People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdès

Institute of Electrical and Electronic Engineering
Department of Electronics

Project Report Presented in Partial Fulfilment of

the Requirements of the Degree of

'MASTER'

In: Electronics

Option: Computer Engineering

Title:

Atrial Fibrillation Analysis by Deep Learning

Presented By:

- **Agli Wafa**

Supervisor:

Prof. Daamouche Abdelhamid

Registration Number: 2023/2024

Acknowledgements

In the name of ALLAH, the most gracious, the most merciful.

All deepest thanks are due to ALLAH, the merciful, the compassionate for the uncountable gifts given to me

I express profound gratitude to Professor **Abdelhamid Daamouche**, my esteemed supervisor, for affording me the opportunity to enroll as one of his students. His invaluable guidance and insightful critiques have motivated me to dedicate the utmost diligence to this endeavor, resulting in the originality of my work. Under his mentorship, I have gained authentic research experience and significantly expanded my understanding of the subject matter.

I extend my gratitude to **Charaf eddine** his willingness to patiently address my inquiries, despite the complexities inherent in the subject matter. He has greatly contributed to my understanding and progression. I am truly grateful for his mentorship and the wealth of knowledge he has shared, which has undoubtedly enriched my journey in AI.

Many thanks to all my teachers at the Institute of Electrical and Electronic Engineering

Finally, my thanks go to who have contributed in accomplishing this work, either from near or far, may Allah grant them all the best

Dedication

This study is wholeheartedly dedicated to me at first and to my beloved family.

Wafa, This work is dedicated to you, the dreamer who never let go of the belief that one day you would soar to great heights. Through years of tireless effort and unwavering determination, you followed your passion, constantly striving to learn and improve. Each stumble became a stepping stone, and each failure was met with a resilient comeback, proving time and again that you could rise above any challenge.

Today, as I stand here, having completed my master's degree, I proudly say, "This is just the beginning." The journey ahead is bright, filled with endless possibilities and boundless success. They always said, **"Believe in yourself, and the world will follow."** Here I am, a testament to that faith and perseverance, ready to embrace the future with the same spirit and tenacity that brought me here.

Family, To my dear mom who was standing by my side through every step forward. Her unwavering encouragement and constant presence have been the driving force behind my achievements. Thank you, Mom, for your boundless love and support. To my father -May Allah be Glory to Him, saves his soul-. To my dear sister, who has always been my guiding star. You have been an inspiring example, and I say with joy, "I'm right behind you!" Thank you for leading the way and showing me what is possible. To my entire family, your unconditional love and support have been my foundation. I am forever grateful for the love that has shaped me into the person I am today.

Last but not Least, I extend my heartfelt gratitude to Charaf for his unwavering support and assistance over this period, which has been invaluable to this project.

Abstract

Atrial fibrillation (AF), an increasingly prevalent cardiac arrhythmia, is a major contributor to stroke, heart failure, and premature mortality. Traditional manual screening for AF using electrocardiography (ECG) is not only time-consuming but also susceptible to human error, underscoring the urgent need for automated diagnostic tools. This study addresses this challenge by developing advanced computer-aided diagnostic methods leveraging deep learning for the automatic detection of AF.

We introduce innovative one-dimensional (1D) and two-dimensional (2D) convolutional neural network (CNN) models specifically designed for the precise classification of ECG signals into normal or atrial fibrillation categories. Our methodology includes a meticulous preprocessing phase where each ECG record is filtered and peaks are accurately detected using the XQRS algorithm. The signals are then segmented into beats with an 80-sample window, which serve as critical features for subsequent classification.

The extracted features are fed into our CNN architectures for classification. The models are trained and evaluated using the MIT-BIH Atrial Fibrillation Database, and their generalization capability is further validated with unseen data from the PhysioNet/Computing in Cardiology Challenge 2017 database, following an inter-subject approach. To enhance the robustness of our models, we employ data augmentation techniques.

Our comprehensive evaluation demonstrates that the 1D-CNN model achieves a remarkable total accuracy of 95% and an F1 score of 96.81%, while the 2D-CNN model attains an exceptional accuracy and F1 score of 99.84%. These results underscore the efficacy of our approach in accurately classifying ECG signals and highlight the potential of our models for real-world clinical applications, offering a substantial improvement in AF screening processes.

Contents

Acronyms

ACC American College of Cardiology

AF Atrial Fibrillation

AFib Atrial Fibrillation

AHA American Heart Association

AI Artificial Intelligence

ANN Artificial Neural Network

AV Atrioventricular

CNN Convolutional Neural Network

CPU Central Processing Unit

CVDs Cardiovascular Diseases

DL Deep Learning

DNN Deep Neural Network

ECG Electrocardiogram

EKG Electrocardiogram

ESC European Society of Cardiology

F1 F1 Score

GRU Gated Recurrent Unit

HDF5 Hierarchical Data Format version 5

HRV Heart Rate Variability

iBeat Intelligent Beat

LA Left Atrium

LSTM Long Short-Term Memory

LV Left Ventricle

ML Machine Learning

MIT Massachusetts Institute of Technology

MIT-BIH AFDB MIT-BIH Atrial Fibrillation Database

NSR Normal Sinus Rhythm

PCA Principal Component Analysis

PhysioNet Physiological Data Networking

PV Pulmonary Veins

RA Right Atrium

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

RR R-R Interval

SVT Supraventricular Tachycardia

VCG Vectocardiogram

VS Code Visual Studio Code

WFDB WaveForm DataBase

WHO World Health Organization

List of Figures

List of Tables

Listings

General Introduction

Cardiovascular disease (CVD) remains the leading cause of mortality worldwide, responsible for approximately 17.9 million deaths annually, which constitutes 31% of all global fatalities, according to the World Health Organization (WHO). Among the various cardiovascular disorders, atrial fibrillation (AF) is particularly prevalent, especially among the elderly. AF is characterized by irregular and often rapid heart rhythms, which significantly contribute to increased morbidity and mortality. Timely and accurate diagnosis of AF is therefore paramount for effective prevention and management.

Notably, about 25% of individuals suffering from atrial fibrillation are asymptomatic, which complicates early detection. For these asymptomatic patients, electrocardiography (ECG) serves as a crucial diagnostic tool. The ECG records the heart's electrical activity, and in cases of AF, it reveals a distinctive pattern that helps differentiate this arrhythmia from other cardiac conditions.

While several wearable ECG monitoring devices, such as Android Wear, Apple Watch, and Kardia, are designed for the precise detection of atrial fibrillation, these devices often come with significant drawbacks. They are typically expensive, complex to use, and require extended periods of monitoring to yield accurate results. These limitations underscore the need for more accessible and efficient diagnostic tools.

In response to these challenges, the development of computer-aided diagnostic tools that leverage deep learning technologies has become increasingly essential for the automatic detection of atrial fibrillation. This master's thesis introduces innovative convolutional neural network (CNN) models specifically designed to accurately detect AF. Our approach focuses on using a single beat, represented by 80 samples, as the

primary feature for optimal performance. The efficacy and generalization of these models are then validated on a completely unseen dataset, ensuring their applicability across diverse subject groups.

This dissertation is organized into three comprehensive chapters, in addition to an introduction and a general conclusion. The structure is as follows:

1. **Chapter 1: Atrial Fibrillation (AF):** This chapter provides an in-depth overview of atrial fibrillation, elucidating the condition with clear scientific explanations. It covers the symptoms, risk factors, and diagnostic methodologies, with particular emphasis on the distinctive behavior of the ECG signal in the presence of AF.

2. **Chapter 2: State of The Art:** The second chapter delves into the latest and most effective studies related to the detection of AF through ECG signal analysis. This review highlights various methodologies and their respective efficacy in AF diagnosis.

3. **Chapter 3: Proposed Methodology and Experimental Results:** The third chapter details our proposed CNN models, including the strategies for segmentation and denoising of ECG signals. It discusses the optimization of these models through a combination of relevant input features and the extracted features from the CNN. The chapter further presents and analyzes the results of our experiments, demonstrating the effectiveness of the proposed approach.

The thesis concludes with a general summary that encapsulates our findings and outlines future research directions, with the aim of further advancing the field of automated AF detection.

Chapter 01

Atrial Fibrillation

Chapter 1

Atrial Fibrillation

1.1 Introduction

According to the World Health Organisation (WHO), Cardiovascular diseases (CVDs) are the leading cause of death globally (around 17.9 million lives each year)[1]. Atrial Fibrillation or AFib is known as a major CVD, involving a large number of population. It affects in particular old people, especially those who suffer from heart failure (a major cause of hospitalisation), which leads to a serious complications such as **Blood clots**, **stroke**, **Death**. With proper medical supervision and medication, atrial fibrillation (AFib) is manageable. Early diagnosis and treatment can help prevent associated complications [2].

This chapter contains a detailed overview of Atrial Fibrillation, its symptoms, causes, and risks, and how can the early detection of atrial fibrillation prevent serious consequences such as sudden death. In addition, we spotlight on electrocardiogram (ECG) and its uses on AF detection.

1.2 Background: The Heart

1.2.1 Cardiac Anatomy

The heart is a muscle organ found in the mediastinum between the lungs, its primary job is to pump blood into the circulatory system. It is constructed by a bundle of muscle divided into four main chambers: two upper chambers called atria and two lower chambers called ventricles as shown in Figure 1-1. The right side of the heart contains the right atrium (RA) and the right ventricle (RV), while the left side contains the left atrium (LA) and the left ventricle (LV). These chambers are divided by continuous partitions, such as the interatrial septum between the LA and RA, and the interventricular septum between the LV and RV [3].

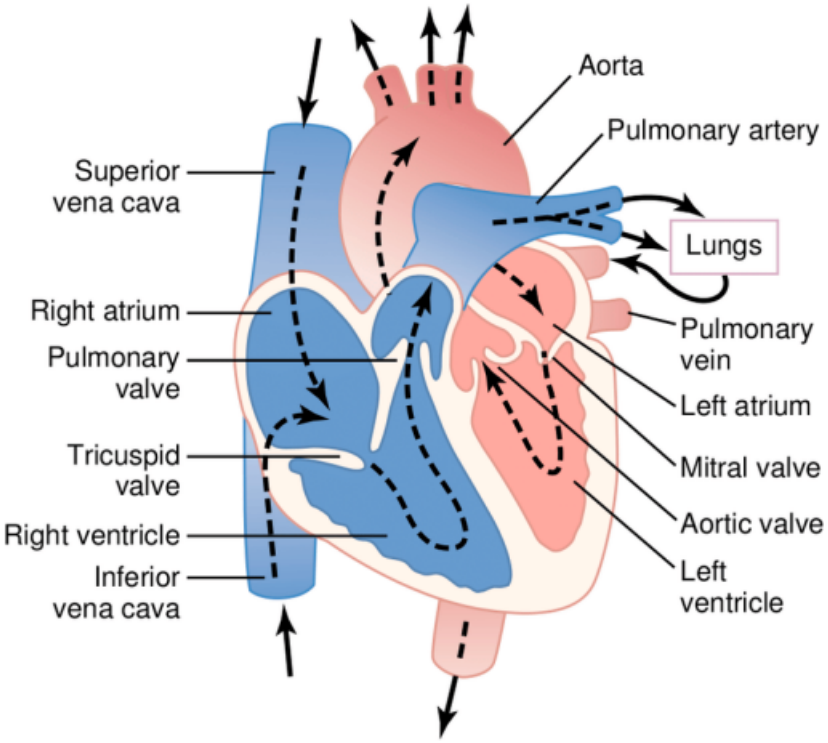


Figure 1-1: Schematic representation of cardiac anatomy [4]

Additionally, the atria and ventricles are further divided by the atrioventricular septum. Blood flows from the atria to the ventricles through two atrioventricular

orifices, which are openings in the atrioventricular septum. These openings are regulated by two atrioventricular valves that open and close rhythmically during each heartbeat. The tricuspid valve controls the flow between the right atrium (RA) and right ventricle (RV), while the mitral valve controls the flow between the left atrium (LA) and left ventricle (LV). The LA and RV are connected to the lungs through the pulmonary veins (PV) and pulmonary arteries, respectively. On the other hand, the LV and RA are connected to the rest of the body through the aorta and the superior/inferior vena cava, respectively. To prevent the backflow of blood, both the pulmonary artery and the aorta are equipped with valves.

1.2.2 Understanding the Role of The Heart

The heart serves as **a vital pump,** orchestrating the circulation of blood throughout the body through a rhythmic sequence of contractions and relaxations in its chambers, known as the cardiac cycle [5]. This cycle begins with atrial diastole, during which the atrial chambers relax, allowing deoxygenated blood to enter the right atrium (RA) through the vena cavae and oxygenated blood to enter the left atrium (LA) through the pulmonary veins (PVs). Next, atrial systole occurs, characterized by the contraction of the atrial chambers, which propels blood through the atrioventricular valves into the ventricles. While the atria undergo diastolic and systolic phases, the ventricles remain in a state of diastole, allowing blood to accumulate from the atria. Following this, the ventricles undergo systole, expelling blood from the left ventricle (LV) into the aorta for distribution throughout the body, and from the right ventricle (RV) into the pulmonary arteries to be transported to the lungs.

1.3 The Electrocardiogram

1.3.1 Flashback: The First Electrocardiogram

Birth of ECG: Heartbeat tracking has a long history, reaching back to the late 1700s, and was further developed by Dutch scientist Willem Einthoven, who was

awarded the Nobel Prize for his innovative medical technology. Gabriel Lippman's development of the capillary electrometer, which measures voltage variations on the skin caused by cardiac pulses, marked a turning point in the field in 1872. In 1887, A.D. Waller used Lippman's apparatus to record the first heartbeat measurement[6]. However, the earlier researchers failed to account for capillary friction and inertia, which led to errors in their recordings. In 1901, Willem Einthoven made this right. At first, he used mathematical methods to manually repair defects in the recognizable peaks and valleys on photographic paper. Later, he automated the correction and transformed the procedure by creating the string galvanometer as shown in Figure 1-2.

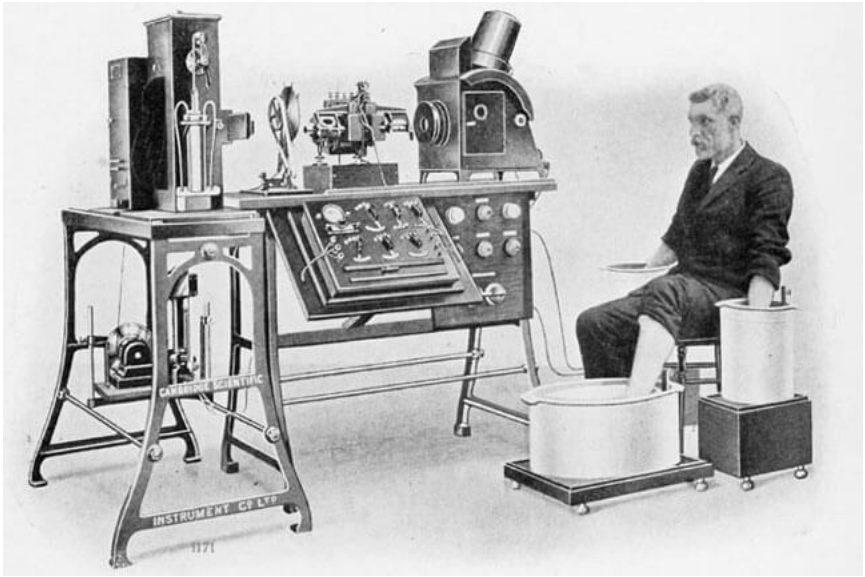


Figure 1-2: Heartbeat Demonstrated: Willem Einthoven's Magnificent 600-Pound Machine [6]

1.3.2 Definition

An electrocardiogram (EKG or ECG) is a painless procedure that measures the heart's electrical activity [7]. Where small electrodes are placed on the skin of the chest wrists, ankles and connected in a specific order to a machine that, when turned on, measures electrical activity all over the heart and transforms the signals into patterns or waves [8]. The ECG is a useful tool for diagnosing heart rhythm abnormalities, such as atrial fibrillation.

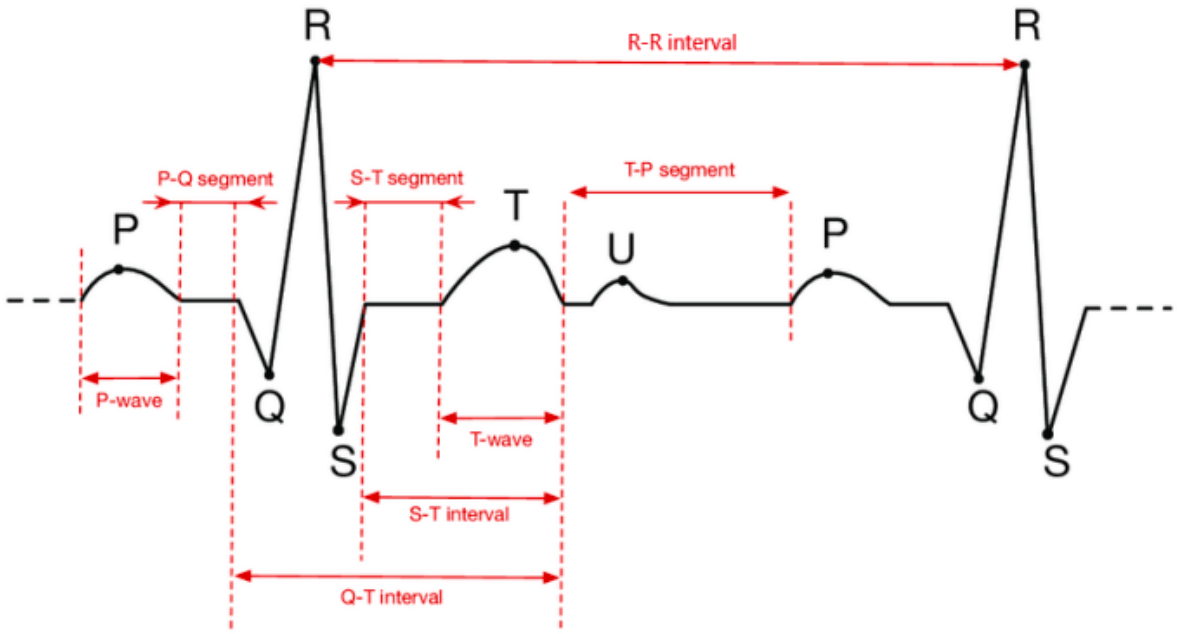


Figure 1-3: Graphic description of the different parts conforming an ECG signal during two normal sinus rhythm heartbeats [9]

The number of electrodes used varies depending on the specific information being measured, and the resulting potentials are then combined to create the ECG lead. Each lead provides a unique perspective on the electrical activity of the heart. The placement of the electrodes has a direct impact on the shape and polarity of the waves in the ECG signal. For example, a current moving away from the electrode would result in a negative slope, while a depolarizing wavefront moving towards the electrode would produce a positive deflection on the signal.

The typical physiological NSR beat is described by the P wave, the PQ interval, QRS complex, J point, ST segment, QT interval, T wave, and RR interval, as displayed in Figure 1-3. Each of these elements reflects a specific event:

- P wave: atrial depolarization. Typically presents an amplitude lower than 300 µV and a duration of less than 120 ms.
- PR segment: delay between atrial and ventricular depolarization. It measures

the speed of the AP transition through the AV node. A normal PQ interval ranges between 0.12 to 0.22 s

- QRS complex: ventricular depolarization. Typically of a duration less than 0.10 s. During this phase atrial repolarization also takes place but it is masked by the much larger amplitude of ventricular depolarization.
- J point: onset of ventricular repolarization
- ST segment and T wave: ventricular repolarization. There is no cardiac muscle activity during this wave.

1.3.3 Lead systems

Each ECG lead records the electrical activity of the heart from a distinctive axis projection. As a result, each lead collects a unique spatial perspective or cardiac electrical activity. There are two types of leads used to record ECGs: unipolar and bipolar. While unipolar leads detect the voltage fluctuation of a single electrode relative to a reference, Bipolar leads measure the voltage differential between two electrodes.
The number and position of electrodes are determined by the exploration's goal and duration. In clinical settings, a 12-lead ECG is commonly utilized for 10 seconds of cardiac exploration. However, for long-term ambulatory recordings, fewer electrodes are employed.

