnosis of atrial fibrillation from surface electrocardiograms based on computer-detected atrial activity”, *Journal of Electrocardiology*, vol. 25, no. 1, pp. 1–8, Jan. 1, 1992, ISSN: 0022-0736. DOI: 10.1016/0022-0736(92)90123-H.

[49] T. F. Yang, B. Devine, and P. W. Macfarlane, “Artificial neural networks for the diagnosis of atrial fibrillation”, *Medical & Biological Engineering & Computing*, vol. 32, no. 6, pp. 615–619, Nov. 1994, ISSN: 0140-0118. DOI: 10.1007/BF02524235.

[50] I. Christov, G. Bortolan, and I. Daskalov, “Automatic detection of atrial fibrillation and flutter by wave rectification method”, *Journal of Medical Engineering & Technology*, vol. 25, no. 5, pp. 217–221, Oct. 2001, ISSN: 0309-1902. DOI: 10.1080/03091900110065942.

[51] J. A. Queiroz, A. Junior, F. Lucena, and A. K. Barros, “Diagnostic decision support systems for atrial fibrillation based on a novel electrocardiogram approach”, *Journal of Electrocardiology*, vol. 51, no. 2, pp. 252–259, Apr. 2018, ISSN: 1532-8430. DOI: 10.1016/j.jelectrocard.2017.10.014.

[52] J. Sacha, “Interaction between heart rate and heart rate variability”, *Annals of Noninvasive Electrocardiology*, vol. 19, no. 3, pp. 207–216, 2014, ISSN: 1542-474X. DOI: 10.1111/anec.12148.

[53] F. Shaffer and J. P. Ginsberg, “An overview of heart rate variability metrics and norms”, *Frontiers in Public Health*, vol. 5, p. 258, Sep. 28, 2017, ISSN: 2296-2565. DOI: 10.3389/fpubh.2017.00258.

[54] S. Dash, K. H. Chon, S. Lu, and E. A. Raeder, “Automatic real time detection of atrial fibrillation”, *Annals of Biomedical Engineering*, vol. 37, no. 9, pp. 1701–1709, Sep. 2009, ISSN: 0090-6964, 1573-9686. DOI: 10.1007/s10439-009-9740-z.

[55] L. li, C. Liu, C. Liu, Q. Zhang, and B. Li, *Physiological Signal Variability Analysis Based on the Largest Lyapunov Exponent*. Jan. 1, 2009, 1 p., Pages: 5. DOI: 10.1109/BMEI.2009.5305169.

[56] R. He, K. Wang, N. Zhao, Y. Liu, Y. Yuan, Q. Li, and H. Zhang, “Automatic detection of atrial fibrillation based on continuous wavelet transform and 2d convolutional neural networks”, *Frontiers in Physiology*, vol. 9, p. 1206, 2018, ISSN: 1664-042X. DOI: 10.3389/fphys.2018.01206.

[57] M.-L. Huang and Y.-S. Wu, “Classification of atrial fibrillation and normal sinus rhythm based on convolutional neural network”, *Biomedical Engineering Letters*, vol. 10, no. 2, pp. 183–193, May 1, 2020, ISSN: 2093-985X. DOI: 10.1007/s13534-020-00146-9.

[58] Y. Xia, N. Wulan, K. Wang, and H. Zhang, “Detecting atrial fibrillation by deep convolutional neural networks”, *Computers in Biology and Medicine*, vol. 93, pp. 84–92, Feb. 1, 2018, ISSN: 1879-0534. DOI: 10.1016/j.compbiomed.2017.12.007.

[59] O. Faust, A. Shenfield, M. Kareem, T. R. San, H. Fujita, and U. R. Acharya, "Automated detection of atrial fibrillation using long short-term memory network with RR interval signals", *Computers in Biology and Medicine*, vol. 102, pp. 327–335, Nov. 1, 2018, ISSN: 1879-0534. DOI: 10.1016/j.compbiomed.2018.07.001.

[60] Z. Xiong, M. P. Nash, E. Cheng, V. V. Fedorov, M. K. Stiles, and J. Zhao, "ECG signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network", *Physiological Measurement*, vol. 39, no. 9, p. 094 006, Sep. 24, 2018, ISSN: 1361-6579. DOI: 10.1088/1361-6579/aad9ed.

[61] C. Yuan, Y. Yan, L. Zhou, J. Bai, and L. Wang, "Automated atrial fibrillation detection based on deep learning network", in *2016 IEEE International Conference on Information and Automation (ICIA)*, Aug. 2016, pp. 1159–1164. DOI: 10.1109/ICInfA.2016.7831994.

[62] L. Chen and Y. He, "Identification of atrial fibrillation from electrocardiogram signals based on deep neural network", *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 4, pp. 838–846, May 1, 2019. DOI: 10.1166/jmihi.2019.2626.

[63] N. Cortez, L. Sanchez, and I. Couso, "Semi-supervised recurrent variational autoencoder approach for visual diagnosis of atrial fibrillation", *IEEE Access*, vol. PP, pp. 1–1, Mar. 9, 2021. DOI: 10.1109/ACCESS.2021.3064854.

[64] G. D. Clifford, C. Liu, B. Moody, L.-w. H. Lehman, I. Silva, Q. Li, A. E. Johnson, and R. G. Mark, "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017", *Computing in cardiology*, vol. 44, 10.22489/CinC.2017.065–469, Sep. 2017, ISSN: 2325-8861.

[65] D. Azariadi, V. Tsoutsouras, S. Xydis, and D. Soudris, "ECG signal analysis and arrhythmia detection on IoT wearable medical devices", in *2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, May 2016, pp. 1–4. DOI: 10.1109/MOCAST.2016.7495143.

[66] M. Bsoul, H. Minn, and L. Tamil, "Apnea MedAssist: Real-time sleep apnea monitor using single-lead ECG", *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 416–427, May 2011, Conference Name: IEEE Transactions on Information Technology in Biomedicine, ISSN: 1558-0032. DOI: 10.1109/TITB.2010.2087386.

[67] H. R. Marston, R. Hadley, D. Banks, and M. D. C. M. Duro, "Mobile self-monitoring ECG devices to diagnose arrhythmia that coincide with palpitations: A scoping review", *Healthcare*, vol. 7, no. 3, p. 96, Sep. 2019, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute. DOI: 10.3390/healthcare7030096.

[68] L. A. Sposato, L. E. Cipriano, G. Saposnik, E. Ruíz Vargas, P. M. Riccio, and V. Hachinski, "Diagnosis of atrial fibrillation after stroke and transient ischaemic attack: A systematic review and meta-analysis", *The Lancet. Neurology*, vol. 14, no. 4, pp. 377–387, Apr. 2015, ISSN: 1474-4465. DOI: 10.1016/S1474-4422(15)70027-X.

[69] G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, M. Kosel, and E. P. Scilingo, "Predicting mood changes in bipolar disorder through heartbeat nonlinear dynamics", *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 4, pp. 1034–1043, Jul. 2016, Conference Name: IEEE Journal of Biomedical and Health Informatics, ISSN: 2168-2208. DOI: 10.1109/JBHI.2016.2554546.

[70] G. Bosco, E. De Marzi, P. Michieli, H. R. Omar, E. M. Camporesi, J. Padulo, A. Paoli, D. Mangar, and M. Schiavon, "12-lead holter monitoring in diving and water sports: A preliminary investigation", *Diving and Hyperbaric Medicine*, vol. 44, no. 4, pp. 202–207, Dec. 2014, ISSN: 1833-3516.

[71] S.-J. Jung, H.-S. Shin, and W.-Y. Chung, "Driver fatigue and drowsiness monitoring system with embedded electrocardiogram sensor on steering wheel", *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 43–50, 2014, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2012.0032, ISSN: 1751-9578. DOI: 10.1049/iet-its.2012.0032.

[72] A. Lourenco, A. P. Alves, C. Carreiras, R. Duarte, and A. Fred, *CardioWheel: ECG Biometrics on the Steering Wheel*. Sep. 7, 2015, 267 pp., Pages: 270, ISBN: 978-3-319-23460-1. DOI: 10.1007/978-3-319-23461-8_27.

[73] G. Tuboly, G. Kozmann, O. Kiss, and B. Merkely, "Atrial fibrillation detection with and without atrial activity analysis using lead-i mobile ECG technology", *Biomedical Signal Processing and Control*, vol. 66, p. 102 462, Apr. 1, 2021, ISSN: 1746-8094. DOI: 10.1016/j.bspc.2021.102462.

[74] P. Hamilton, "Open source ECG analysis", in *Computers in Cardiology*, ISSN: 0276-6547, Sep. 2002, pp. 101–104. DOI: 10.1109/CIC.2002.1166717.

[75] A. Lourenço, H. Silva, P. Leite, R. Lourenço, and A. Fred, "Real time electrocardiogram segmentation for finger based ECG biometrics", in *Proceedings of the International Conference on Bio-inspired Systems and Signal Processing - BIOSIGNALS, (BIOSTEC 2012)*, Backup Publisher: INSTICC ISSN: 2184-4305, SciTePress, 2012, pp. 49–54, ISBN: 978-989-8425-89-8. DOI: 10.5220/0003777300490054.

[76] A. Lourenço, H. Silva, C. Carreiras, and A. Fred, "Outlier detection in non-intrusive ECG biometric system", in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2013, pp. 43–52, ISBN: 978-3-642-39094-4. DOI: 10.1007/978-3-642-39094-4_6.

[77] Fernández, *Learning from Imbalanced Data Sets*, 1st ed. 2018 edition. New York, NY: Springer, Nov. 1, 2018, 396 pp., ISBN: 978-3-319-98073-7.

[78] F. J. Pulgar, F. Charte, A. J. Rivera, and M. J. del Jesus, "Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips and guidelines", *Information Fusion*, vol. 54, pp. 44–60, Feb. 1, 2020, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.07.004.

[79] F. J. Pulgar, F. Charte, A. J. Rivera, and M. J. d. Jesus, “AEkNN: An AutoEncoder kNN-based classifier with built-in dimensionality reduction”, *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 436–452, Feb. 28, 2019, Publisher: Atlantis Press, ISSN: 1875-6883. DOI: 10.2991/ijcis.2018.125905686.

[80] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion”, *Journal of Machine Learning Research*, vol. 11, no. 110, pp. 3371–3408, 2010, ISSN: 1533-7928.

[81] W. Liu, P. Pokharel, and J. Principe, “Correntropy: A localized similarity measure”, in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, ISSN: 2161-4407, Jul. 2006, pp. 4919–4924. DOI: 10.1109/IJCNN.2006.247192.