1.3.4 The Standard 12-Lead ECG

The most popular configuration for recording cardiac electrical activity is the 12-lead ECG. Ten electrodes are placed in the chest and limbs to collect it. Six of the twelve leads are precordial or chest leads, and the remaining six front-facing leads. Precordial leads assess cardiac electrical activity on the transversal plane, whereas frontal leads measure it on the frontal plane.
Three bipolar limb leads and three enhanced unipolar limb leads make up the frontal

lead. The letters I, II, and III stand for the three bipolar limb leads. The voltage differences between the left arm (VLA), right arm (VRA), and left leg (VLL) are measured in order to produce these leads using the following relations:

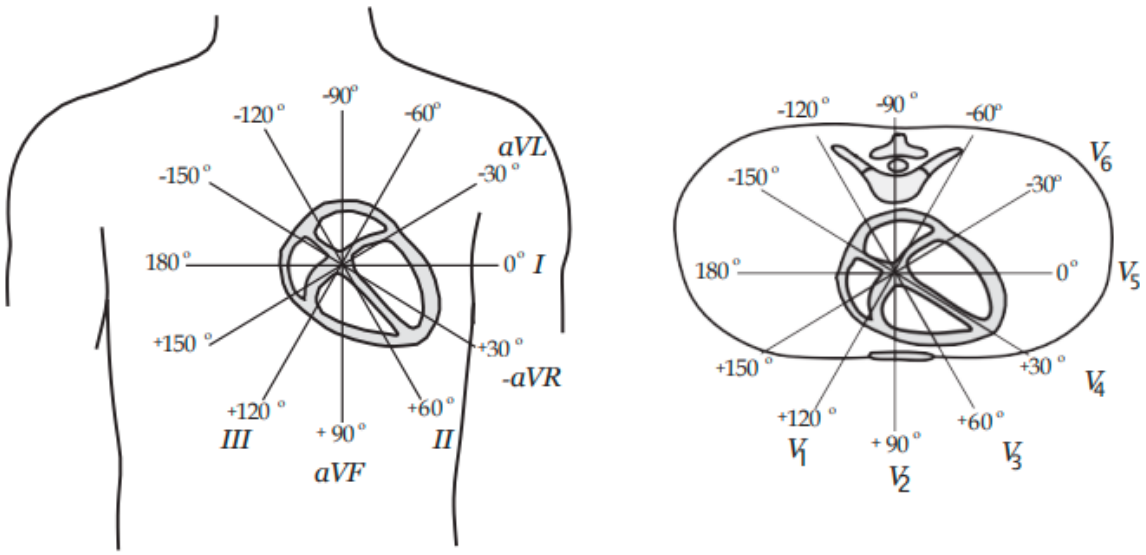


Figure 1-4: Lead angles reference system for the frontal (left) and horizontal (right) planes [10]

$$I = V_{LA} - V_{RA} \tag{1.1}$$

$$II = V_{LL} - V_{RA} \tag{1.2}$$

$$III = V_{LL} - V_{LA} \tag{1.3}$$

The triangle with the heart at its center is known as "Einthoven's triangle," and it is formed by these three leads. As shown in Figure 1-5 , lead I measures the heart's activity at 0^{o}, lead II at $+60^{o}$, and lead III at $+120^{o}$.

Instead, augmented unipolar leads, or aVF, aVL, and aVR, are designated as follows:

$$aV_F = V_{LL} - \frac{V_{LA} + V_{RA}}{2} \tag{1.4}$$

$$aV_L = V_{LA} - \frac{V_{RA} + V_{LL}}{2} \tag{1.5}$$

$$aV_R = V_{RA} - \frac{V_{LA} + V_{LL}}{2} \tag{1.6}$$

They offer a shifted 30^{o} angle view from each unipolar limb lead, complementing them. The corresponding angles for aVF, aVL, and aVR are 90^{o}, -30^{o}, and -150^{o}, respectively.

Precordial leads, also known as unipolar leads, gauge the potential difference between a chest location and an imaginary center known as Wilson Central Terminal. The calculation of this center point involves averaging VLA, VRA, and VLL. Positioned in the fourth intercostal gap to the right and left of the sternum, respectively, leads V1 and V2 face the surface of the right ventricle. V4 is positioned at the nipple line's fifth intercostal gap. V3 lies in the middle between V2 and V4. V6 is positioned on the same horizontal line as V4—the midaxillary line.

1.3.5 Ortogonal Leads

The electrical activity of the heart is projected into the X, Y, and Z planes by the orthogonal lead system. The vectocardiogram (VCG), also known as a 3D loop representation, is created by combining the various leads.

The vector's tip, which indicates the predominant direction of the cardiac wavefront, traces the loop.

The most popular electrode arrangement for obtaining the orthogonal projections is the Frank lead system. This lead system makes use of seven electrodes that are placed in the left foot, back, neck, and chest to get a left side, bottom, and front view of the heart.

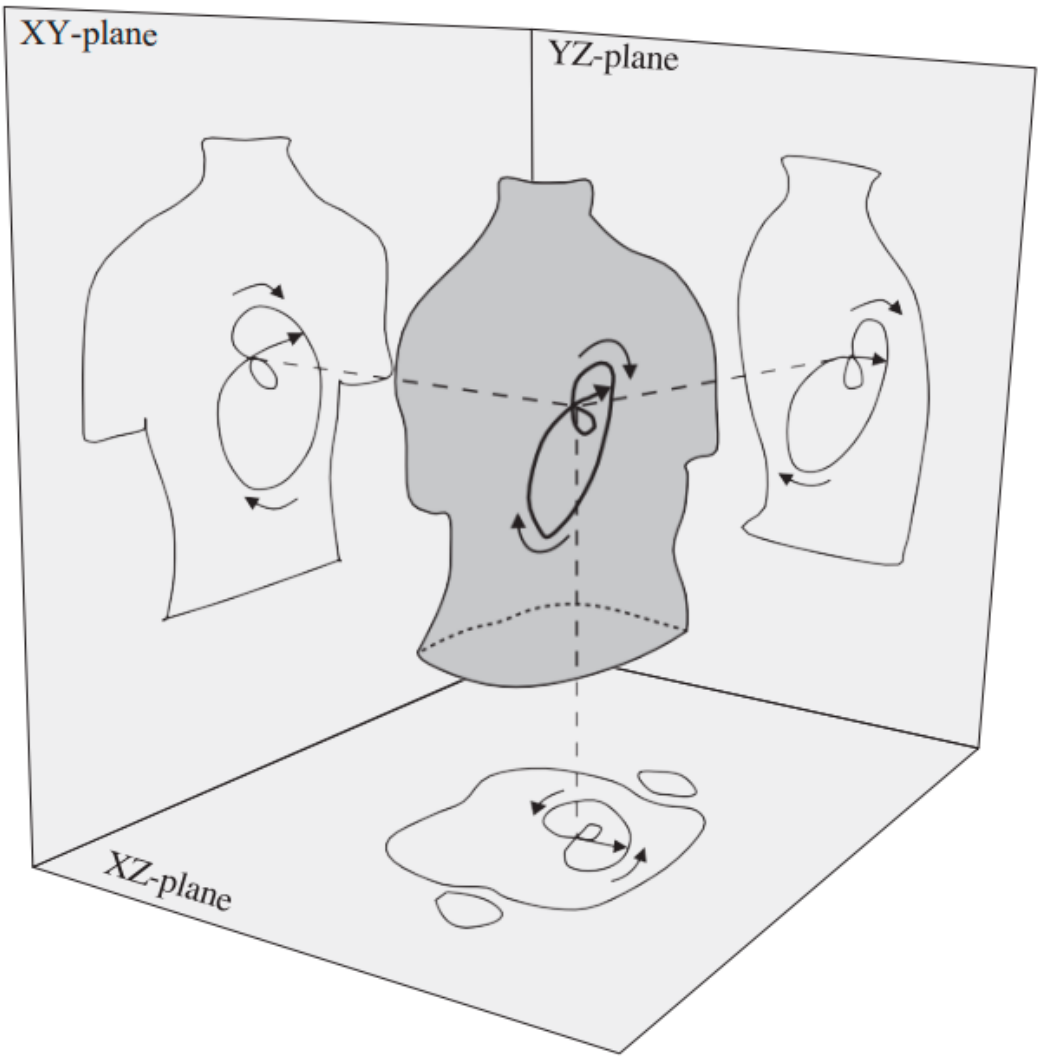


Figure 1-5: Ortogonal projections of the vectocardiographic loop [10]

1.3.6 Heart rate and heart rhythm:

- **Heart rate**

  The heart rate is defined as the number of times the heart beats in a minute. This is the number of times it pumps to push blood round the body.

- **Heart rhythm**

  Is the pattern in which the heart beats. This rhythm can be described as either

regular or irregular, and can also vary in speed, ranging from fast to slow.

1.3.7 Characteristics of the normal heart rhythm

We can divide ECG characteristics into two big categories:

1. **Morphological features:** ECG interpretation involves evaluating the morphology (look) of the waves and intervals on the ECG curve. Thus, before studying each component in depth, a brief summary of the waves and intervals is provided (View Figure 1-6)

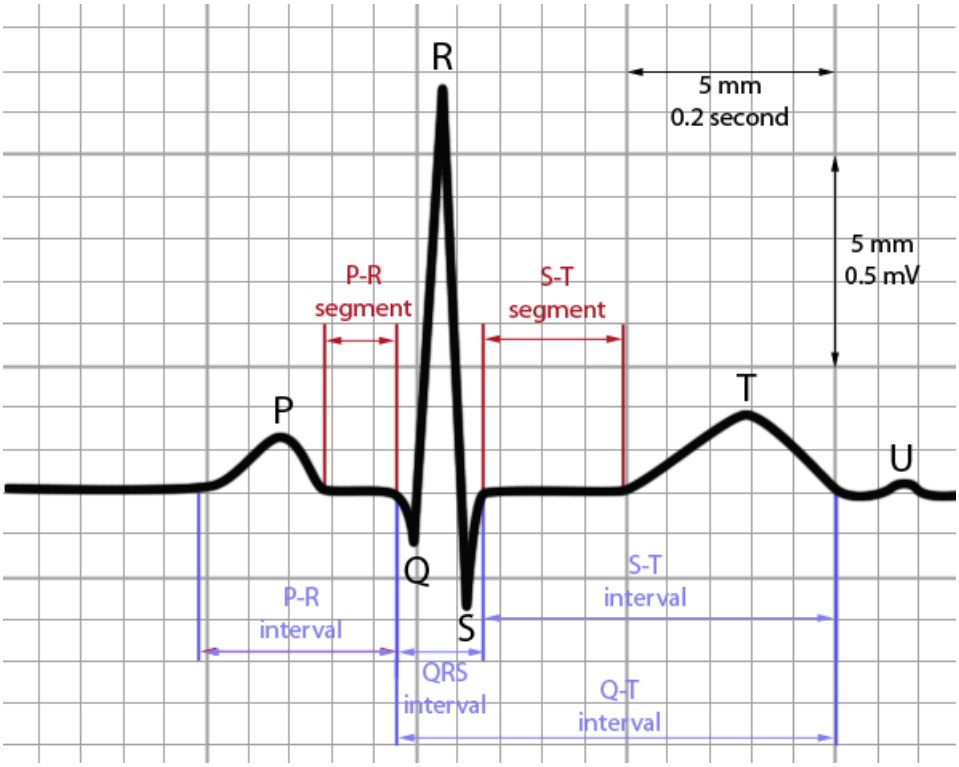


Figure 1-6: Normal ECG heart beat

Table 1.1 displays the values of the parameters in Figure1-6 that are typically observed in healthy adults:

Table 1.1: Normal heart beat waves and intervals typical values [11].

	P wave	PQ interval	QRS Complex	ST interval	QT interval	T wave
Duration (ms)	0.08 - 0.10	0.12 - 0.20	0.08	0.20	0.36	0.20
Amplitude (mv)	0.25	0	$Q \prec 0 R \succ 0 S \prec 0$	0	-	$T \succ 0$

2. **Heart rate variability features:** Heart rate variability, or HRV, is a physiological phenomenon characterized by variations in the time interval between consecutive heartbeats in milliseconds. In a normal healthy heart, there is constant variation in the milliseconds between heartbeats.

   One of the most common methods for calculating HRV is to use time or frequency.

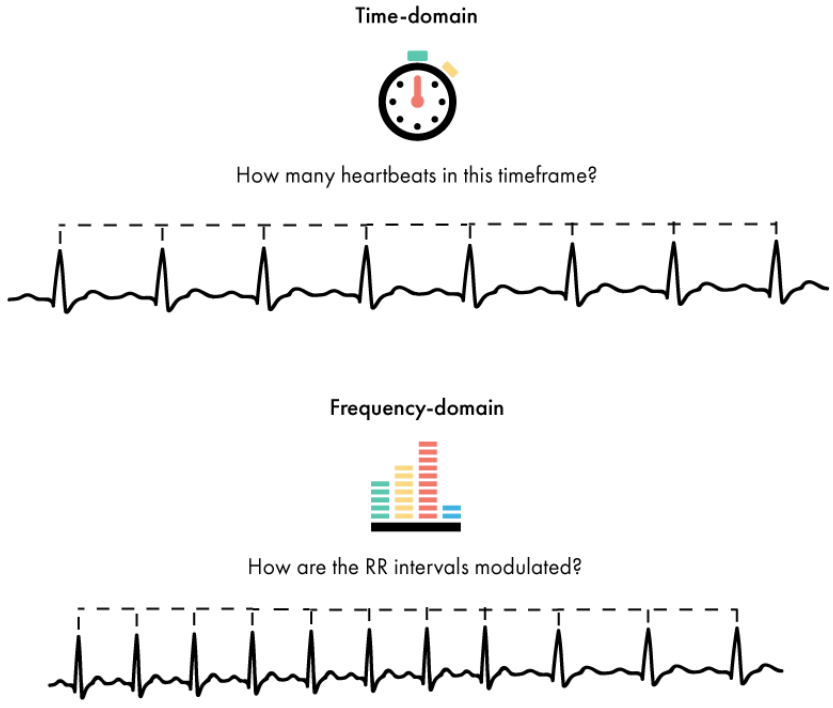


Figure 1-7: Heart rate variability domains

1.4 Heart rhythm abnormalities

In this section, we will outline the various types of arrhythmias, which are among the most complex and insufficiently studied issues in modern cardiology. This makes them one of the most urgent problems in the field.

1.4.1 Definition

Heart rhythm abnormalities, also known as arrhythmias, are a series of irregular heartbeats that can result in a heart rate that is either too slow or too fast, or irregular in nature[7].

1.4.2 Classification of arrhythmias

Shown in Figure 1-8 a summary of the various heart rhythm abnormalities that can be identified through medical diagnosis.

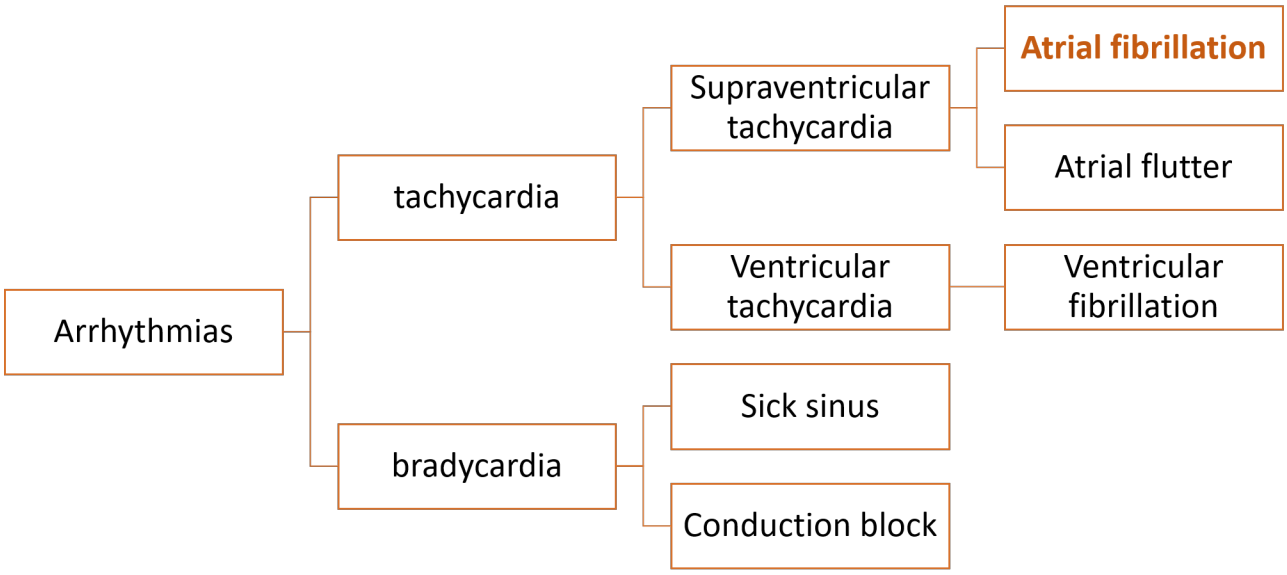


Figure 1-8: Classification of heart rhythm abnormalities

Cardiac arrhythmias can be categorized into the following groups:

1. **Bradyarrhytmias (bradycardia): Slow Heart Rate**

   Bradycardia is a heart rate that is too slow. What is considered too slow, can depend on the age and physical condition. Elderly people, for example, are more prone to bradycardia. [8].

   They are characterized by a resting heart rate < 60 beats/minutes for adults (a resting heart rate of fewer than 60 beats per minute (BPM) qualifies as bradycardia).

2. **Tachyarrhythmias (Tachycardia): Fast Heart Rate**

   Refers to a heart rate that is too fast. Generally speaking, for adults, a heart rate of more than 100 beats per minute (BPM) is considered too fast.

   (a) **Ventricular Tachycardia**: is a fast heart rate that occurs in the lower chambers (ventricles) of the heart. This type of arrhythmia may be either well-tolerated, requiring immediate diagnosis and treatment[8].

   The seriousness depends largely on whether other cardiac dysfunction is present and on the degree of the ventricular tachycardia.

   (b) **Supraventricular Tachycardia (SVT)**: Atrial or supraventricular tachycardia (SVT) is the most common type of abnormal tachycardia in adults. with a fast heart rate that starts in the upper chambers of the heart. In atrial or supraventricular tachycardia, electrical signals in the heart's upper chambers fire abnormally.

   SVTs are usually identified by an explosion of rapid heartbeats that can be chronic or begin and end suddenly. That can last a few seconds or several hours and may cause the heart to beat over than 160 times per minute. Symptoms include palpitations, chest pains, upset stomach, decreased appetite, lightheadedness or weakness.

   SVTs, or supraventricular tachycardias, often involve **atrial fibrillation** and **atrial flutter.**

1.5 Atrial Fibrillation

1.5.1 Definition

By the American College of Cardiology (ACC), the American Heart Association (AHA) and the European Society of Cardiology (ESC) atrial fibrillation is defined as *"Tachyarrhythmia characterized by mostly uncoordinated atrial activation with consequent deterioration of atrial mechanical function"*. It is the most common cardiac arrhythmia, occurring in 1-2% of the general population" [12].

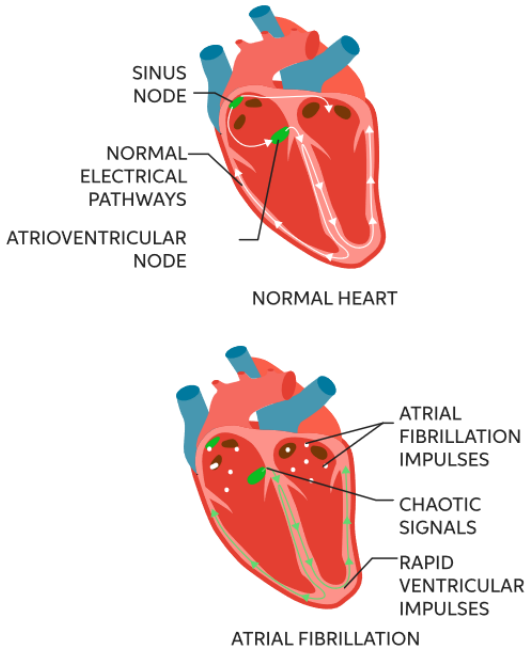


Figure 1-9: Normal Heart vs Atrial Fibrillation Heart [13]

Atrial fibrillation is a rapid heart rate caused by irregular electrical impulses in the upper chambers of the heart. These signals result in rapid, uncoordinated, weak contractions of the atria. While the heart should pump blood properly so the body gets the oxygen it needs; **What happens in** atrial fibrillation (AFib), the heart does not beat appropriately rather than pulsing in a normal pattern, the atria quiver in irregular and fast way [14].