[82] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, *Contractive Auto-Encoders: Explicit Invariance During Feature Extraction*. Jan. 1, 2011, Journal Abbreviation: Proceedings of the 28th International Conference on Machine Learning, ICML 2011 Publication Title: Proceedings of the 28th International Conference on Machine Learning, ICML 2011.

[83] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, 1st ed. 2013, Corr. 2nd printing 2018 edition. New York: Springer, May 17, 2013, 613 pp., ISBN: 978-1-4614-6848-6.

[84] A. Draghici and J. A. Taylor, “The physiological basis and measurement of heart rate variability in humans”, *Journal of physiological anthropology*, vol. 35, p. 22, Sep. 28, 2016. DOI: 10.1186/s40101-016-0113-7.

[85] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE”, *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

Appendix A

Autoencoder Code Implementations

Standard Autoencoder

```
n_bottleneck = round(n_input*(1-compression))

inputs = Input(shape=(n_input,))
bottleneck = Dense(n_bottleneck,name='encoded')(inputs)
outputs = Dense(n_input, activation='linear')(bottleneck)

sae = Model(inputs=inputs, outputs=outputs)
opt = Adam(learning_rate=0.001)
sae.compile(optimizer=opt, loss='mse')
```

Robust Autoencoder

```
from keras import backend as K
n_bottleneck = round(n_input*(1-compression))

inputs = Input(shape=(n_input,))
bottleneck = Dense(n_bottleneck, name='encoded')(inputs)
outputs = Dense(n_input, activation='linear')(bottleneck)

rae = Model(inputs=inputs, outputs=outputs)
opt = Adam(learning_rate=0.001)

def correntropy_loss(y_true, y_pred):
    sigma = 1
    alpha = y_pred - y_true
    kernel = (1 / (np.sqrt(2*np.pi*sigma))) * K.exp(-K.square(alpha)/(2*sigma*sigma))

    return -K.sum(kernel)

rae.compile(optimizer=opt, loss=correntropy_loss, metrics=['mse'])
```

Contractive Autoencoder

```
from keras import backend as K
n_bottleneck = round(n_input*(1-compression))

inputs = Input(shape=(n_input,))
bottleneck = Dense(n_bottleneck, name='encoded')(inputs)
outputs = Dense(n_input, activation='linear')(bottleneck)

cae = Model(inputs=inputs, outputs=outputs)
opt = Adam(learning_rate=0.001)

def correntropy_loss(y_true, y_pred):
    sigma = 1
    alpha = y_pred - y_true
    kernel = (1 / (np.sqrt(2*np.pi*sigma))) * K.exp(-K.square(alpha)/(2*sigma*sigma))

    return -K.sum(kernel)

cae.compile(optimizer=opt, loss=correntropy_loss, metrics=['mse'])
```

Denoising Autoencoder

```
n_bottleneck = round(n_input*(1-compression))

inputs = Input(shape=(n_input,))
noise = GaussianNoise(noise_sd)(inputs)
bottleneck = Dense(n_bottleneck, activation='sigmoid', name='encoded')(noise)
outputs = Dense(n_input, activation='linear')(bottleneck)

dae = Model(inputs=inputs, outputs=outputs)
opt = Adam(learning_rate=0.001)
dae.compile(optimizer=opt, loss='mse')
```

Sparse Autoencoder

```
n_bottleneck = round(n_input*(1-compression))

inputs = Input(shape=(n_input,))
bottleneck = Dense(n_bottleneck,name='encoded', activity_regularizer=regularizers.l1(10e-5))(inputs)
outputs = Dense(n_input, activation='linear')(bottleneck)

spae = Model(inputs=inputs, outputs=outputs)
opt = Adam(learning_rate=0.001)
spae.compile(optimizer=opt, loss='mse')
```

Variational Autoencoder

```
from keras import backend as K
n_bottleneck = round(n_input*(1-compression))
n_intermediate = round((n_bottleneck+n_input)/2)

def sampling(args):
    z_mean, z_log_sigma = args
    epsilon = K.random_normal(shape=(n_bottleneck,))
    return z_mean + K.exp(z_log_sigma) * epsilon

inputs = Input((n_input,))
encoded = Dense(n_intermediate)(inputs)

z_mean = Dense(n_bottleneck)(encoded)
z_log_sigma = Dense(n_bottleneck)(encoded)
z = Lambda(sampling,output_shape=(n_bottleneck,))([z_mean,z_log_sigma])

decoded = Dense(n_intermediate)(z)
outputs = Dense(n_input,activation='linear')(decoded)

vae = Model(inputs=inputs, outputs=outputs)

encoder = Model(inputs=inputs, outputs=[z_mean,z_log_sigma,z],
                    name='encoder')

regularization = 10
def vae_loss(input_img, output):
    reconstruction_loss = K.sum(K.square(output-input_img))
    kl_loss = - 0.5 * K.sum(1 + z_log_sigma - K.square(z_mean) - K.square(K.exp(z_log_sigma)), axis=-1)
    total_loss = K.mean(reconstruction_loss + regularization*kl_loss)
    return total_loss

vae.compile(optimizer='adam', loss=vae_loss, metrics=['mse'])
```

Appendix B

Additional Results

B.1 Hidden Layers and Compression Level

Table B.1: Train and validation losses varying the number of hidden layers and the compression level achieved at the AE's bottleneck.

# Hidden layers	Compression level	# Epochs	Batch size	Train loss	Validation loss
1	25%	66	2000	3.39E-05	4.86E-05
1	**50%**	**80**	**2000**	**3.44E-05**	**3.22E-05**
1	75%	122	2000	3.98E-05	3.95E-05
1	90%	144	2000	2.17E-04	2.11E-04
1	95%	213	2000	9.05E+00	9.11E-04
3	25%	49	2000	2.25E-04	2.13E-04
3	50%	59	2000	2.01E-04	1.96E-04
3	**75%**	**72**	**2000**	**1.82E-04**	**1.64E-04**
3	90%	89	2000	3.26E-04	3.60E-04
3	95%	106	2000	9.63E-04	1.20E-03
5	25%	56	2000	2.48E-04	4.00E-04
5	**50%**	**61**	**2000**	**2.63E-04**	**2.79E-04**
5	75%	76	2000	2.18E-04	4.54E-04
5	90%	97	2000	3.39E-04	4.05E-04
5	95%	75	2000	1.10E-03	1.10E-03

B.2 Waveform Reconstructions

Standard Autoencoder (SAE)

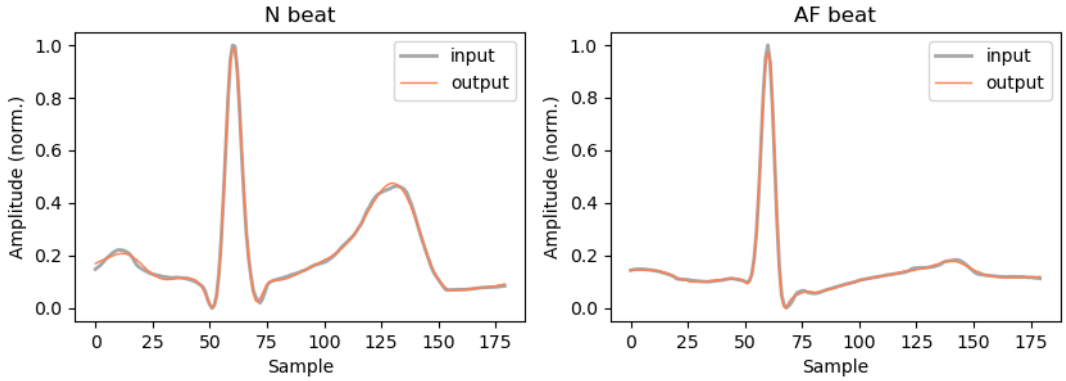


Figure B.1: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Standard Autoencoder.

Robust Autoencoder (RAE)

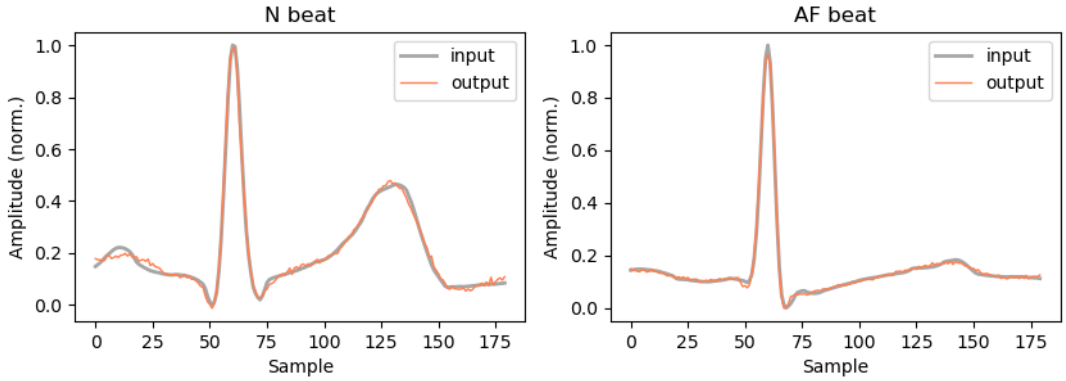


Figure B.2: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Robust Autoencoder.

Contractive Autoencoder (CAE)

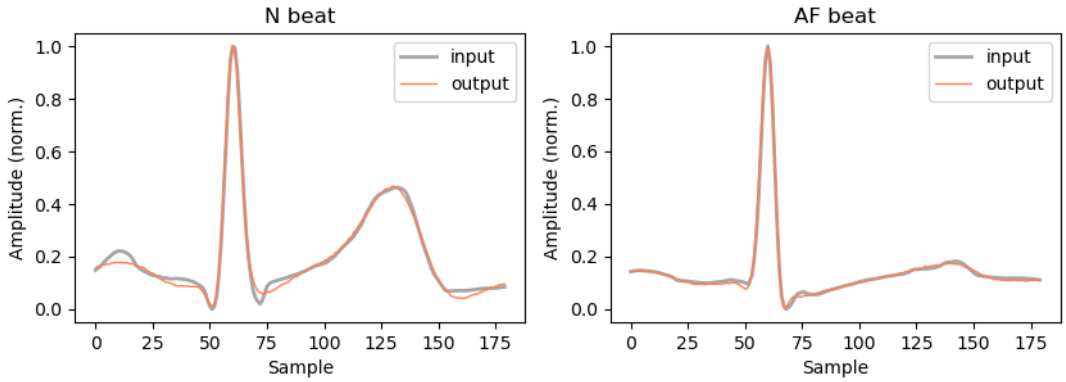


Figure B.3: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Contractive Autoencoder.