1.5.2 Symptoms of atrial fibrillation

Approximately 25% of all individuals with afib are asymptomatic (they have no symptoms)[8] and insensible of their condition until it is often discovered first during a hospital admission or a physical examination. Those who do have atrial fibrillation often experience debilitating symptoms despite treatment such us:

- Heart palpitations, which are sensations of a racing, uncomfortable, irregular heartbeat or a flip-flopping in the chest.
- Feeling faint at times.
- Angina (i.e. chest pain or discomfort).
- Dyspnoea (i.e. Shortness of breath).
- Asthenia (Weakness; lack of energy and strength).
- Fatigue.
- Sleeping difficulties.

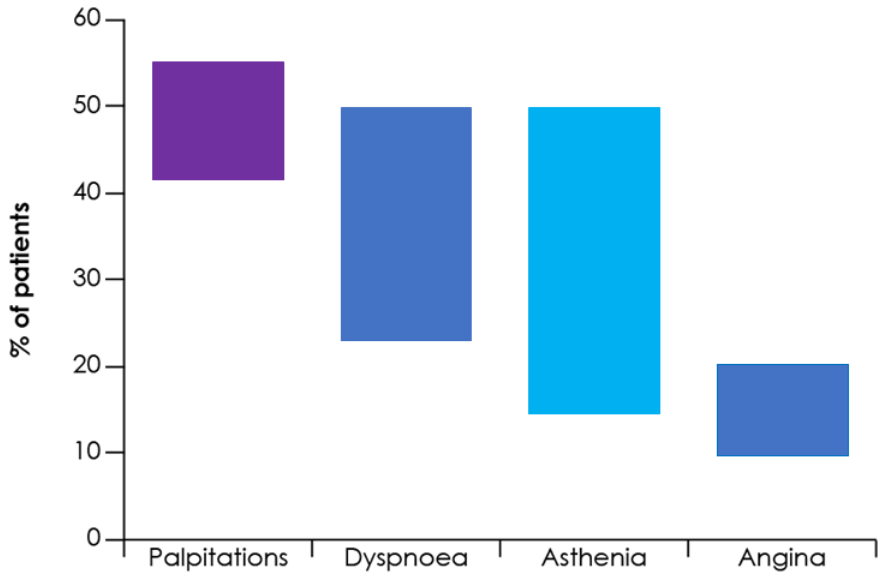


Figure 1-10: Prevalence of atrial fibrillation symptoms [15]

1.5.3 Types of atrial fibrillation

Atrial fibrillation (AF) can present in several ways and correct classification can guide the choice of treatment. It is grouped according to the duration of the arrhythmia.

Table 1.2 outlines the different classes of AF:

Table 1.2: Patterns of atrial fibrillation [16]

AF Pattern	Definition
First diagnosed	Not previously diagnosed, irrespective of arrhythmia duration or the presence and severity of symptoms
Paroxysmal	Self-terminating, usually within 48h, may continue for up to 7 days.
Persistent	Lasts more than 7 days
Long-standing persistent	Continuous AF for more than 1 year with rhythm control
Permanent	The patient and physician accepts the AF, and the rhythm is not controlled

Atrial fibrillation is a cumulative disease that usually evolves towards permanent atrial fibrillation. This is generally a stepwise process in which persons with paroxysmal atrial fibrillation head for an increasing number of episodes until the arrhythmia is persistent. Once persistent, the number of episodes with persistent atrial fibrillation tend to increase until the arrhythmia is long-standing persistent. It should be noted, however, that some patients have paroxysmal or persistent atrial fibrillation during their disease course, while others never return to sinus rhythm after a first diagnosis [17].

Other types of atrial fibrillation:

- **Lone atrial fibrillation :** is used to describe a patient younger than 60 years of age, who do not have any other concomitant heart diseases or risk factors, and whose echocardiographic examination is normal. This type of atrial fibrillation has a good prognosis and generally do not require anticoagulation therapy [18].

- **valvular atrial fibrillation:** it affects people who have valve disease or an artificial valve. Non valvular atrial fibrillation is caused by other things, such as high blood pressure.

1.5.4 Causes of atrial fibrillation

A risk factor is something that increases the risk of developing a disease or condition. The main risk factors for getting atrial fibrillation are abnormalities or damage to the heart's structure are the most common cause of atrial fibrillation [14].

Possible causes of atrial fibrillation include :

1. getting older, particularly being 65 or older.
2. High blood pressure.
3. diabetes.
4. lung cancer.
5. Heart attack.
6. Congenital heart defects.
7. Exposure to stimulants, such as medications, caffeine, tobacco or alcohol.
8. Sick sinus syndrome: improper functioning of the heart's natural pacemaker.
9. History of previous heart surgeries...

Atrial fibrillation is common in people with other heart conditions, also with other medical conditions. However, in lone atrial fibrillation, the cause is often unclear, and serious complications are rare where patients don't have any heart defects or damage.

1.5.5 Risks of having atrial fibrillation

AFib can cause potentially life-threatening health issues, We review the following

1.5.5.1 Blood Clots

Having atrial fibrillation increases the risk of developing a blood clot inside the chambers of the heart. This is because the atrial fibrillation disturbs the normal flow of blood through the heart, causing turbulence. The turbulence causes the blood to form small clots. If a clot forms in the heart, it can travel through your bloodstream and cause a stroke.

1.5.5.2 Stroke

The relationship between AF and stroke runs both ways. On the one hand, AF markedly increases the risk of stroke[19]. For instance, AF increases stroke risk five-fold compared with people without the arrhythmia[20],[19]. On the other hand, strokes increase the likelihood of developing AF[21]. For instance, older people are at increased risk: 25% of all stroke in people older than 80 years occur in AF patients[20]. Women are also at higher risk of experiencing a stroke from AF compared with men[19].

About 50% of people die within a year of a atrial fibrillation (AF)-related stroke. This compares with a mortality rate of 27% among people with strokes unrelated to AF[22]. A study from Ireland reported five-year survival rates after an AF-related stroke of 39.2%.

1.5.5.3 Heart failure

AFib can lead to heart failure, especially when the heart rate is high. When the heart rate is irregular, the amount of blood flowing from the atria to the ventricles varies for each heartbeat.

1.6 Atrial fibrillation signature in the ECG signal

Atrial fibrillation (AF) diagnosis from an electrocardiogram (ECG) hinges upon identifying several distinctive features. One prominent indicator is the absence of the P-wave on the ECG, attributable to the rapid and irregular atrial activity as shown in Figure 1-12. Instead, observers typically note low-amplitude and erratic "fibrillatory" waves occurring between the QRS complexes, indicative of disordered atrial depolarization. Additionally, irregular and relatively shortened RR intervals often manifest in ECGs with AF episodes. These irregularities stem from the erratic firing activity in the atria, which variably affects ventricular response during the cardiac cycle, as it is conducted through the atrioventricular node.

Detection and diagnosis of atrial fibrillation (AFib) typically occur non-invasively in clinical settings, primarily through the evaluation of electrocardiograms (ECGs). Key characteristics of AFib within the ECG trace include:

Table 1.3: Characteristics of ECG signal in atrial fibrillation [8]

Heart rhythm	Irregular heart rhythm (irregular RR interval)
Heart rate	Ventricular rate : 60-100 beat Atria rate : 400 - 600 beat Average heart rate : less than 100 if AF is uncontrolled
P wave	No P waves
PR intervals	No PRi intervals since there are no P waves
QRS complex	0.06 - 0.10seconds

AF can be characterised from the irregular R-peak-to-R-peak (RR) intervals and the presence of low amplitude fibrillatory waves (Figure 1-11).

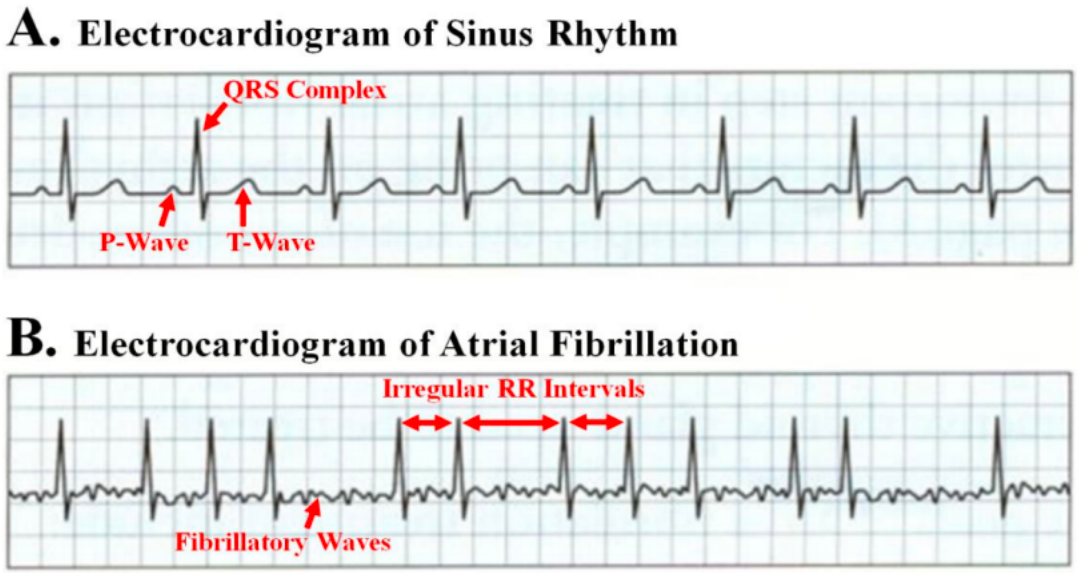


Figure 1-11: Illustration and comparison of typical electrocardiogram (ECG) recordings of **A) sinus rhythm, and B) atrial fibrillation (AF)** [23]

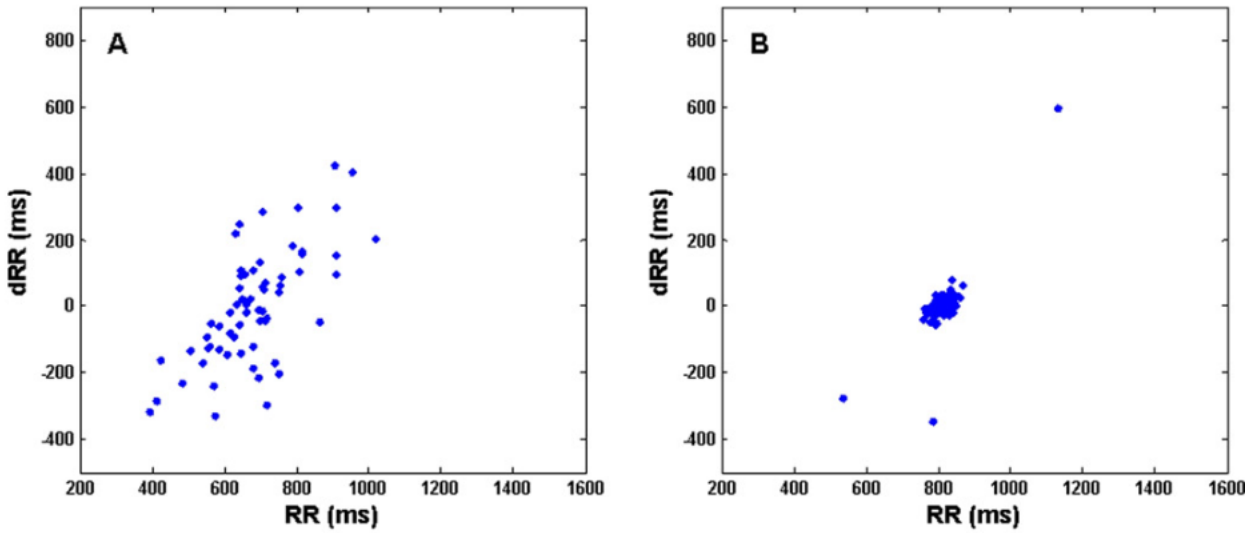


Figure 1-12: (A) Distribution of RR interval in AF and (B) in normal ECG [24]

1.7 Smart Devices for Accurate AFib Detection

Technological innovations are continually growing in all sectors, but they are especially fascinating in medicine. The introduction of new technological devices, such as smart phones and smart watches, has enabled a wide range of activities, including recording daily steps, distance traveled, sleeping patterns, calories burned, and even monitoring heart rate. However, the true potential of these technical improvements resides in their ability to aid in disease diagnosis and abnormality detection, resulting in timely treatment and prevention of potential problems. This is especially important in the case of atrial fibrillation, a very common disorder with potentially serious repercussions [25].

The most significant drawback is that AF is typically diagnosed after extensive monitoring. On the other hand, intelligent devices with accurate monitoring capabilities may overcome this constraint.

This section will introduce many clever gadgets meant to detect atrial fibrillation:

1. **iBeat:**is a smartwatch that uses heart rate monitoring to avoid heart-related disorders. Its purpose was to save lives, more especially to lower the hundreds of thousands of deaths caused by heart disease each year. Because iBeat can identify heart attack and other cardiac disorders' signs, it's a great tool for preventing cardiovascular illness [26].

2. **KardiaMobile:** takes a 30-second, medical-grade EKG anywhere, at any time. The world's top cardiac care physicians and patients use it to detect atrial fibrillation, bradycardia, tachycardia, or normal heart rhythm.

3. **Apple Watch:** detects anomalies in the heart's health and sends out alarms when anything is dangerous. The smartwatch detects abnormally high or low heart rates as well as irregular heart rhythms, and it warns the patient even if they are not experiencing any symptoms. Notably, the Apple Watch features an emergency button that enables those affected to get assistance when necessary. His emergency contacts and emergency services are alerted as soon as he clicks

the button. His companion can see how active he is thanks to the Apple Watch's ability to track exercise, blood pressure, and sleep in addition to heart rate.

Figure 1-13: From left to right : Apple watch, Kardia mobile, iBeat

1.8 Conclusion

Despite good progress in the management of patients with atrial fibrillation (AF), this arrhythmia remains one of the major causes of stroke, heart failure, sudden death, and cardiovascular morbidity in the world. It is also associated with poor quality of life and adverse symptoms. Moreover, the number of patients with AF is expected to rise steeply in the coming years. An early diagnosis of atrial fibrillation requires rhythm documentation using an electrocardiogram (ECG) to reduce the risk of such complications.

In the next chapter, we will delve into the latest techniques and methods used to address this disease, aiming to improve outcomes and enhance patient care.

Chapter 02

State of the art

Chapter 2

State of The Art

2.1 Introduction

Have you ever wondered how google translates an entire web page to a different language in a matter of seconds or your phone gallery group's images based on their location? all of this is a product of deep learning.
This chapter delves into deep learning and the most recent studies in using this form of analysis for assessing atrial fibrillation.

2.2 Theory of Deep Learning

In this section, we will outline the fundamentals behind a deep neural network, which is essential for understanding what deep learning technologies have to offer and the optimizations they use.

2.2.1 History and Origins

The concept of artificial neural networks dates back to 1940, when Walter Pitts and Warren McCulloch found that neurons in the human brain perform logical operations and have binary outputs based on a specified threshold: active or not active [27].

Mathematical models have sparked attention in the AI community. At that time,

computers were not yet advanced enough to handle sophisticated algorithms. Scientists lost interest in Neural Networks over time due to lack of advancement and alternative viable technologies. Backpropagation was discovered by Seppo Linnainmaa in 1970, and it would subsequently change the performance of neural networks. However, Neural Networks were not the focus of the Artificial Intelligence Community until 2012, when students won the ImageNet Large Scale Visual Recognition Competition with outstanding results. Since then, neural networks have been a popular topic in artificial intelligence.

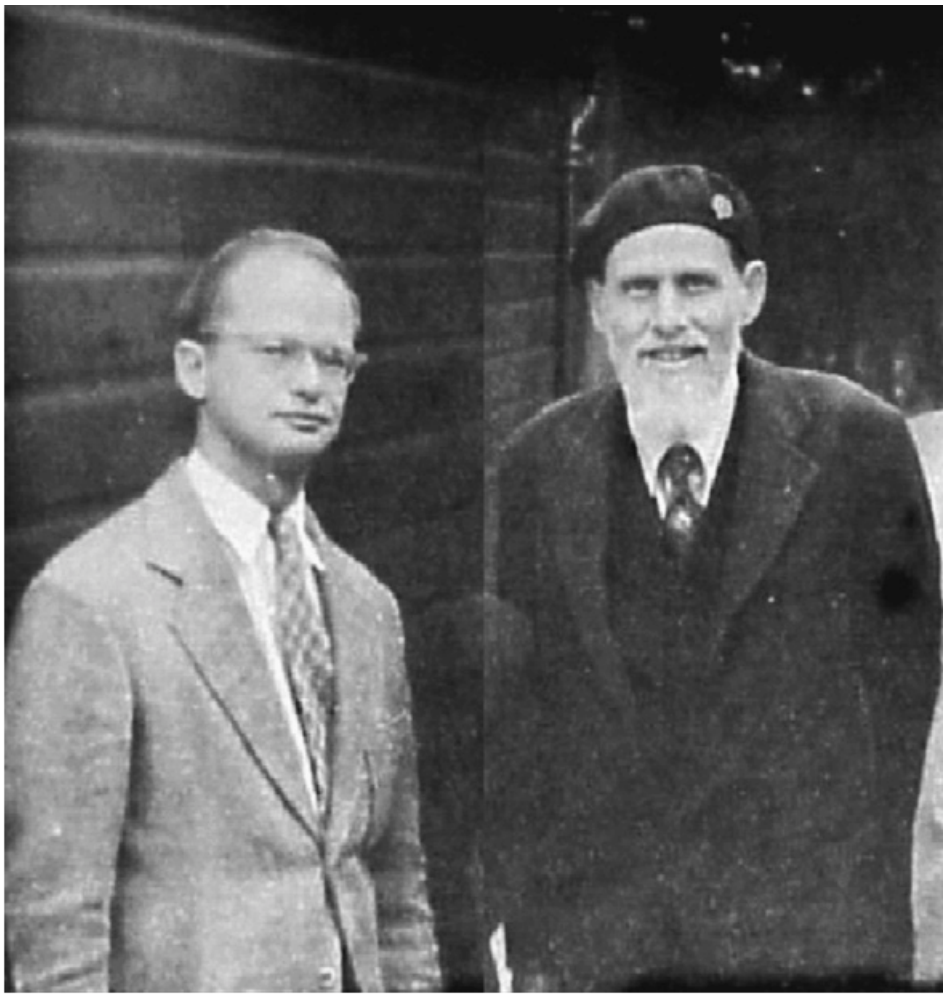


Figure 2-1: McCulloch (right) and Pitts (left) in 1949 [28]

2.2.2 Definition

Deep Learning is a subset of machine learning (ML) which in turn is a subset of artificial intelligence -a technique that enables a machine to mimic human behavior-. Machine learning is a technique to achieve AI through algorithms trained with data and finally deep learning (DL) is a type of machine learning inspired by the structure of the human brain in terms of deep learning this structure is called **an artificial neural network** or ANN [29].

Machine learning and deep learning both are subsets of artificial intelligence but there are many similarities and differences between them. Table 2.1 and Figure 2-2 show main differences between the two.

Table 2.1: Main Differences between machine learning and deep learning

Traditional Machine Learning	**Deep Learning**
Requires more human intervention to correct and learn	Learns on its own from environment and past mistakes
Can train on smaller data sets	Requires large amounts of data
Shorter training and lower accuracy	Longer training and higher accuracy
Can train on a CPU (Central Processing Unit)	Better to use a GPU to train (Graphics Processing Unit)

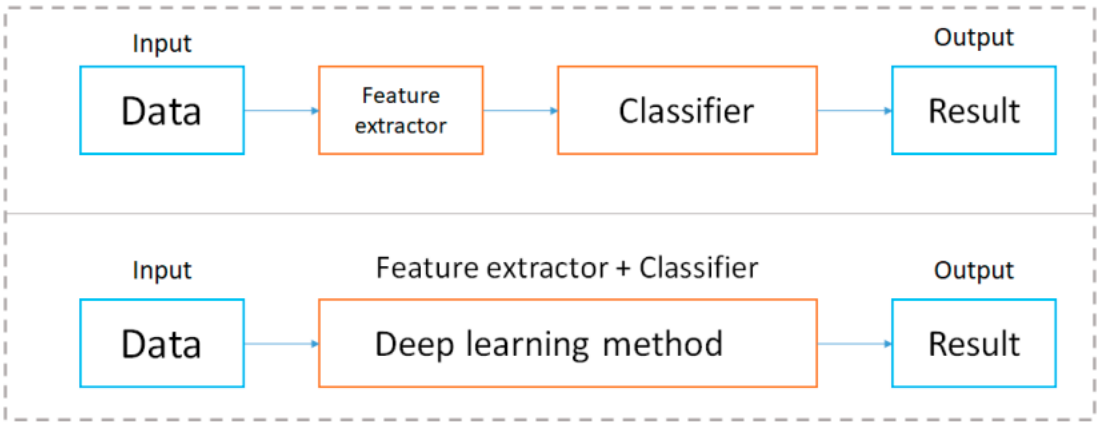


Figure 2-2: The difference between traditional machine learning and deep learning

2.2.3 Strategic Application of Deep Learning

Deep Learning can outperform human specialists in some scenarios, making it a potential solution for the following problems [30]:

- In situations where human specialists are not available.
- Situations in which humans are unable to explain conclusions made with their expertise (language comprehension, medical decisions, and speech recognition).
- Cases where the problem solution changes over time (price prediction, stock choice, weather prediction, and tracking).
- Adaptive solutions for specific scenarios, such as personalization and biometrics.
- Examples of large-scale problems outside our reasoning abilities, such as sentiment analysis, Facebook ad matching, and webpage ranking calculations.

2.2.4 Advantages of Deep Learning

Based on [30], we summarize the advantages in the following items:

1. **Universal Learning Approach:** Because DL can perform in almost all application domains, it is also known as universal learning.
2. **Robustness:** In general, precisely defined features are not necessary in deep learning algorithms. Instead, the optimum features are learned in an automatic manner relevant to the task at hand.
3. **Generalization:** Different data types and applications can employ the same deep learning (DL) technology, known as transfer learning (TL), as discussed in the next section. Furthermore, it is an effective approach for problems with insufficient data.
4. **Scalability:** Deep learning is very scalable. Microsoft invented ResNet, a 1202-layer neural network widely used in supercomputing. Lawrence Livermore National Laboratory (LLNL), a big company working on growing network

frameworks, adopted a similar technique, allowing for the implementation of thousands of nodes.

2.2.5 Challenges in Deep Learning

Although deep learning has made great strides in many areas, there are still certain issues that need to be resolved. The following are a few of the primary obstacles in deep learning:

- **Data availability:** To learn from, a lot of data is needed. Acquiring as much training data as possible is crucial for deep learning applications.
- **Computational Resources:** It is computationally expensive to train the deep learning model because it needs specialized hardware, such as GPUs.
- **Time-consuming:** Working with sequential data might take a very long time, even in days or months, depending on the computational resources available.
- Deep learning models are **intricate and operate in a mysterious manner**. The outcome is really hard to understand.
- **Overfitting:** Repetition of training causes the model to become excessively specialized for the training set, which results in overfitting and subpar performance on fresh data.

2.3 Neural Networks: From Biological to Artificial

Neural networks are computational models that closely resemble the structure and function of organic brain networks. They are made up of interconnected pieces called artificial neurons, which process information and learn patterns from data.

To understand how neural networks work, we must first consider how biological neural networks function. The human brain is made up of billions of cells called neurons that communicate with one another using electrical and chemical impulses.

Figure 2-3 compares a biological neuron and an artificial neuron, illustrating its structural components [31].

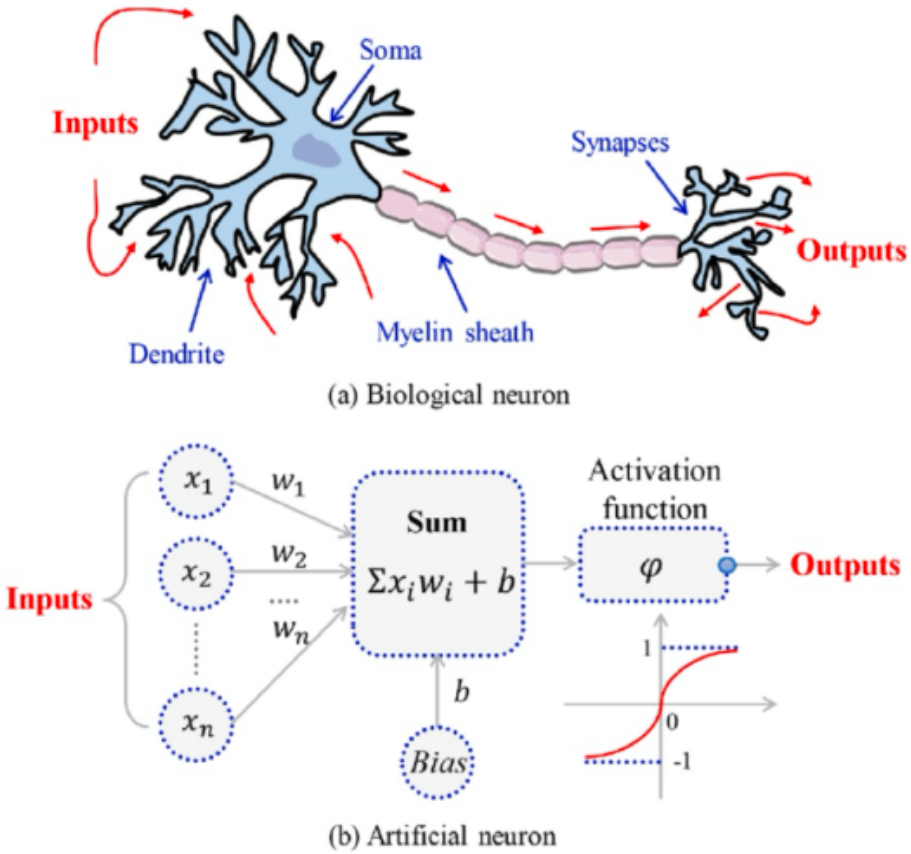


Figure 2-3: a) Biological Neuron b) Artificial Neuron [31]

A biological neuron is made up of three basic parts: dendrites, soma, and axon. Dendrites function as neuronal the antennas, receiving impulses from other neurons. Soma functions similarly to the neuron's processor, integrating and processing signals. The axon functions as a neuron's cable, conveying processed signals to neighboring neurons via synaptic connections. Synapses function similarly to neural connectors, transferring signals from one neuron to another.

In artificial neural networks, this complex process is simplified into weighted sums and activation functions. An artificial neuron has multiple inputs, each assigned a

specific weight that signifies its importance. These inputs are summed up and passed through an activation function to produce an output. This is how artificial neurons transform inputs into outputs [31].

Table 2.2: Comparison of Biological and Artificial Neurons [31]

Aspect	**Biological Neurons**	**Artificial Neurons**
Structure	Complex and organic structure, consisting of dendrites, soma, axon, and synapses.	Simple and mathematical structure, consisting of inputs, weights, bias, and activation function.
Function	Process and transmit electrical and chemical signals, using action potentials and neurotransmitters.	Process and transmit numerical values, using weighted sums and activation functions.
Learning	Learn and adapt through synaptic plasticity, changing the strength and number of synapses based on experience and stimuli.	Learn and adapt through weight adjustment, changing the value and number of weights based on error and feedback.
Efficiency	Highly efficient and parallel, processing and transmitting signals at high speed and low energy consumption.	Less efficient and sequential, requiring more time and power to perform computations and communications.

2.4 Artificial Neural Network

Artificial neurons, or units, are components of artificial neural networks. The Artificial Neural Network of a system is made up of these units grouped in a sequence of layers.

Now after having explored what artificial neurons are, lets look at how they interact to build a neural network. Artificial neurons are frequently grouped in layers to form a neural network. The first layer receives the input data, the last layer generates the output, and the intermediate layers are known as hidden layers. Each layer applies a specific change to the data before forwarding it to the next layer. A neural network's number of layers and neurons determines its ability to learn difficult functions.

The input layer consists of several neurons, each representing one of the features of the data. The hidden layer contains multiple neurons that perform computations on the input data. Finally, the output layer has one or more neurons that generate the final prediction or classification.

Deep Learning and neural networks are often used interchangeably in speech, which can be confusing. As a result, it's worth noting that the "deep" in deep learning simply refers to the number of layers in a neural network. A deep learning algorithm is defined as a neural network with more than three layers, including the inputs and output. A neural network with only two or three layers is considered a simple neural network [32].

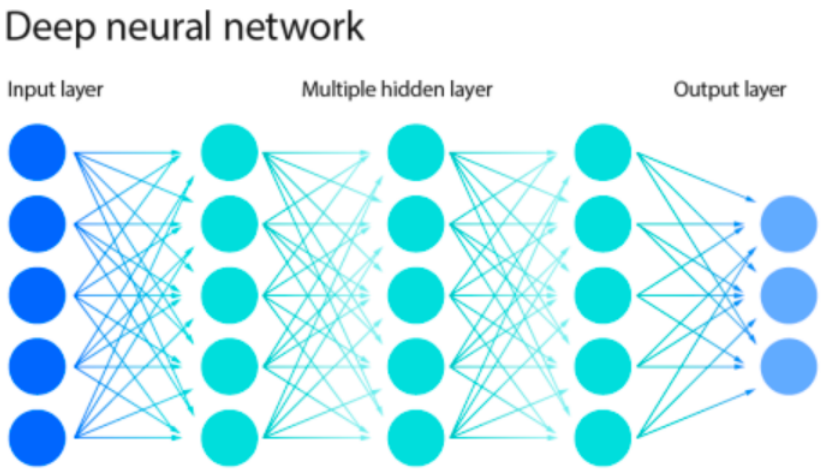


Figure 2-4: Deep Neural Network Layers [32]

2.5 Activation Function

The activation function is an important component of artificial neurons since it determines whether or not a neuron should be activated based on input signals. Activation functions add nonlinearity to neural networks, allowing them to handle complicated issues beyond linear separability. Linear separability indicates that data may be separated by a straight line [31].

However, not every data collection is linearly separable. Some data sets are more complicated and require curved or nonlinear borders to distinguish them. This is when activation functions come in useful. They enable the network to learn nonlinear functions and define nonlinear bounds. Depending on the network architecture and the type of problem, various activation functions serve different objectives. We utilize the following activation functions at this work:

1. **Sigmoid:** This function transfers the input to a value between 0 and 1, resulting in a smooth curve. It is useful for binary classification tasks, such as determining if an email is spam or not. However, it has several disadvantages, such as being susceptible to saturation and vanishing gradients, which means that the network stops learning when the input is too large or too little.

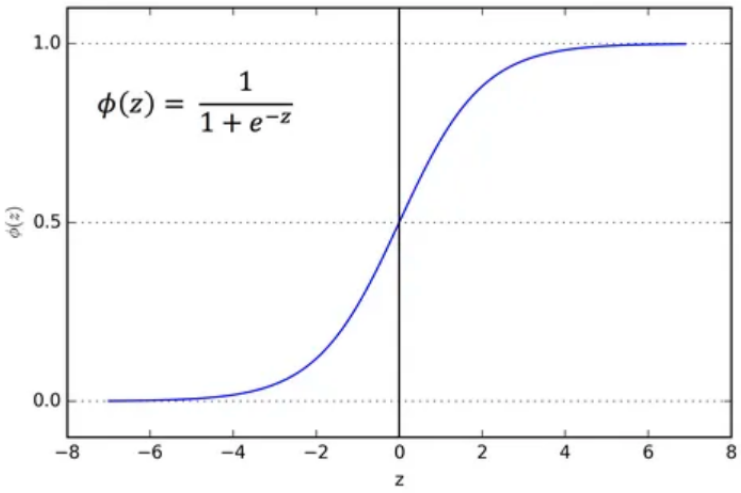


Figure 2-5: Segmoid Activation Function Graph [33]

2. **ReLU:** This function translates the input to either 0 or the input itself, resulting in a linear and nonlinear region. It is useful for accelerating convergence while avoiding the vanishing gradient problem. However, it has some disadvantages, such as being prone to dying neurons, which means that some neurons stop reacting to input and become inactive.

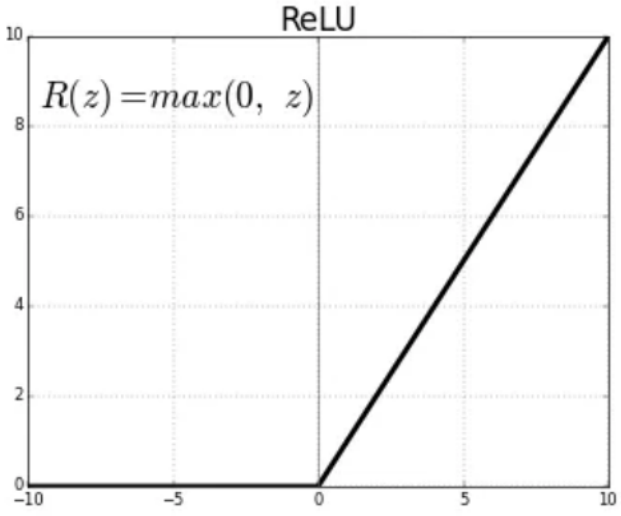


Figure 2-6: Rectified Linear Unit Activation Function Graph [33]

2.6 Training a Neural Network

Neural networks function by altering the weights of the connections between neurons based on the error of network predictions compared to the actual data.

The training procedure involves the network learning from the data and improving its performance. Various techniques can be used for training, including gradient descent, backpropagation, stochastic gradient descent, and so on.

The training procedure aims to minimize the error or loss function, which assesses how well the network fits the data. The network works better when there are fewer errors or losses.

The training process can be repeated until the network reaches a satisfactory level of accuracy or meets some predefined criteria.

2.7 Deep learning for the detection of atrial fibrillation

In this section, we discuss in detail the approaches that closely related to our work, which are widely used to detect AF.

2.7.1 1D-CNN Architecture

Typically, a 1-D CNN model involves two kinds of layers (as shown in Figure 2-7) Convolution layers and Multilayer Feed Forward (MLFF) layers.

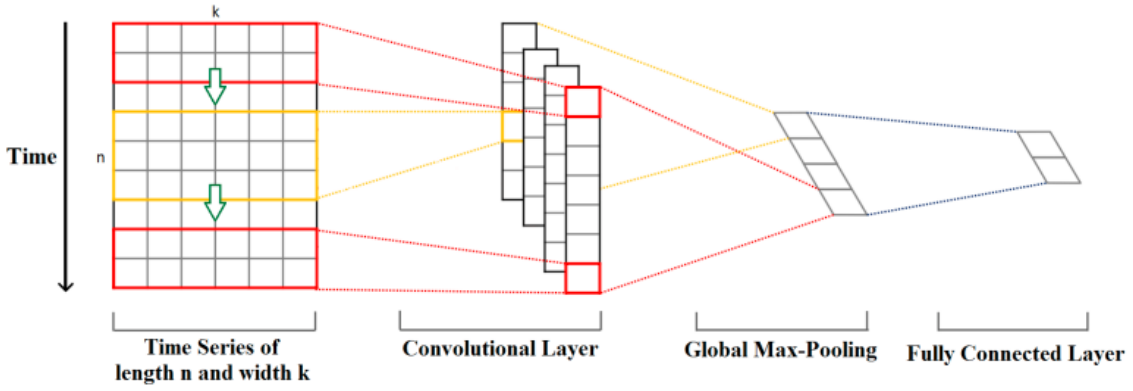


Figure 2-7: 1D-CNN Architecture

1. **Convolution layers:** Convolution, sometimes known as shiftcompute, is the process of sliding a kernel over input signals.

2. **Pooling layer:** Pooling is a useful technique for reducing a mapping's dimensionality and emphasizing its key features. In order to decrease the convolution output's dimension, pooling is typically used after the convolution layer.

3. **Flatten:** Convolution layers may produce output with a depth greater than one. The output of the convolution layers is concatenated to create a flat structure called "flatten," which can be supplied as input to a multilayer feed forward network as shown in Figure 2-8.

4. **Multilayer Feed Forward Network:** A fully connected layer (Figure 2-8), or MLFF, is a network structure in which every neuron in one layer is connected to every other neuron in the layer above it. Convolution layers' flattened output is received by MLFF, which transfers it to the output

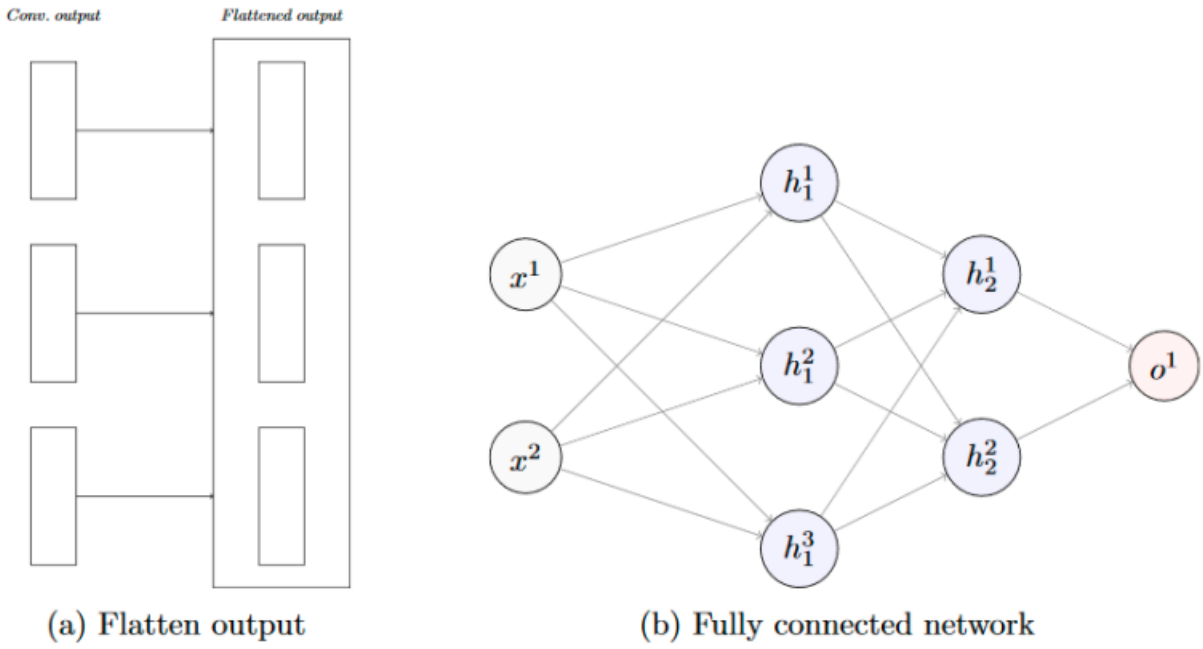
5. **Output Layer:** One-hot encoding format is most commonly used to represent a classification outcome. The result of a N class problem is a vector with dimension N. This output vector's elements can only have values of 0 (ON) or 1 (OFF). Additionally, only one vector element may be "ON" at once. Take the case of four classes, for instance. The four classes are represented by:

Class 1: [1 0 0 0]

Class 2: [0 1 0 0]

Class 3: [0 0 1 0]

Class 4: [0 0 0 1]



(a) Flatten output (b) Fully connected network

Figure 2-8: a) Flatten Output, b) Fully Connected Network

2.7.2 1D CNN vs 2D CNN for Atrial Fibrillation Detection

The automatic detection of atrial fibrillation (AF) is of paramount importance due to its association with an elevated risk of embolic stroke. Traditional approaches for AF detection have involved the transformation of one-dimensional (1D) time-series electrocardiogram (ECG) signals into two-dimensional (2D) spectrograms, which were then utilized to train complex AF detection systems. This process, however, is computationally intensive and often incurs significant implementation costs.

Convolutional Neural Networks (CNNs) have demonstrated exceptional efficacy in analyzing 2D image data, excelling in tasks such as object recognition, classification, and prediction by extracting high-level features through multiple hidden convolutional layers. Despite their success with 2D data, the intrinsic nature of physiological signals, including ECGs, is inherently 1D. Consequently, there has been a growing interest in adapting CNNs for the direct processing of 1D signals.

One approach to leverage CNNs for signal processing involves the conversion of 1D signals into 2D representations, such as spectrograms, which then facilitate the application of conventional 2D CNN architectures. This method allows the network to operate on signal data as if it were handling image data, benefiting from the mature techniques developed for 2D image analysis. However, the transformation of 1D signals into spectrograms introduces additional computational complexity and may lead to the loss of crucial temporal information, which is essential for the accurate characterization of time-series data.

Using 1D time-series data directly in a neural network, particularly a 1D CNN, is often preferable to converting it into spectrograms. This preference stems from the 1D CNN's ability to maintain the original temporal resolution and characteristics of the signal, thereby preserving its intrinsic temporal patterns and correlations. The direct use of 1D CNNs in processing time-series data ensures that the sequential na-

ture of the data is retained, which is critical for the accurate analysis of physiological signals.

Furthermore, the transformation of 1D signals into 2D spectrograms not only increases computational overhead but also risks omitting significant temporal features due to the added complexity of the conversion process. In contrast, 1D CNNs, with their lower computational demands and specialized efficiency in handling time-series data, present a practical and effective solution. They capitalize on the sequential nature of the data, facilitating a more streamlined and accurate analysis process.