Denoising Autoencoder (DAE)

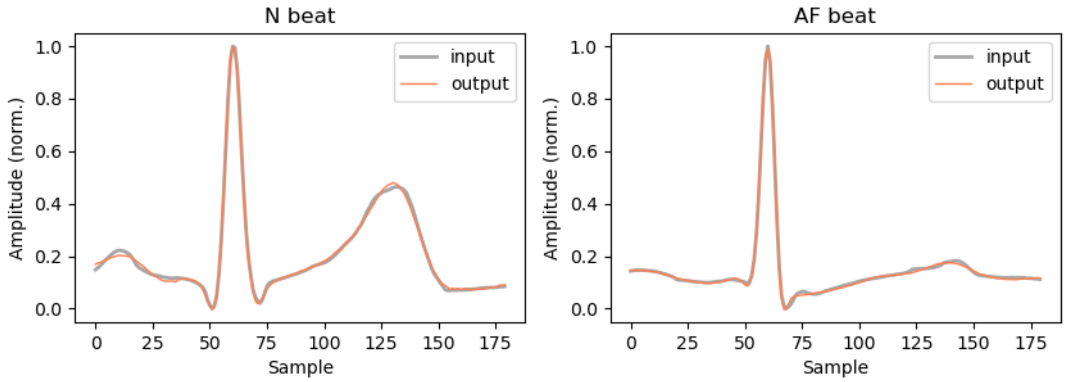


Figure B.4: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Denoising Autoencoder.

Sparse Autoencoder (SpAE)

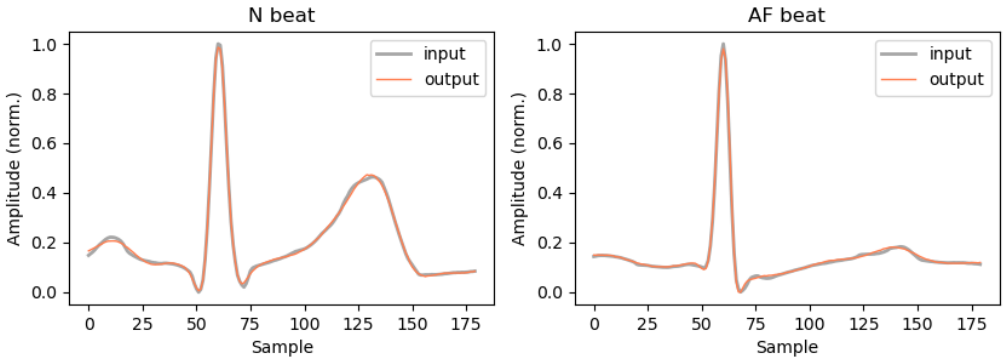


Figure B.5: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Sparse Autoencoder.

Variational Autoencoder (VAE)

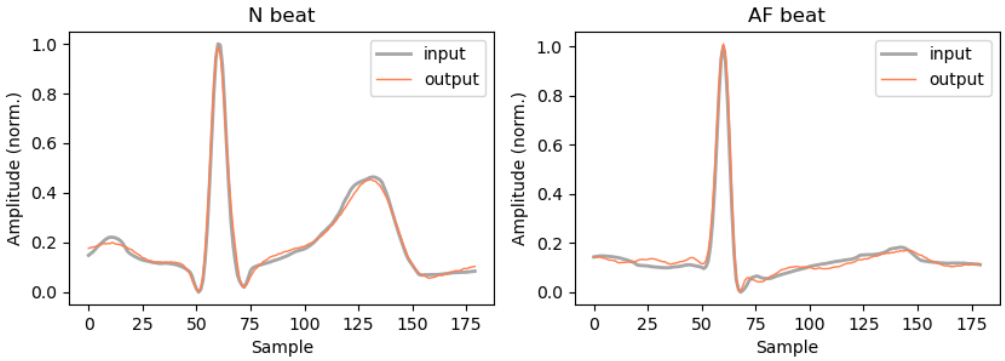


Figure B.6: Reconstruction examples of ECG waveforms in NSR (left) and AF (right) using the Variational Autoencoder.

B.3 Latent Features Distributions

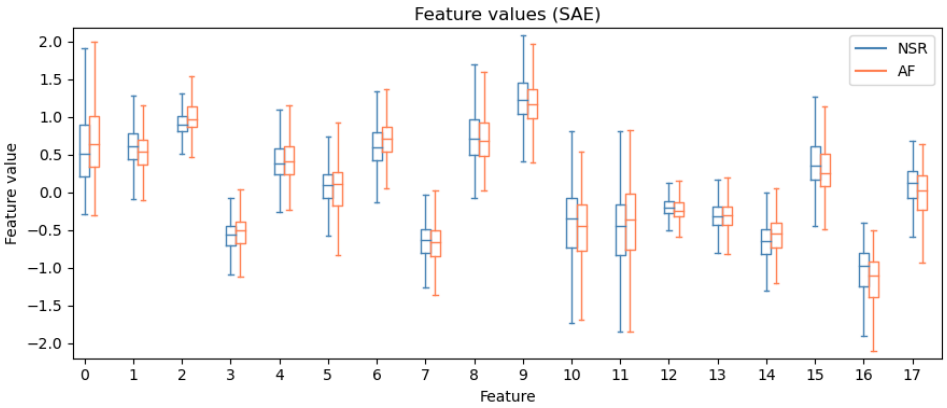


Figure B.7: Feature value distributions of the hidden features generated by the SAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