In the context of AF detection, the utilization of 1D CNNs offers a compelling pathway for the rapid and reliable examination of ECG signals. Their ability to directly process 1D time-series data without the need for transformation into 2D spectrograms provides an advantageous framework for the development of efficient, cost-effective, and robust AF detection systems.

2.8 Literature Review

Electrocardiography (ECG) manual screening for AF is costly (time consuming) and error-prone. The application of advanced signal processing and deep learning approaches in the development of computer-aided diagnosis systems to enable automated identification of AF can help reducing subjectivity and human errors as well as improve the accuracy and timeliness of diagnosis.
This section highlights the most recent advancements in deep learning for the investigation of atrial fibrillation and discusses the drawbacks of each method.

2.8.1 Related Works

Existing atrial fibrillation classification works can be classified into two paradigms: intra-patient and inter-patient.
In intra-subject scheme, the dataset is based on segments of ECGs of subjects in

both train and test/validation sets, so the ECG record will appear in two subsets. According to [34], Because the patient's traits are learned during the training phase, the findings of this paradigm are biased yielding about 100% accuracy in the test phase [35]. In real-world circumstances, the trained model should be capable of handling inter-patient changes during the training process. Although, intra-subject evaluation process leads to a relatively high performance in most of the cases, these results are not realistic.

Using the inter-subject technique avoids using segments from the same subjects in both training and validation/test sets, making it more practical.

The proposed AF detection models have limited generalization and performance on new datasets. Current approaches have primarily been tested on a single dataset. Previous research has primarily used intra-subject approaches for training and evaluation, rather than inter-subject paradigms.

According to [36] all deep learning methods used to automatically detect atrial fibrillation are illustrated in the following block diagram:

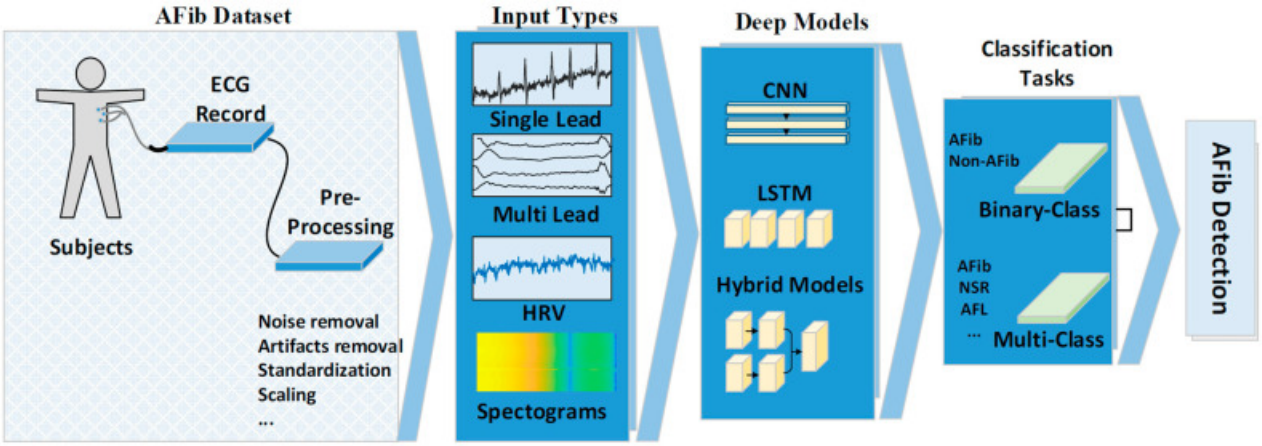


Figure 2-9: Block diagram representation of the general approach for deep learning-based atrial fibrillation detection [36]

For instance, [36] examined the newest state of the art related to AFib detection. Figure 2-10 depicts the distribution of publications on atrial fibrillation detection using deep learning by year of publication.

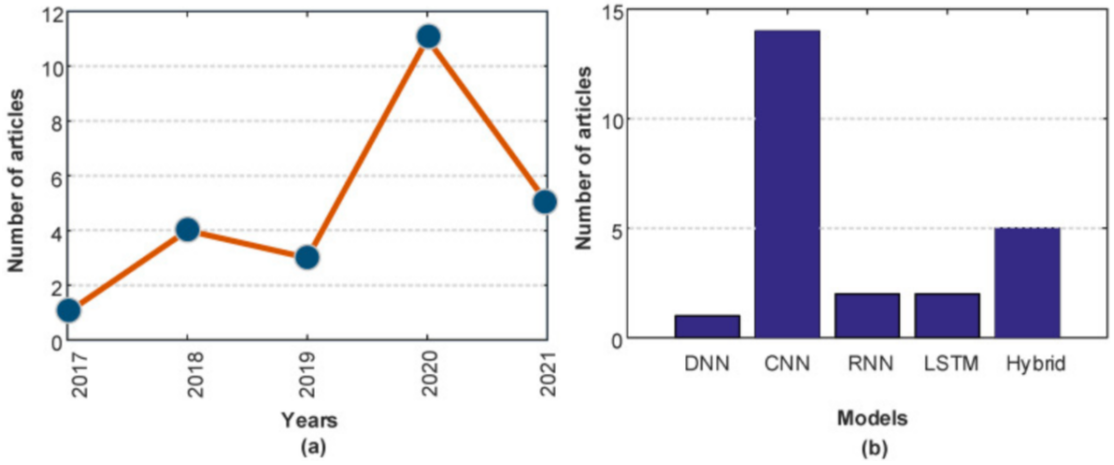


Figure 2-10: (a)type of model deployed (b). CNN, convolutional neural network; DNN, deep neural network; LSTM, long short-term memory; RNN, recurrent neural network [36]

After analyzing the various methods utilized in the articles, we can conclude the advantages and cons of employing various deep learning algorithms for Atrial Fibrillation detection (Table 2.3).

Table 2.3: Deep learning models developed for automatic AF detection

Deep Models	Advantage/Disadvantage
DNN	In terms of speed, it is more advantageous.
CNN	Strong in obtaining representative properties, but lacking in design difficulties and parameter tuning
RNN	Although it is used because of its memory structure, it is poor at representing sequences.
LSTM	Although useful for sequence representations, it is slow and consumes a lot of resources.
Hybrid (CNN+LSTM)	The use of both representation and sequence features together is advantageous, but it takes more time and cost.

Most articles use a 1D CNN for AF detection because of its ability to successfully

extract features from raw ECG data and provide solid results in classification tests. Unlike standard neural networks, one-dimensional CNNs excel at capturing local relationships in time series data, making them ideal for ECG signal interpretation. Furthermore, 1D CNNs strike an appropriate compromise between computational efficiency and accuracy, avoiding the complexities and resource demands associated with RNNs and LSTMs. This technique enables precise and efficient AF detection, which is critical for quick medical action.

2.9 Proposed Method

In the preceding sections, we reviewed contemporary approaches for detecting atrial fibrillation (AF). It was observed that employing a one-dimensional Convolutional Neural Network (1D CNN) to extract features directly from raw electrocardiogram (ECG) data proves to be highly efficient. Moreover, the lack of continuous ECG monitoring devices poses a significant challenge for cardiologists in identifying AF episodes, particularly during periods when patients are sleeping or under similar conditions.

Therefore, in this master's project, we propose a novel methodology to address these challenges. The key components of our approach are as follows:

- QRS Detection Algorithm for R Peak Identification: We propose the utilization of a robust QRS detection algorithm to precisely identify the R peaks in ECG signals. The identified R peaks will then be used to extract individual heartbeats. These beats will be reshaped into image representations, allowing the transformation of our 1D model into a two-dimensional (2D) format. This step enhances the model's applicability to new records and facilitates the integration of 2D CNN techniques.

- Development of 1D-CNN and 2D-CNN Models: We propose the development of innovative 1D-CNN and 2D-CNN models specifically tailored for AF detection. The design will focus on optimizing cost efficiency by utilizing only two convolu-

tional layers. This streamlined architecture aims to maintain high performance while reducing computational complexity and resource requirements.

- Hyperparameter Tuning for Optimal Performance: We will conduct extensive hyperparameter tuning to identify the optimal configuration of parameters. This process ensures that the model achieves the best possible performance by adjusting parameters such as learning rate, batch size, and the number of filters in each convolutional layer.

- Fixed Seed for Consistent Results: To ensure consistent results across different runs and facilitate reproducibility, we will establish a fixed seed value for initializing weights uniformly. This practice is critical for achieving reliable and comparable outcomes in the model training and evaluation processes.

- Signal-Specific Data Augmentation: We will implement data augmentation techniques specifically suited for ECG signals. These techniques will expand the training dataset, thereby enhancing the model's robustness and improving its ability to handle unseen data effectively. Examples of augmentation techniques include adding noise, varying signal amplitude, and time-warping.

- Inter-Subject Approach for Generalization: An inter-subject approach will be employed to improve the model's generalization capability. This approach involves training and validating the model on data from different subjects, ensuring that the model can generalize well to unseen data and is not overfitted to a specific subset of the population.

The proposed methodology aims to leverage the strengths of both 1D and 2D CNNs while addressing the limitations of existing AF detection techniques. By focusing on cost efficiency, computational simplicity, and robust performance, this approach offers a comprehensive solution for the automatic detection of atrial fibrillation, facilitating its timely and accurate diagnosis in clinical settings.

2.10 Conclusion

This chapter reviewed recent research on detecting atrial fibrillation disease using ECG data. 1D-CNNs are commonly used. In the following chapter, we will offer our proposed system design to help handle these challenges.

Chapter 03

Methodology and Results

Discussion

Chapter 3

Methodology and Results Discussion

3.1 Introduction

In the previous chapter, we reviewed various methods for detecting atrial fibrillation (AF) and concluded that utilizing a one-dimensional Convolutional Neural Network (1D CNN) to extract features from raw ECG data is particularly efficient. This chapter introduces the proposed methodology for AF detection, offering a detailed exposition of each phase, from signal filtering and peak detection to feature extraction and the overall system architecture.

We present a comprehensive, step-by-step account of the advanced system architecture that enhances the capabilities of CNNs, ensuring precise AF detection from a single heartbeat. This systematic approach involves several stages, each meticulously designed to improve the accuracy and reliability of the detection process. The chapter concludes with an in-depth analysis of the experimental results, providing valuable insights into the system's performance and demonstrating its efficacy in identifying atrial fibrillation.

3.2 Development Tools and Programming Languages

This section discusses the key tools utilized in the project, highlighting notable Python libraries, such as NumPy, Pandas, Matplotlib, Scikit-learn, Keras, and TensorFlow. These libraries offer extensive capabilities for effective data manipulation, analysis, visualization, and deep-learning tasks.

3.2.1 Programming Tools

3.2.1.1 Python

Python is an interpreted high-level general-purpose programming language. Created by Guido van Rossum. Its language constructs and object-oriented approach aim to help programmers write clear logical codes for small and large-scale projects. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). This language can be used on a server to create web applications and connect them to database systems. It can also read and modify files, to handle big data and perform complex mathematics and so many other functionalities [37].

3.2.1.2 VScode

Visual Studio Code (VS Code) is a free and open-source code editor developed by Microsoft that supports a variety of programming languages and development tools.
It offers features such as debugging, task running, and version control and is integrated into a seamless interface. The lightweight nature of VS Code combined with its powerful extension marketplace makes it a popular choice among developers owing to its versatility and customizability [38].

3.2.1.3 Anaconda

Anaconda is a free and open-source distribution of the Python programming language for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. [39].

3.2.2 Useful Frameworks and Libraries

3.2.2.1 Tensorflow:

TensorFlow is an open-source machine learning framework developed by Google. It offers a wide range of tools and libraries for building and deploying machine learning models. With support for tasks like deep learning, TensorFlow provides flexible and scalable APIs, making it a versatile framework for machine learning development.

3.2.2.2 Keras:

Keras is a high-level neural networks API written in Python. It provides a user-friendly interface to build and train deep learning models.
Keras is built on top of other deep learning frameworks such as TensorFlow and allows for fast prototyping and easy experimentation.

3.2.2.3 Pandas

Pandas is a powerful Python library for data manipulation and analysis, offering efficient handling of structured data through DataFrames. It is widely used by data scientists and

analysts.

3.2.2.4 NumPy

NumPy is a vital Python library for numerical computing, offering powerful support for large arrays and mathematical operations. It is extensively used in scientific and data analysis libraries, thanks to its efficiency and seamless integration with other Python tools like Pandas and Matplotlib.

3.2.2.5 Matplotlib

Matplotlib is a versatile data visualization library in Python, offering extensive functions and tools for creating high-quality plots, charts, and graphs. It integrates well with other libraries like NumPy and Pandas, making it a popular choice for exploratory data analysis and result presentation.

3.2.2.6 Scikit-learn

Scikit-learn is a Python library for machine learning, offering a userfriendly interface and various algorithms for tasks like classification, regression, and clustering. It is widely used and known for its simplicity, performance, and strong community support.

3.3 Evaluation Metrics for Classification Tasks

The evaluation metric is crucial for attaining the best classifier during classification training. Choosing an appropriate assessment metric is crucial for achieving optimal classifier efficiency [40]. There are various metrics for measuring the performance of classifiers based on the following four indicators:

- **True Positive (TP):** is a test result that correctly identifies the presence of a class or characteristic.
- **True Negative (TN):** A test result that accurately indicates the lack of a class or characteristic.
- **False Positive (FP):** is a test result that incorrectly suggests the presence of a class or characteristic.
- **False Negative (FN):** is a test result that incorrectly implies the absence of a class or characteristic.

3.3.1 Confusion Matrix

Refer to the graphical representation in Figure 3-1. A confusion matrix is a compact table that assesses the performance of a classification model by displaying true-positives, false-negatives, and true-negatives. It calculates key measures, such as precision, recall, accuracy, and F1 score. Confusion matrices can be created using R and Python[41].

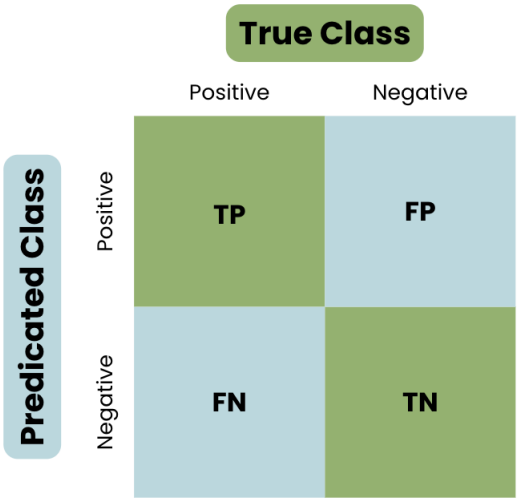


Figure 3-1: Confusion matrix for multi-class classification

The scikit-learn library in Python was used to calculate a confusion matrix.

```
from sklearn.metrics import confusion_matrix

# Example true labels and predicted labels
true_labels = [0, 1, 1, 0, 1, 0]
predicted_labels = [0, 1, 0, 0, 1, 1]

# Calculate confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)

print("Confusion Matrix:")
print(cm)
```

Listing 3.1: Example of Confusion matrix usage

The code above prints the following result: (In this matrix, the rows represent the actual classes, and the columns represent the predicted classes. So, for example, the value at row 1, column 1 (2) indicates the number of instances where the true label was 0 and the predicted label was also 0)

```
Confusion Matrix:
[[2 1]
 [1 2]]
```

3.3.2 Accuracy

Accuracy measures the frequency with which the model is correct. However, relying solely on the accuracy can sometimes lead to misleading high-performance results. The accuracy is given by Equation 3.1.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{Total Predictions} \tag{3.1}$$

Here, the Total Predictions represent the sum of the true positives, true negatives, false positives, and false negatives.

This is a commented section. None of these lines will be rendered. You can comment out multiple lines using this environment.

Because it is a simple mathematical formula, we can use the following Python function:

```
def calculate_accuracy(true_positive, true_negative, false_positive, false_negative):
    total_predictions = true_positive + true_negative + false_positive + false_negative
    accuracy = (true_positive + true_negative) / total_predictions
    return accuracy
```

Listing 3.2: Function for Accuracy calculation

3.3.3 Precision

Precision represents the accuracy of a model in classifying a sample as positive. This was calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.2}$$

3.3.4 sensitivity

Recall, often called sensitivity, is a metric that assesses a model's ability to correctly identify positive samples. The calculation involves dividing the number of true positives (TP) by the sum of TP and FN. The equation for recall (sensitivity) is as follows:

$$\text{Precision} = \frac{TP}{TP + FN} \tag{3.3}$$

3.3.5 F1 Score

The F1-score evaluates a classifier's precision and recall in a single metric using the harmonic mean. It is often used to compare the performance of two classifiers.

The F1-score was calculated using the following formula:

$$\text{F1\_score} = \frac{2 \cdot (\text{precision} \cdot \text{recall})}{\text{precision} + \text{recall}} \tag{3.4}$$

In Equation 3.4, the precision and recall values are multiplied by two, and their sum is included in the denominator. This calculation generates a single F1-score to evaluate the classifier's performance [42].

3.3.6 The loss function

To put it simply, the loss function is a way to gauge how effectively the algorithm models the data that you have provided.

- **Binary Cross-entropy / Log Loss:** Binary cross-entropy, also known as log loss or logistic loss, is a loss function commonly used in binary classification tasks. It measures the dissimilarity between predicted probabilities and actual binary labels (0 or 1). The binary cross-entropy loss function is particularly well-suited for problems where the output of the model is a probability value indicating the likelihood of an instance belonging to one of the two classes.
  For a single instance, let y be the true label (0 or 1) and p be the predicted probability that the instance belongs to class 1. The binary cross-entropy loss L is computed as follows:

$$L(y, p) = -\left(y \cdot \log(p) + (1 - y) \cdot \log(1 - p)\right) \tag{3.5}$$

  Where:

  - y is the true label (0 or 1), p is the predicted probability that the instance belongs to class 1, log denotes the natural logarithm.

  By minimizing the binary cross-entropy loss function during training, the model learns to generate predicted probabilities that are closer to the true labels, thereby improving its ability to discriminate between the two classes.

3.4 Global conception

This section delineates our proposed automatic classification methodology for the detection of AF, leveraging short-duration electrocardiogram (ECG) signals.

The proposed methodology for detecting atrial fibrillation (AF) leverages advanced signal processing techniques and Deep Learning models to accurately classify heart rhythms from raw electrocardiogram (ECG) signals. The entire process is divided into several critical stages, each playing a vital role in ensuring the accuracy and reliability of AF detection. The subsequent sections provide a comprehensive overview of each step, from initial data preprocessing to final model evaluation.

- Raw ECG Signal Acquisition: ECG signals are collected from patients, capturing the electrical activity of the heart over time. These raw signals often contain noise and artifacts that must be addressed to improve the accuracy of subsequent analyses.

- Peak Detection Using XQRS Algorithm: The XQRS algorithm is employed to identify the R-peaks in the ECG signals. Accurate detection of these peaks is essential for segmenting the ECG signals into individual heartbeats.

- Segmentation Around Peaks: Once the R-peaks are identified, the ECG signals are segmented into 80-sample windows centered around each peak. Each segment represents a single heartbeat and serves as a fundamental unit for feature extraction and classification.

- Beat Normalization: The segmented heartbeats undergo normalization to standardize their amplitude and duration.

- Model Training and Data Splitting: With the preprocessed and segmented ECG signals ready, the next step involves training machine learning models to classify heartbeats as either normal or indicative of AF.

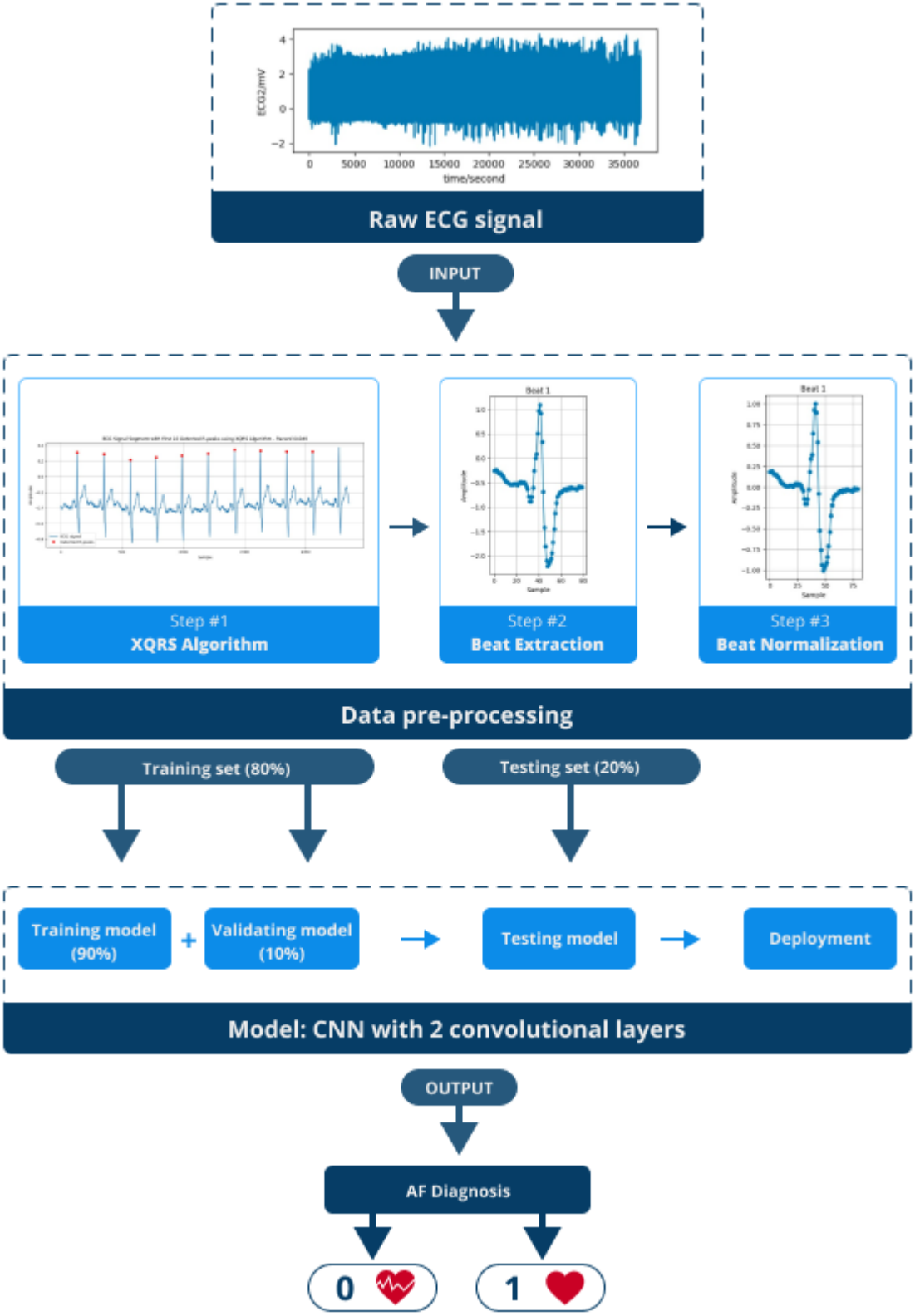


Figure 3-2: Global Conception Architecture

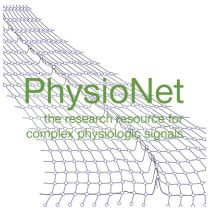
Overall, we describe the architecture of our early detection and monitoring system as Figure 3-2 shows.

3.5 Data Analysis

This section focuses on improving data quality before moving to deep learning part. We will read annotation files, use the XQRS algorithm to detect R-peaks, extract beats, and apply normalization to ensure consistency and accuracy before moving on to the deep learning part.

3.5.1 ECG Data Acquisition

PhysioNet web site:



The PhysioNet website offers a free web access to many recorded physiological signals in databases, most of them being dedicated to the study and the analysis of the ECG signal. In our project, we used the **MIT-BIH AFDB** and the **2017 PhysioNet/CinC Challenge atrial fibrillation database**.

1. **MIT-BIH Atrial Fibrillation Database (AFDB):** This database contains 25 long-term electrocardiogram (ECG) recordings from people who mostly experience paroxysmal atrial fibrillation. Each recording lasted 10 h and included two ECG signals recorded at 250 samples/s with a 12-bit resolution (range from ±10 mV). The data, which came from Boston's Beth Israel Deaconess Medical Center, were manually prepared with rhythm annotations identifying various rhythms such as atrial fibrillation and flutter. Although most records include carefully corrected beat annotations, some are automatically recognized [43]. Each ECG record includes the following extensions:

- **.dat files:** 23 records include two ECG signals, lead I and II (raw ECG signals).

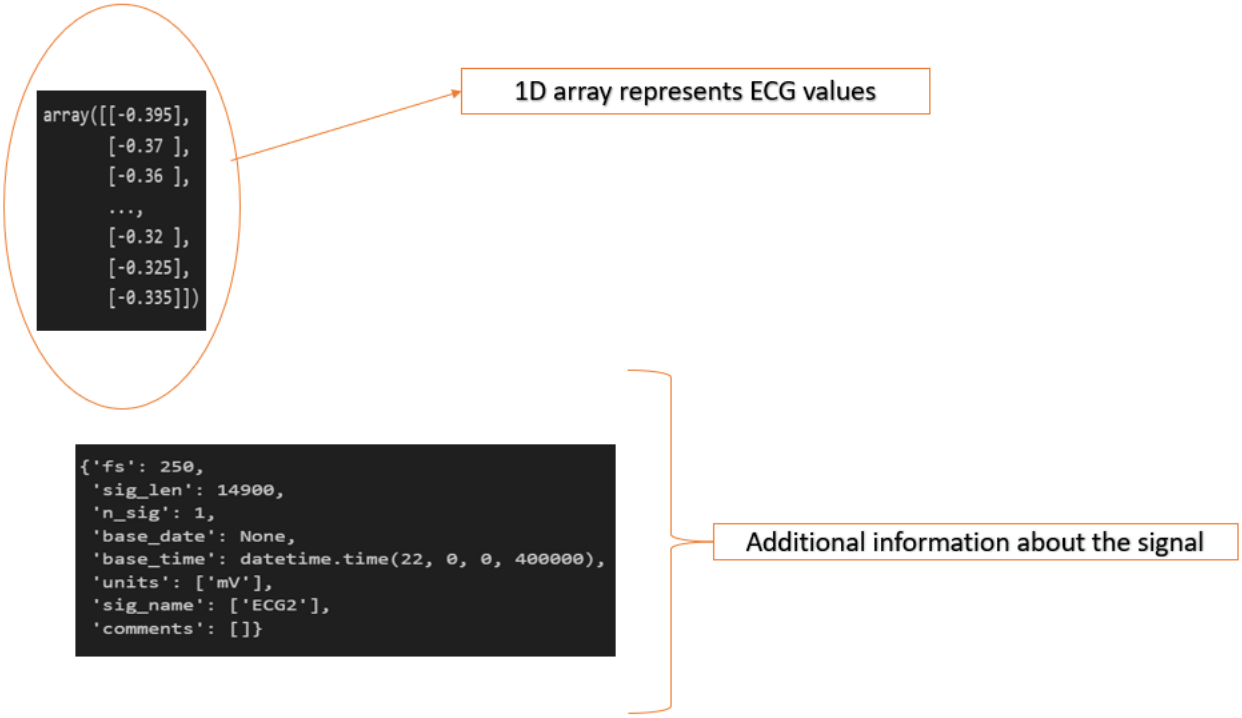


Figure 3-3: Reading (.dat) file

- **.atr files:** annotations marking rhythms and any detected arrhythmias (AFib [atrial fibrillation] , N [normal] , AFL [atrial flatter] , J [junction]).
- **.qrs files:** annotation files (R peak locations) that may be useful for studies of methods for automated AF detection, where such methods must be robust with respect to typical QRS detection errors.

Figure 3-4: Reading (.qrs) file

2. **The 2017 PhysioNet/CinC Challenge:** aims to encourage the development of algorithms to classify, from a single short ECG lead recording (between 30 and 60 s in length), whether the recording shows normal sinus rhythm, atrial fibrillation (AF), an alternative rhythm, or is too noisy to be classified.

Type	# recording	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

Figure 3-5: Data profile for CinC Challenge (used data)

This dataset is used to test the performance of the proposed model in later sections. It contains the following extensions:

- **.mat files:** it is a binary file containing digitized signal samples. Luckily Python offers multiple tools to treat such files.

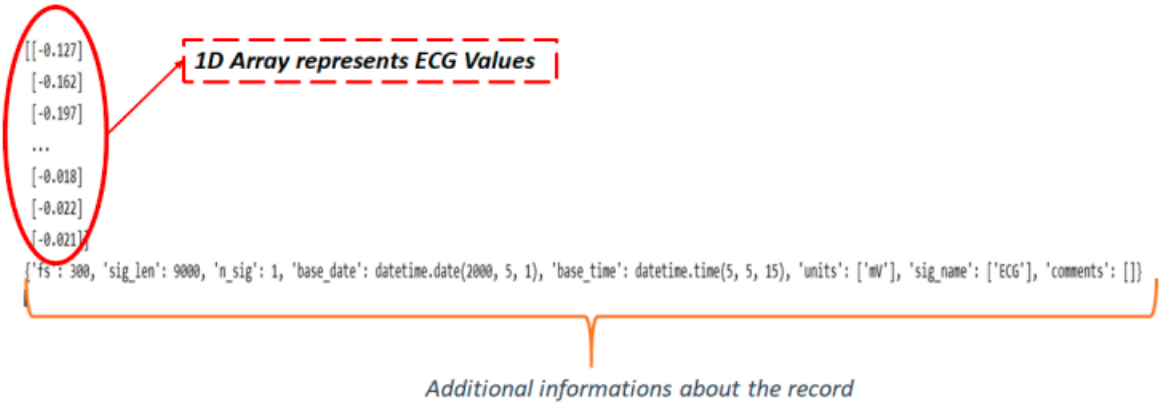


Figure 3-6: Reading (.mat) file

- **.hea files:**it is a Short text file describing the content of associated signal file (Name of record, number of leads, sampling rate, length, time of acquisition, day of acquisition, start value etc.)

- **.csv files:** it contains Class of each ECG record.

997	A00997	O
998	A00998	N
999	A00999	N
1000	A01000	N
1001	A01001	N
1002	A01002	N
1003	A01003	N
1004	A01004	N
1005	A01005	A
1006	A01006	~
1007	A01007	N

Figure 3-7: Example of .csv file

Where :

Table 3.1: ECG records classes

Class	**N**	**A**	**O**	∼
Record's type	Normal	atrial fibrillation	Other heart problems	Noisy

3.5.2 Data pre-processing

Our initial exploration of the database signals involved the utilization of **The WFDB Python package**, a native Python tool designed for the comprehensive handling of physiological signal and annotation data, encompassing reading, writing, processing, and visualization functionalities.

Additionally, the **BioSPPy package** - Biosignal Processing in Python, was employed to facilitate the pre-processing stage, aiding in the preparation and refinement of the data.

Following the setup of the environment, which included the installation of WFDB and BioSPPy, the process commenced as described Listing 3.3

```
%pip install wfdb
%pip install biosppy
import wfdb
from biosppy.signals import ecg # For segmentation
```

Listing 3.3: Installing necessary libraries and importing each

Electrodes positioned on the right arm and left leg to record the voltage difference provides important information about the electrical activity of the heart, as described in Chapter 1, Equation 1.2 of the Lead II system. Atrial fibrillation and other cardiac anomalies can be identified using these data [44].

We provide an example of a record from our database. Figure 3-6 shows Record '04015' from MIT AFDB, as we can see it consists of lead I and lead II records.

The time in seconds ranges from 0 to 35000, therefore dividing by 3600s yields 9.72 hours (more than 9 hours).

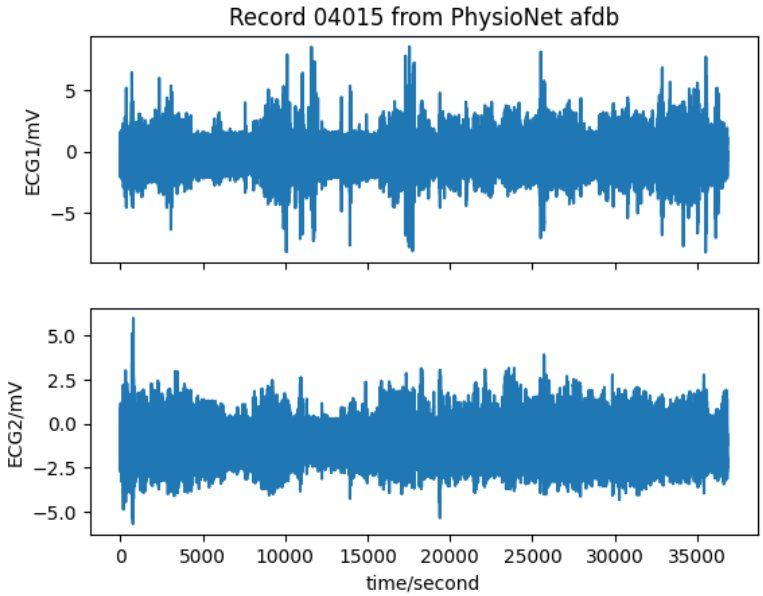


Figure 3-8: The use of WFDB to display raw ECG record '04015'

As depicted in Figure 3-8, two signals are presented: ECG1 and ECG2. As emphasized in [44], our focus lies solely on ECG2, acquired through lead II, as it offers significant insights into atrial fibrillation (AFib).

At this stage, we are equipped to visualize the records, inspect the contents of annotated files, and access pertinent information associated with each record, leveraging the comprehensive resources provided by the WFDB documentation [45].

With a comprehensive understanding of our dataset established, the next step entails denoising and the detection of QRS complexes.

Another library is imported for that purpose which is **neurokit2**

```
%pip install neurokit2
import neurokit2 as nk
```

Listing 3.4: Neurokit2 installation

since it provides various QRS algorithms such as: (pantompkins1985, hamilton2002, zong2003, martinez2004, christov2004, gamboa2008, elgendi2010, engzeemod2012, kalidas2017, nabian2018, rodrigues2021, koka2022, promac).

We have determined that **XQRS** is the best method for qrs complex detection after analyzing all of the algorithms on our database records (Table 3.2).

Figure 3-9 shows some templates that illustrates how XQRS algorithm is accurate:

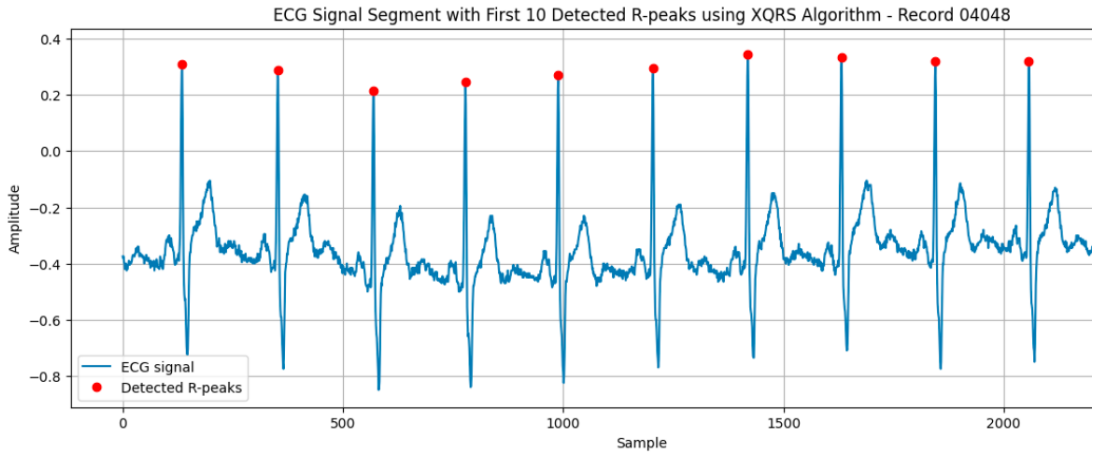


Figure 3-9: First 10 detected R-peaks in record '04048' using XQRS algorithm

Table 3.2: Performance of the XQRS algorithm across the MIT-BIH AFDB

Signals	reference annotations	test annotations	True Positives	False Positives	False Negatives	Sensitivity	Positive Predictivity
04936	53646	55053	53542	1511	104	0.9981	0.9726
07162	39298	39501	8396	31105	30902	0.2136	0.2126
08219	59293	60738	58761	1977	532	0.9910	0.9675
08215	43356	44171	43173	998	183	0.9958	0.9774
08434	39850	40763	39797	966	53	0.9987	0.9763
05121	49881	48288	47010	1278	2871	0.9424	0.9735
04015	44005	44492	42968	1524	1037	0.9764	0.9657
06426	55155	56532	54690	1842	465	0.9916	0.9674
06453	34837	35004	34760	244	77	0.9978	0.9930
04746	47873	48738	47484	1254	389	0.9919	0.9743
04043	61915	63243	61790	1453	125	0.9980	0.9770
07879	56594	58062	56421	1641	173	0.9969	0.9717
04908	61760	63144	61719	1425	41	0.9993	0.9774
05091	36793	36692	35540	1152	1253	0.9659	0.9686
08455	59552	60128	59231	897	321	0.9946	0.9851
06995	55189	56634	55151	1483	38	0.9993	0.9738
07859	60266	61723	59837	1886	429	0.9929	0.9694
04126	42860	43862	42497	1365	363	0.9915	0.9689
04048	39934	40020	39829	191	105	0.9974	0.9952
08405	58856	60403	58657	1746	199	0.9966	0.9711
07910	36599	37789	36588	1201	11	0.9997	0.9682
05261	45534	46499	45343	1156	191	0.9958	0.9751
08378	45515	46310	45487	823	28	0.9994	0.9822

As outlined in Table 3.2, the decision has been made to exclude record '07162' from our analysis to mitigate potential challenges during the training phase of our deep learning model.

We should note that While the primary focus of XQRS is QRS detection, it involves some signal processing steps that can contribute to de-noising and filtering the ECG records.

3.5.3 Beat Extraction (segmentation)

After the successful detection of R-peaks, a crucial step is the extraction of beats. In cases of atrial fibrillation, where beats are closely spaced, a window of 80 samples centered around the R-peak is selected. Each extracted beat is stored in an array, serving as a representation of the feature (or input) for subsequent analysis. It is worth noting that abnormalities within individual beats may manifest as symptoms

of underlying maladies in the patient.

The goal of this study is to analyze the real efficacy of our algorithm in regardless of the QRS detection process. However, given the documented efficiency of the XQRS method and our desire for actual real-world use, we have chosen to use the R peaks generated by this technique. This option enables the smooth preparation of new, unlabeled, or unseen ECG recordings for input into our proposed model.
In the following figures, some of the extracted beats are shown.

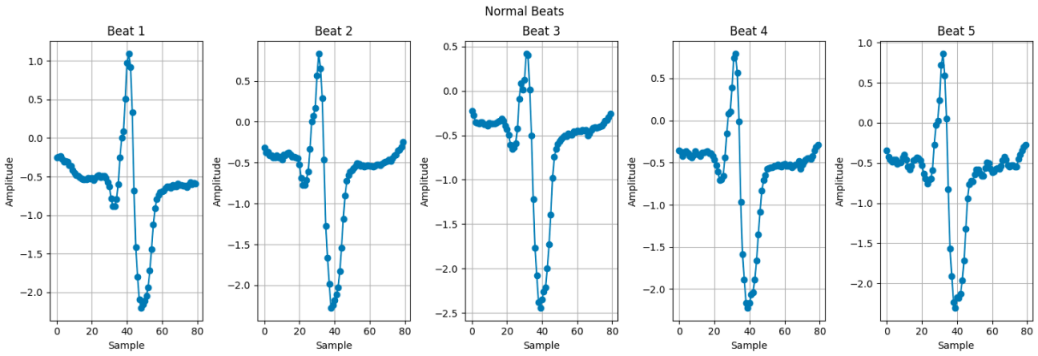


Figure 3-10: First five extracted beats of label 'Normal' of record '08378'

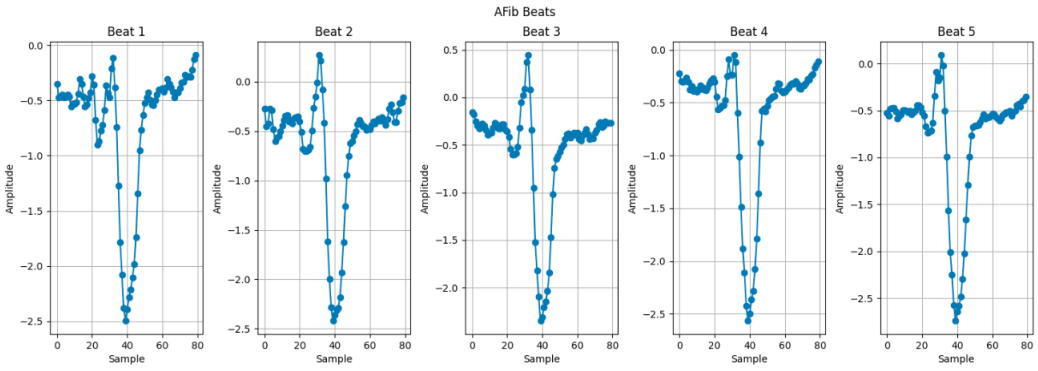


Figure 3-11: First five extracted beats of label 'AFib' of record '08378'

We can clearly notice that we need a normalization step before the CNN part, which will be done after organizing our data in csv files.