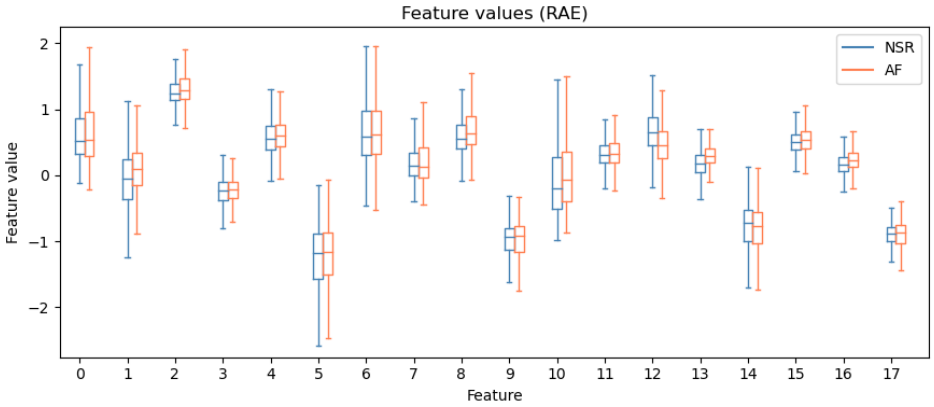


Figure B.8: Feature value distributions of the hidden features generated by the RAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

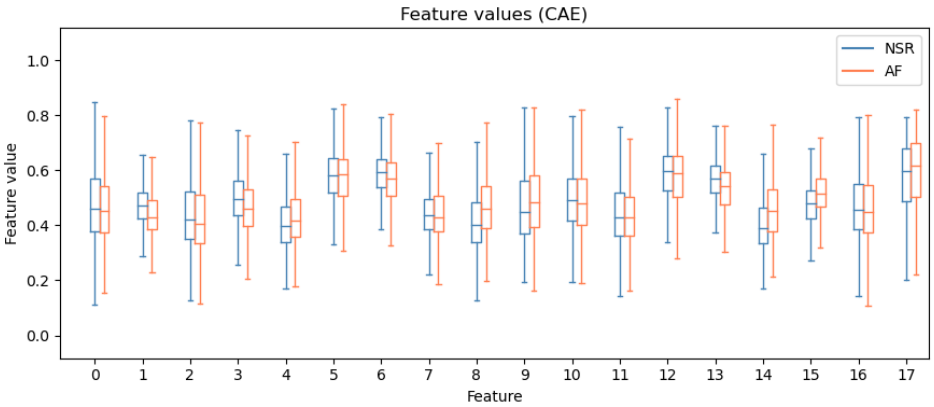


Figure B.9: Feature value distributions of the hidden features generated by the CAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

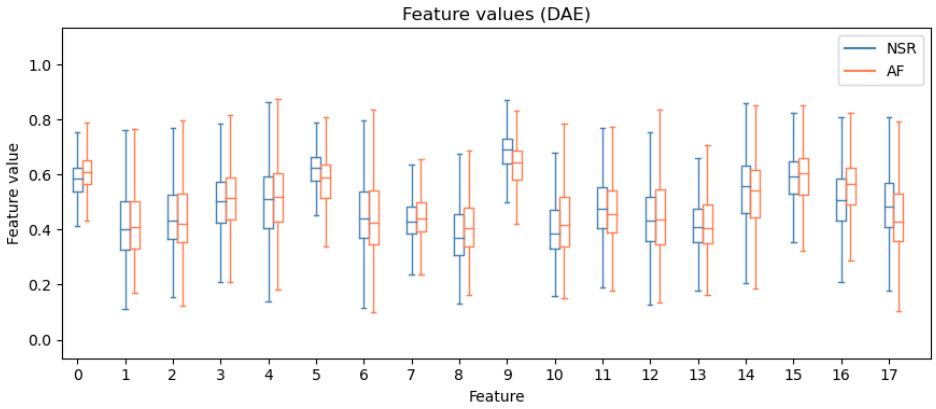


Figure B.10: Feature value distributions of the hidden features generated by the DAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

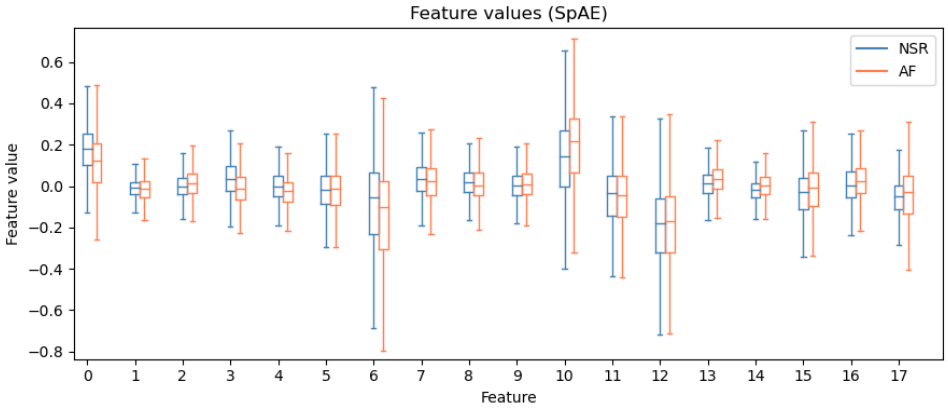


Figure B.11: Feature value distributions of the hidden features generated by the SpAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

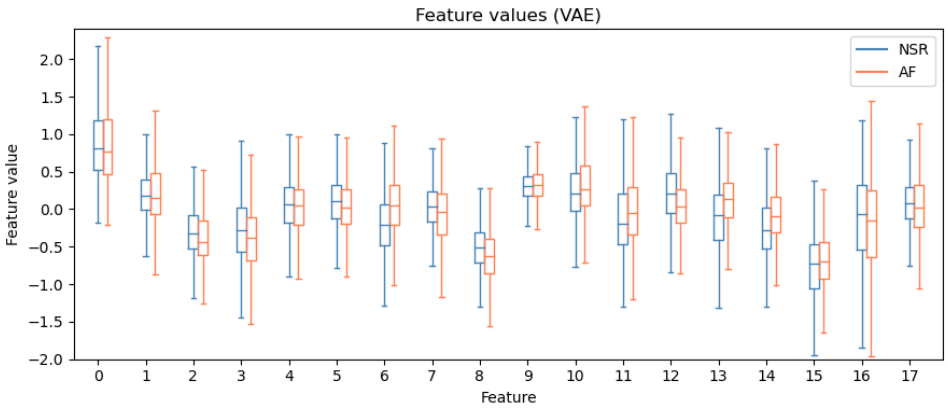


Figure B.12: Feature value distributions of the hidden features generated by the VAE model for NSR and AF test instances. Considerable overlap between the classes' feature distributions is seen.

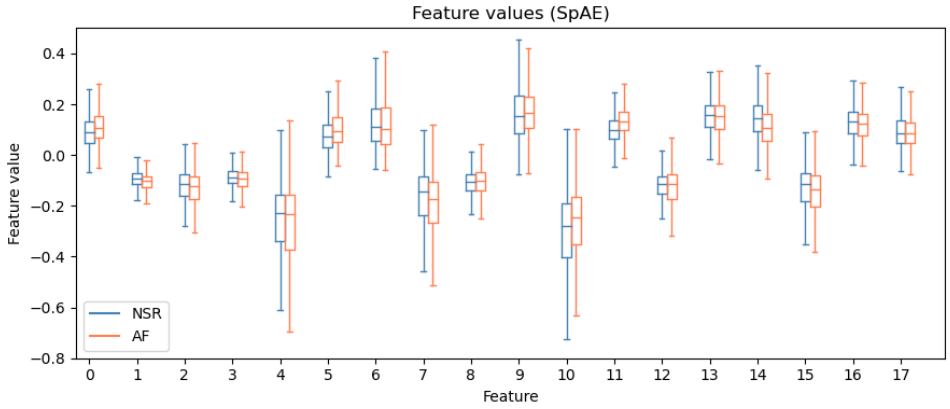


Figure B.13: Feature value distributions of the hidden features generated by the SpAE model for NSR and AF rhythms, using the LeakyReLU activation function.

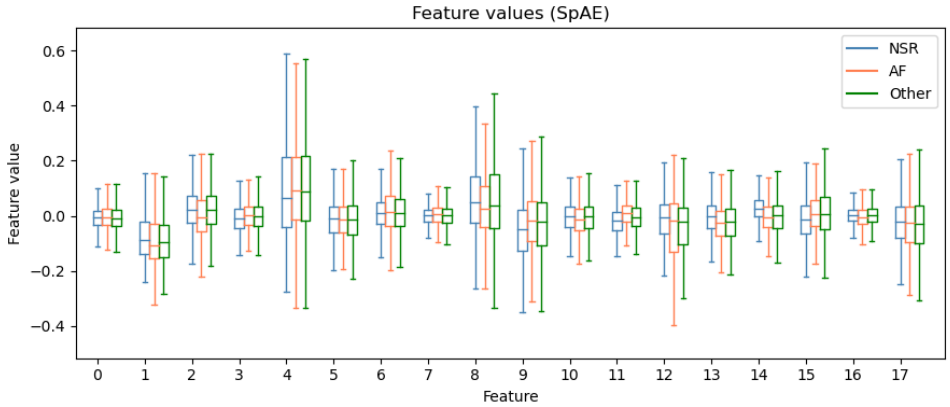


Figure B.14: Latent space generated by the SpAE model for NSR, AF, and other rhythms.