3.5.4 Data split

Partitioning our dataset is the next step after obtaining the features; subsequently, we add up the total occurrences for each category, namely, AFib and Normal. Additionally, we grouped these attributes with their labels to accelerate the process for later operations.

All the normal beats were placed in the CSV file. For each row, 80 cells were used. The total number of rows is the total number of normal beats in a 1h record in the database. Similarly, for AFib, the CSV file has the same shape as the first file.

Another two .CSV files for the labels (0 for AFib and 1 for normal).

Table 3.3 lists the total number of occurrences for each category. To prevent **classification bias**, we ensured that equal numbers of normal and AFib beats were included. This careful balancing of the dataset helps provide a fair comparison and accurate assessment of the classification model's performance for both classes.

Table 3.3: Data profile for the database after segmentation. The two classes are: Normal Sinus Rhythm (N), Atrial Fibrillation (AF).

Class	**N**	**AFib**
Training Beats	19853	19853
Validation Beats	2205	2205
Testing Beats	5515	5515

3.5.5 Normalization of ECG Signals

In order to facilitate effective deep learning classification, an additional pre-processing step called signal normalization was performed on MIT-BIH AFDB. The purpose of signal normalization was to ensure that all ECG signals had a standardized amplitude range between -1 and 1.

In our case we applied Gaussian normalization that involves standardizing the data by subtracting the mean and dividing by the standard deviation, then scaling it to the desired range.

Figures 3-12 and 3-13 show the five beats for normal and atrial fibrillation of record '08378' after normalization (we can compare with 3-10 and 3-11)

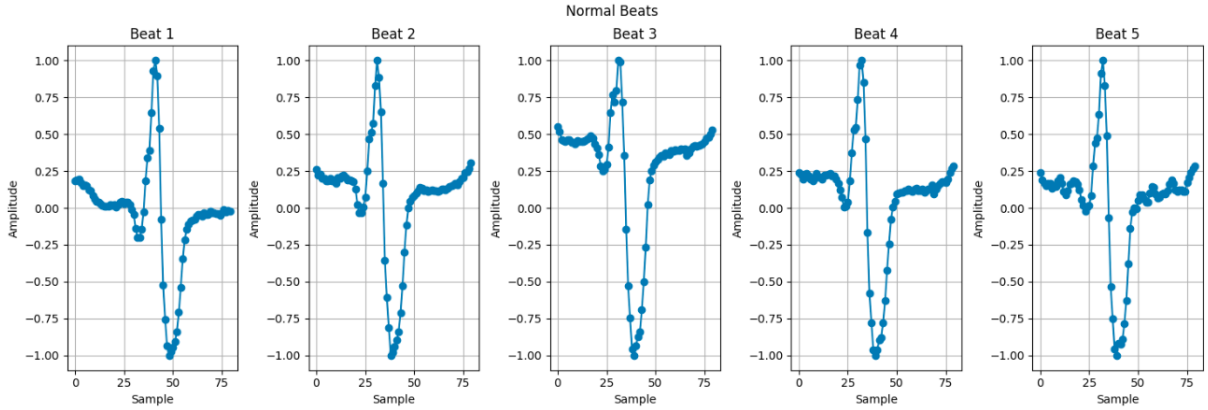


Figure 3-12: First five normalized beats of label 'Normal' of record '08378'

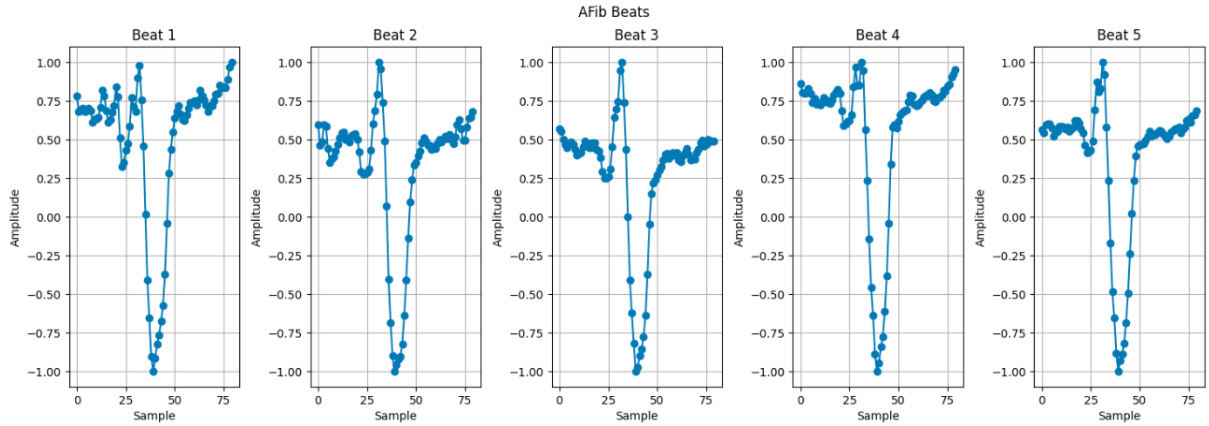


Figure 3-13: First five normalized beats of label 'AFib' of record '08378'

3.6 Proposed Models Description

3.6.1 Experiment 01: 1D-CNN proposed Model

In this section, we employ a 1D CNN classifier to explore the performance of the Convolutional Neural Network (CNN) model implemented using the Keras library. Designed for classification tasks on sequential data, the CNN model takes an input

shape of (80, 1). Its architectural design incorporates pivotal components such as convolutional layers, max pooling layers, dropout layers, and fully connected layers, thereby facilitating the adept extraction of features from the sequential data. The model is meticulously configured with dropout rates set at 0.5, filter sizes spanning 264 and 528 in the convolutional layers, and undergoes rigorous training for 300 epochs.

Below is the architectural layout of the model:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv1d_2 (Conv1D)	(None, 78, 264)	1,056
batch_normalization_1 (BatchNormalization)	(None, 78, 264)	1,056
max_pooling1d_2 (MaxPooling1D)	(None, 39, 264)	0
dropout_3 (Dropout)	(None, 39, 264)	0
conv1d_3 (Conv1D)	(None, 37, 528)	418,704
max_pooling1d_3 (MaxPooling1D)	(None, 18, 528)	0
flatten_1 (Flatten)	(None, 9504)	0
dense_2 (Dense)	(None, 128)	1,216,640
dropout_4 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

Figure 3-14: model.summary() output

We will employ hyperparameter tuning to refine the model's performance further. By adjusting the dropout rate, filter sizes, and number of epochs, we can explore alternative combinations to enhance the model's effectiveness. Specifically, we will vary the dropout rate within the range of 0.5 to 0.6, while adjusting filter sizes to 128 and 528 in selected scenarios. Additionally, the number of epochs will be varied between 200 and 300 for each configuration.

To ensure a robust evaluation of the model's performance, we will allocate 80% of the data for training, 10% for validation, and reserve 20% for testing. This partitioning strategy aims to optimize both the accuracy and efficiency of the model by

meticulously fine-tuning hyperparameters and conducting thorough training.

The results obtained from each configuration will be meticulously documented in a table format for straightforward comparison and in-depth analysis. Ultimately, this methodological approach will enable us to make informed decisions regarding the most suitable hyperparameter values for our 1-D CNN classifier.

Table 3.4: Model Performance over Different Hyper-parameters

Case	**Dropout Rate**	**Epochs**	**1st Conv Size**	**2nd Conv Size**	**Accuracy (%)**	**F1 (%)**
01	0.5	200	128	264	94.26	93.53
02	0.5	300	128	264	95.61	91.79
03	0.6	200	128	264	87.64	90.97
04	0.6	300	128	264	92.70	92.42
05	0.5	200	264	528	92.41	92.63
06	0.5	300	264	528	94.34	94.41
07	0.6	200	264	528	88.91	93.09
08	0.6	300	264	528	92.07	92.24

Note: It is worth noting that the weights in the model are initialized randomly. Consequently, re-running the program may yield different results each time. To ensure reproducibility and obtain consistent results across different computing environments, it is advisable to fix the random seed before training the model. By setting a fixed random seed, the same initial weights will be used for each run, resulting in consistent outcomes.

From Table 3.4, it is evident that the most optimal scenarios, post hyperparameter tuning, are represented by Case 01 and Case 06. To delve deeper into their performance, we will analyze the corresponding confusion matrix and convergence loss.

Training with Another Splitting Ratio

Following this, we proceeded to apply the model for the aforementioned cases (Case 01 and Case 06) utilizing an alternate data partitioning ratio: 70% allocated for training

and 30% for testing. Here are the outcomes derived from this execution:

Table 3.5: Model Performance with Different Splitting Ratio

Case	Dropout	Epochs	Filter Sizes	Accuracy (%)	F1 (%)
01	0.5	200	128/264	***93.35***	***92.43***
06	0.5	300	264/528	***90.79***	***91.88***

3.6.2 Classification Evaluation

Case 01: Results

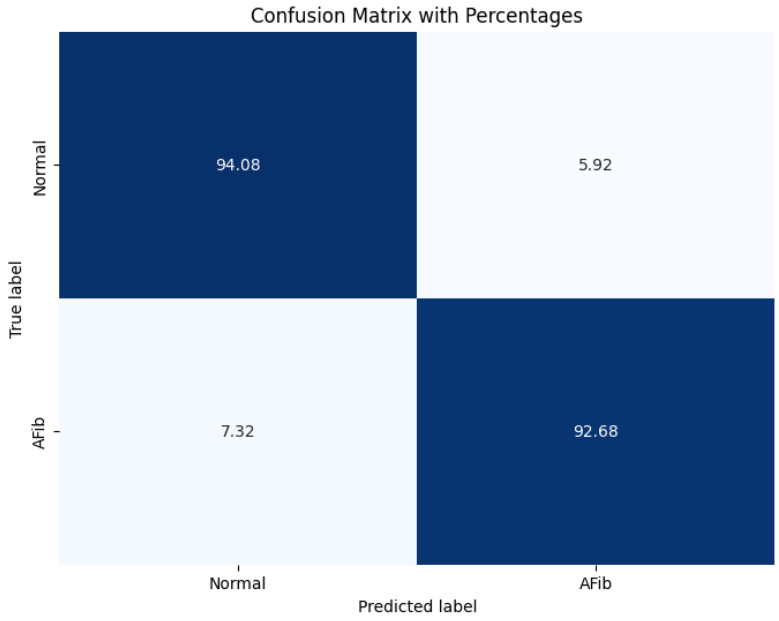


Figure 3-15: Confusion Matrix of Case 01

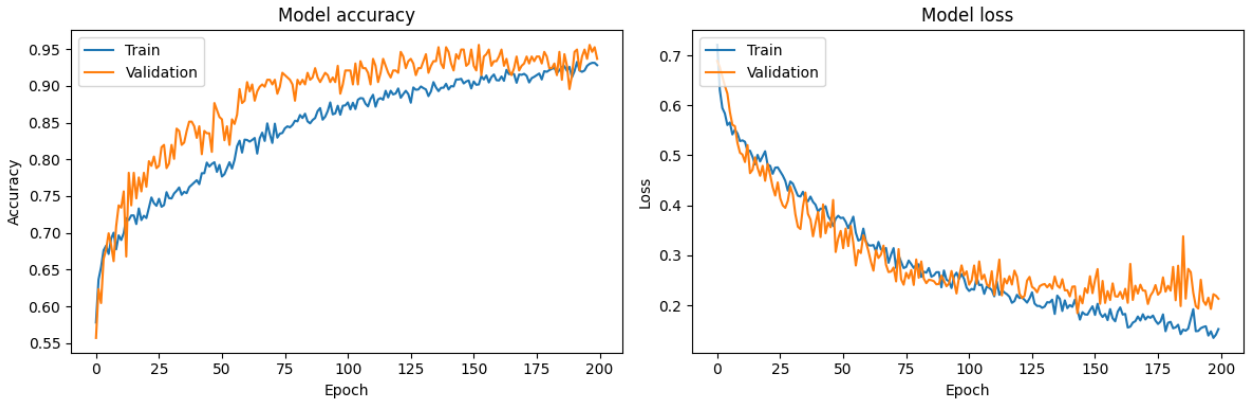


Figure 3-16: Convergence Accuracy and Loss Graphs for Case 01

Case 06: Results

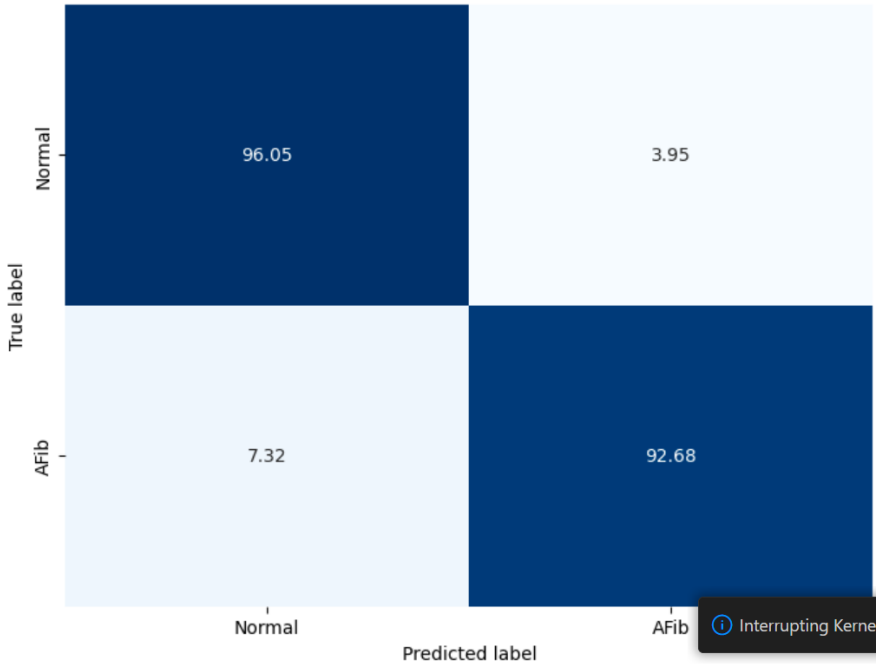


Figure 3-17: Confusion Matrix for Case 06

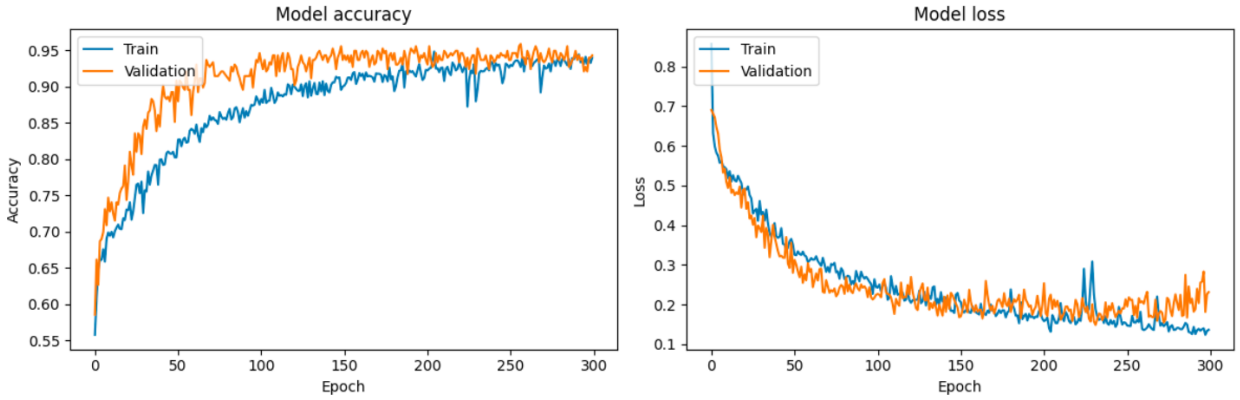


Figure 3-18: Convergence Accuracy and Loss Graphs for Case 06

We can clearly observe that model in case 06 gives best overall results.

Case 01 after Changing Splitting Ratio:

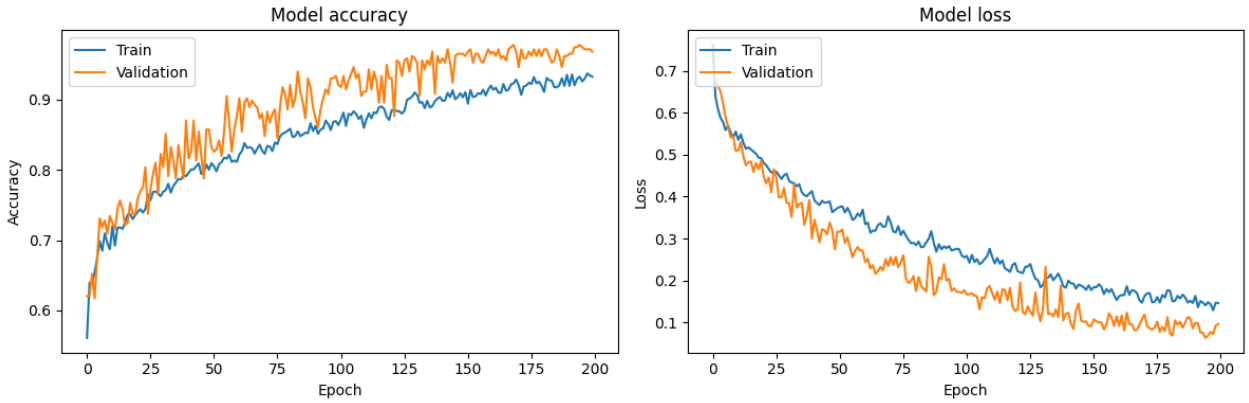


Figure 3-19: Convergence Accuracy and Loss Graphs

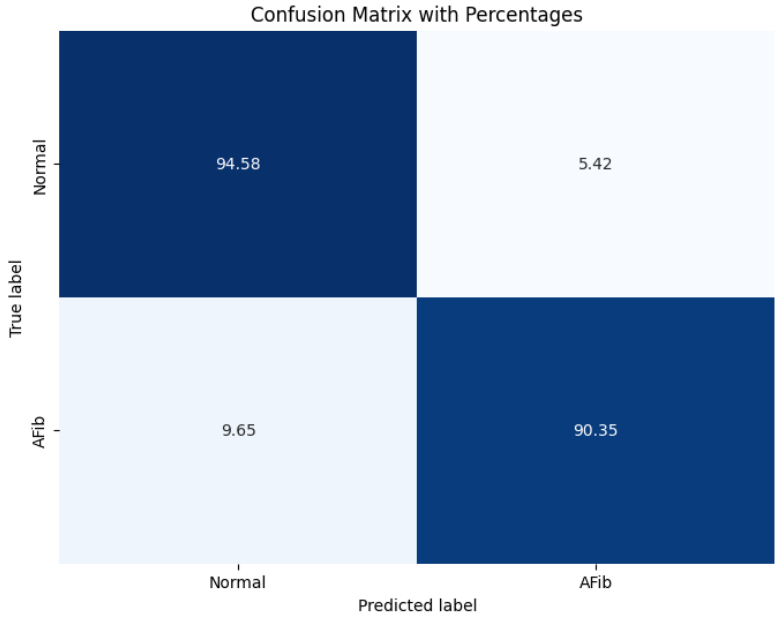


Figure 3-20: Confusion Matrix for case 01 after Changing Splitting Ratio

Case 06 after Changing Splitting Ratio:

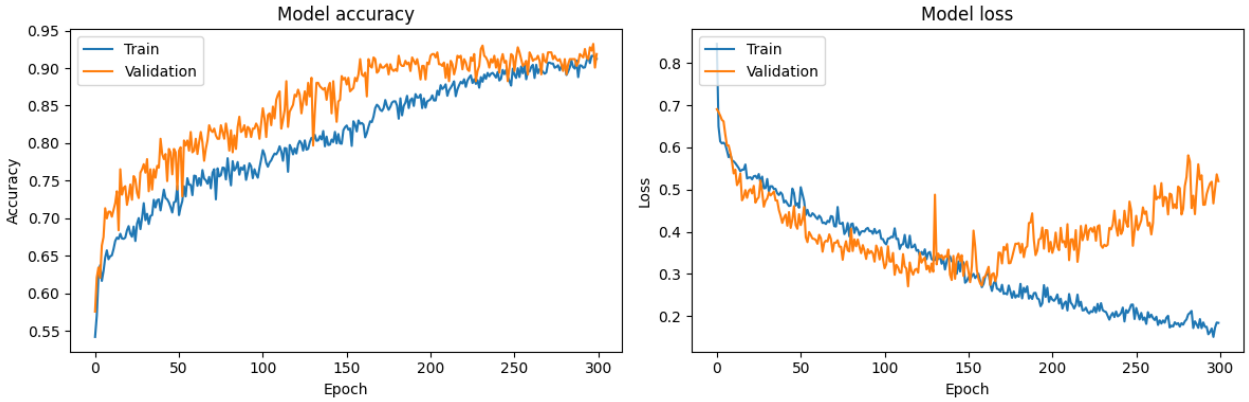


Figure 3-21: Convergence Accuracy and Loss Graphs

Note: This splitting ratio was not factored in, as it led to significant overfitting beyond epoch 150.