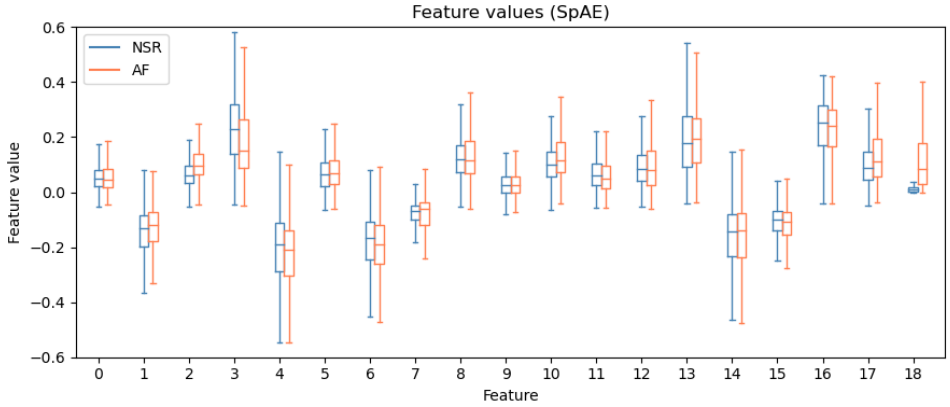


Figure B.15: Feature value distributions of the hidden features generated by the SpAE/LCSD model for NSR and AF test instances. The feature number 18 corresponds to the LCSD distributions, which present significant mean and value range differences.

B.4 t-SNE Plots

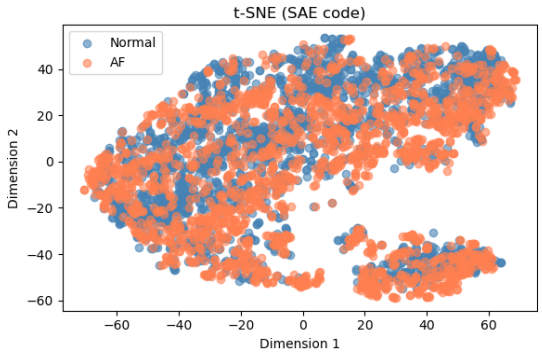


Figure B.16: 2-Dimensional t-SNE plot from a 4000-sample data generated by the SAE.

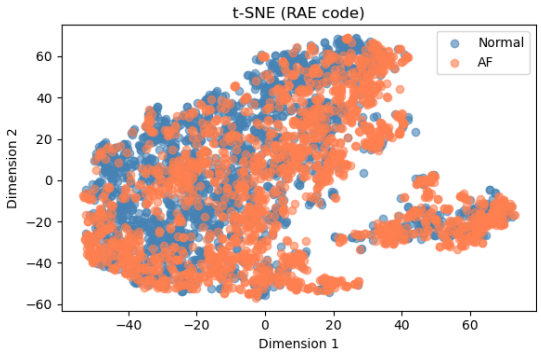


Figure B.17: 2-Dimensional t-SNE plot from a 4000-sample data generated by the RAE.

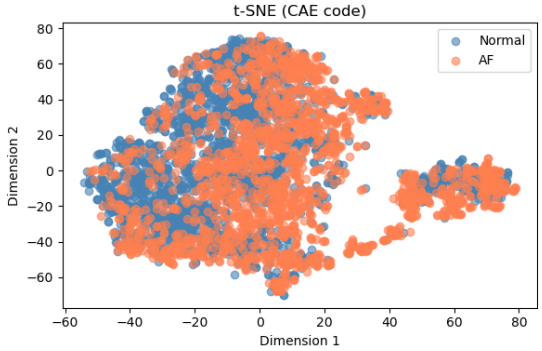


Figure B.18: 2-Dimensional t-SNE plot from a 4000-sample data generated by the CAE.

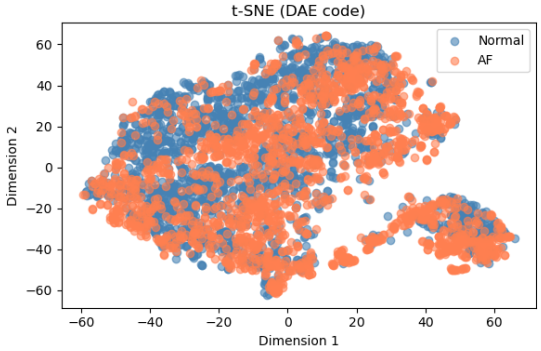


Figure B.19: 2-Dimensional t-SNE plot from a 4000-sample data generated by the DAE.

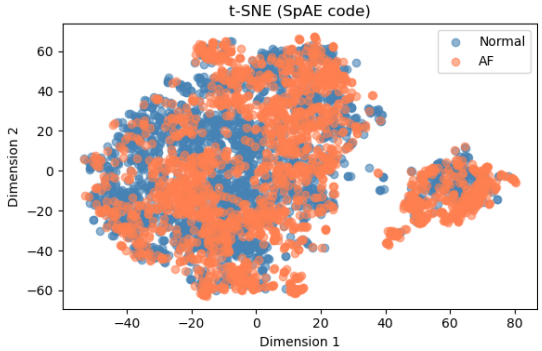


Figure B.20: 2-Dimensional t-SNE plot from a 4000-sample data generated by the SpAE.

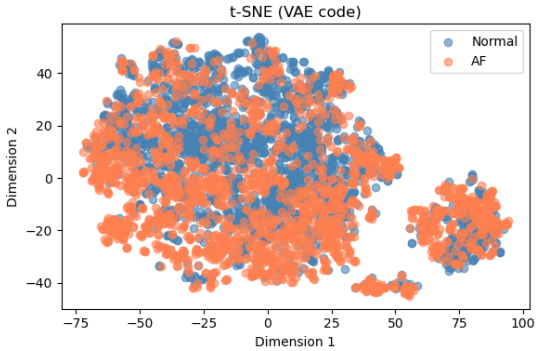


Figure B.21: 2-Dimensional t-SNE plot from a 4000-sample data generated by the VAE.