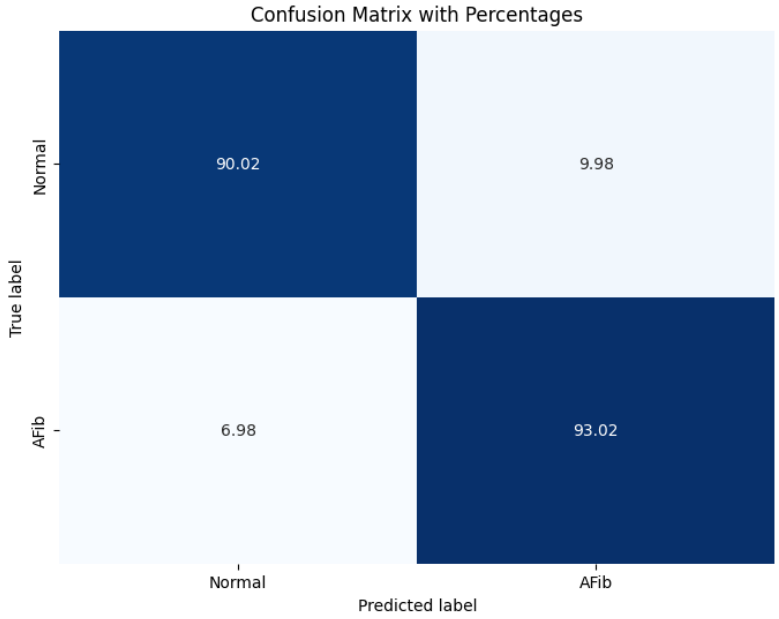


Figure 3-22: Confusion Matrix

3.6.2.1 Discussion

Case 01: The results from Case 01 of our CNN model show that it performed really well. We carefully adjusted some important settings to make sure the model works effectively. For example, we used a dropout rate of 0.5 to prevent the model from learning too much from the training data, and we trained it for 200 rounds to make sure it learns well.

We also chose specific sizes for some parts of the model, which helped it understand the data better and make more accurate predictions. The way we split the data for training, testing, and validation showed that the model can work well with different datasets. This consistency proves that the model is reliable.

Additionally, the graph showing how the model's performance improved over time without getting too good at just memorizing the training data indicates that the model learned well without getting too fixated on specific examples.

Overall, Case 01 of our CNN model shows that we have built a good model that can be used effectively for different classification tasks.

Case 06: This Case 06 stands out as one of the best-performing experimental

cases, showing strong potential for reliable performance under specific conditions. By using a dropout rate of 0.5 and training for 300 epochs, along with employing a first convolutional layer size of 264 and a second convolutional layer size of 528, Case 06 achieved impressive accuracy and F1 score metrics.

The standout feature of Case 06's performance is its remarkable F1 score, reaching 94.41%, which surpasses all other cases. This is significant because the F1 score provides a balanced measure of both precision and recall, demonstrating the model's effectiveness in handling false positives and false negatives.

It's important to note that while Case 06 performed exceptionally well with an 80-20 data split for training and testing, it showed signs of overfitting when the data split was changed to 70-30. During training, we noticed clear signs of overfitting as the validation loss started to increase after epoch 150, while the training loss continued to decrease. This indicated that the model was learning the training data too well, capturing noise and specific patterns that didn't generalize to unseen validation data.

This observation highlights how sensitive the model's performance is to the distribution and quality of the training data. Despite the overfitting observed with the alternative data split, Case 06 maintained its superiority when evaluated against the initial 80-20 data split. This underscores the importance of carefully considering data quality and partitioning strategies to achieve optimal model performance.

Furthermore, the learning curves associated with Case 06 show promising trends, with no significant overfitting except for a slight deviation in validation loss towards the end of training. This indicates that the model's ability to generalize is favorable, particularly when trained on the initial data split.

3.6.3 Experiment 02: Dropout Rate effect

We will now attempt to manually fine-tune the hyperparameters by adjusting the convolutional layer kernels to 64 and 128 while adding dropout after each layer with rates ranging from 0.2 to 0.5. This process aims to optimize the model's performance by systematically varying these parameters and evaluating their impact on accuracy and other performance metrics.

Layer (type)	Output Shape	Param #
conv1d_12 (Conv1D)	(None, 78, 64)	256
batch_normalization_13 (BatchNormalization)	(None, 78, 64)	256
max_pooling1d_12 (MaxPooling1D)	(None, 39, 64)	0
dropout_20 (Dropout)	(None, 39, 64)	0
conv1d_13 (Conv1D)	(None, 37, 128)	24,704
batch_normalization_14 (BatchNormalization)	(None, 37, 128)	512
max_pooling1d_13 (MaxPooling1D)	(None, 18, 128)	0
dropout_21 (Dropout)	(None, 18, 128)	0
flatten_10 (Flatten)	(None, 2304)	0
dense_14 (Dense)	(None, 128)	295,040
dropout_22 (Dropout)	(None, 128)	0
dense_15 (Dense)	(None, 1)	129

Figure 3-23: Proposed Model Summary

Table 3.6: Model Performance over varying droupout rates

Case	Dropout Rate	Epochs	1st Conv Size	2nd Conv Size	Accuracy (%)	F1 (%)
01	0.5	300	64	128	92.58	92.37
02	0.4	300	64	128	92.85	93.41
03	0.3	300	64	128	92.88	93.18
04	0.2	300	64	128	92.89	94.01

3.6.3.1 Results

In our first scenario, as depicted in Figure 3-24, we observed the most favorable performance in terms of learning curves, with no signs of overfitting.

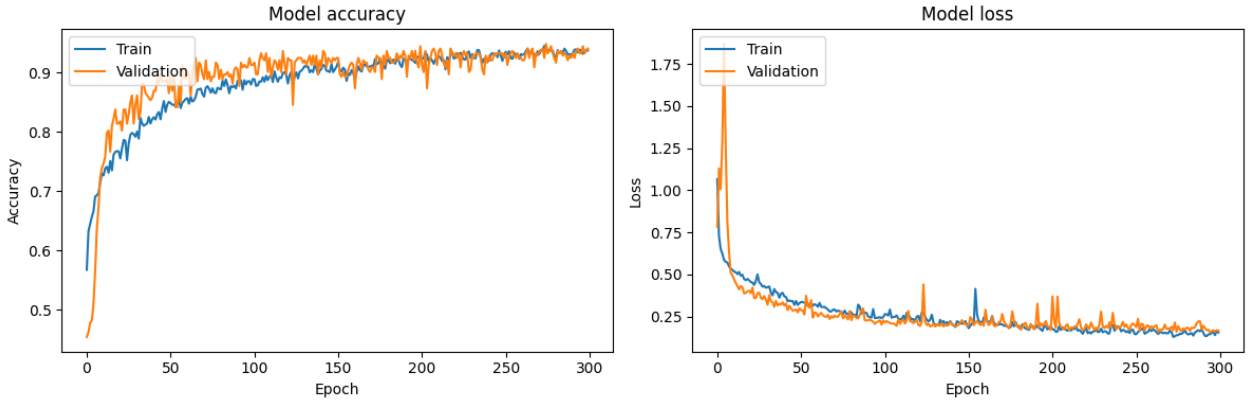


Figure 3-24: Learning Curves for Case 01

For Case 04, which yielded the highest F1 score, we observe a somewhat distorted learning curve towards the end, indicating evident overfitting (see Figure 3-25).

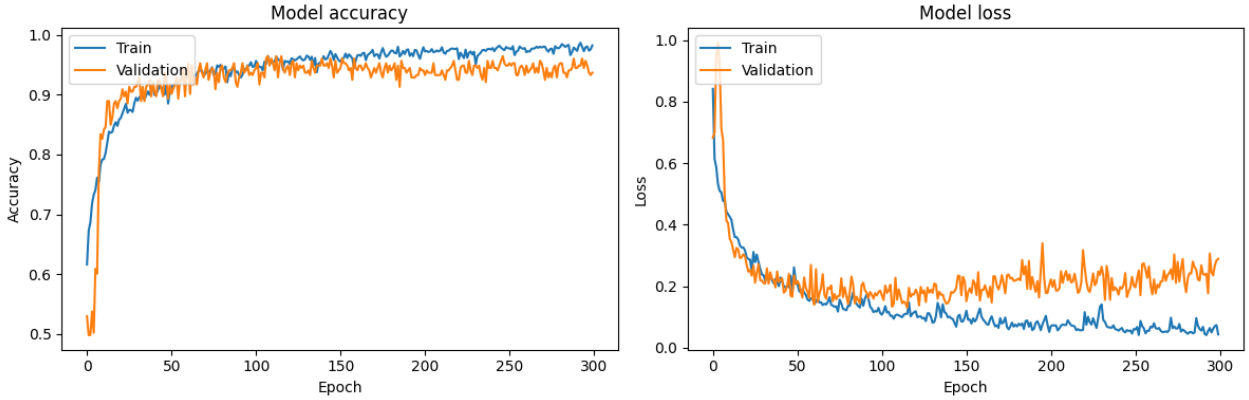


Figure 3-25: Learning Curves for Case 04

3.6.3.2 Discussion:

The proposed model architecture, illustrated in Figure 3-23, underwent thorough evaluation across various dropout rates. Over 300 epochs, the model's performance

was consistently assessed using accuracy and F1 score metrics. Table 3.6 summarizes the correlation between dropout rates and model performance.

Our findings underscore the pivotal role of dropout rates in determining both the model's accuracy and its susceptibility to overfitting. Remarkably, as the dropout rate decreased, accuracy improved. However, this led to an elevated risk of overfitting, highlighting the delicate balance between model complexity and generalization.

The learning curves depicted in Figure 3-24 reveal that Case 01 represents the most optimal configuration among the tested scenarios. This model achieved a remarkable equilibrium between accuracy and generalization, exhibiting minimal signs of overfitting. Conversely, Case 04, despite boasting the highest F1 score, exhibited pronounced overfitting, as evidenced by the distorted learning curve in Figure 3-25.

This experiment clearly shows that the dropout rate influences the model's accuracy; **as the dropout rate decreases, the accuracy increases, but an overfitting problem occurs.**

3.6.4 Expirement 03: Testing on Another Dataset

To evaluate the generalization capacity of the developed model, inter-subject testing was conducted using an independent dataset. The objective was to determine whether the model, trained on one dataset, could effectively classify data from a distinct source.

To ensure reproducibility and consistency across multiple runs, the seed value was set to 15000, ensuring that random initialization and data shuffling processes yield consistent results with each model training iteration.

To gauge the model's performance, a rigorous 5-fold cross-validation strategy was employed. This methodology entails partitioning the dataset into five subsets, employing four subsets for training and reserving one for validation in each fold. By iteratively repeating this process five times, with each subset serving as the validation set once, a robust evaluation of the model's performance is attained.

The selection of 5-fold cross-validation was driven by its equilibrium between com-

putational efficiency and statistical reliability. This approach facilitates a comprehensive appraisal of the model's efficacy while maintaining a manageable computational burden.

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9416 - loss: 0.2126
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9483 - loss: 0.1311
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9261 - loss: 0.2003
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9126 - loss: 0.2998
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9647 - loss: 0.1058
Average Test loss: 0.2049333244562149
Average Test accuracy: 0.932577395439148
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 14ms/step
F1 Score: 0.9457364341085271
```

Figure 3-26: Results after code modifications

Table 3.7: Summary of test accuracy after each fold

	Accuracy	**Loss**
Fold 1	0.9416	0.2126
Fold 2	0.9483	0.1311
Fold 3	0.9261	0.2003
Fold 4	0.9126	0.2998
Fold 5	0.9647	0.1058
Mean Test Values	0.9326	0.2049

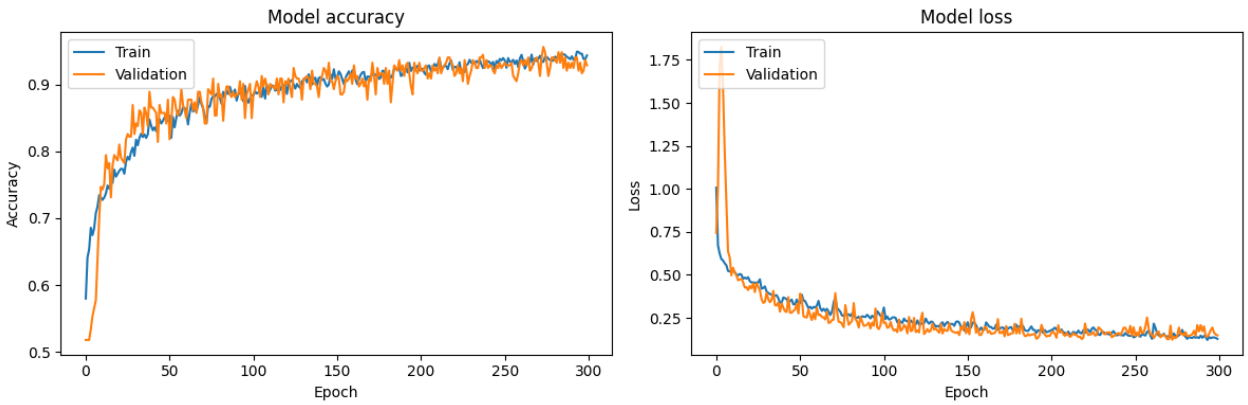


Figure 3-27: Convergence loss and accuracy graphs

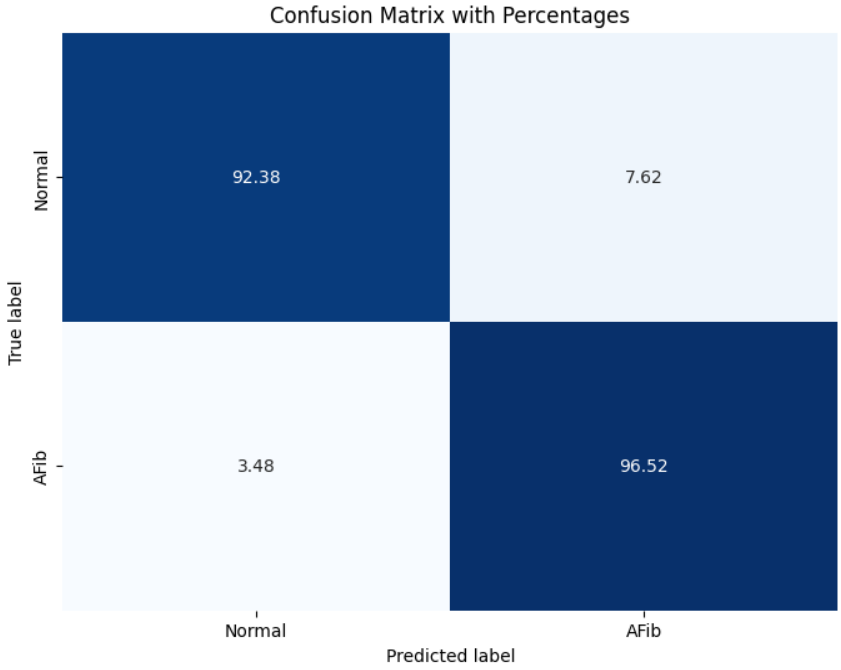


Figure 3-28: Confusion matrix

3.6.4.1 Results

The performance metrics obtained from the 5-fold cross-validation are summarized in Figures 3-26, 3-27, 3-28. The model achieved an average accuracy of 93.36% and an average loss of 0.2049 across the five folds. These results demonstrate the effectiveness of the proposed model in classifying cardiac signals, as evidenced by its high accuracy and low loss.

3.6.4.2 Testing on The 2017 PhysioNet/CinC Challenge Dataset

After evaluating the model's performance on the original dataset, we proceeded to assess its generalization ability using an external dataset, namely, the 2017 PhysioNet/CinC Challenge dataset available on the Physionet website. This dataset comprises a total of 5154 Normal and 771 AFib records and is widely utilized for benchmarking cardiac signal classification algorithms.

It's noteworthy that despite the sampling frequency of 300 samples per second in this dataset, which differs from that used in the Atrial Fibrillation Database (AFDB),

there's no necessity to resample the data. This is owing to the fact that our model isn't reliant on the specific sampling frequency, thus offering a distinct advantage. During model training, our focus was on feature extraction based on 80 samples around the R peak, a methodology that remains effective irrespective of the sampling frequency employed.

To evaluate the model's generalization capability, we saved the trained 1D-CNN model in the native Keras file format, ensuring reproducibility and facilitating future utilization. Subsequently, we loaded this saved model and subjected it to testing using a completely unseen dataset containing normal and atrial fibrillation (AF) beats. The new data underwent preprocessing to align with the format used during training, including reshaping to an appropriate input shape for the model. Performance evaluation of the model on this new dataset was conducted using metrics such as the F1 score and a confusion matrix, validating the model's proficiency in accurately classifying heartbeats and showcasing its resilience across diverse data sources.

```
# Save the model in native Keras format
model.save('cnn_model.keras')
print("Model saved to cnn_model.keras")

# Load the model
loaded_model = tf.keras.models.load_model('cnn_model.keras')
print("Model loaded from cnn_model.keras")

# Recompile the model with the same metrics
loaded_model.compile(optimizer='adam', loss='binary_crossentropy',
    metrics=['accuracy'])
```

Listing 3.5: Save and Load the Model

3.6.4.3 Results and Discussion

The F1 Score obtained from the evaluation of the model on the external dataset was calculated to be 0.891, indicating a strong balance between precision and recall. The confusion matrix for the classification results is as follows:

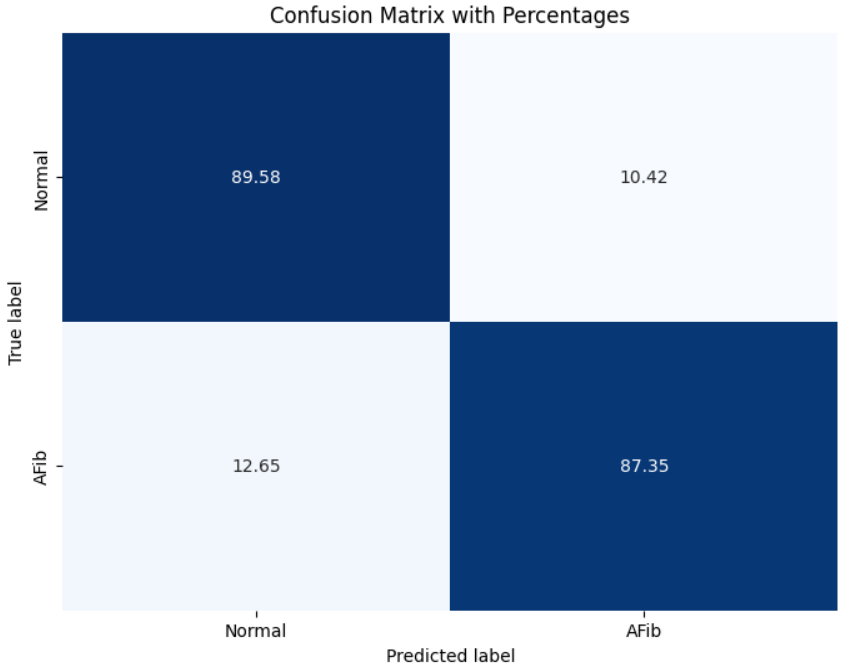


Figure 3-29: Confusion Matrix of The Test on unseen Data

Prior to testing on the completely unseen data, after model training, an F1 score of 94.57 was achieved, indicating high accuracy in classification tasks.

The obtained F1 score of 0.891 and the confusion matrix results provide insights into the model's performance on the external dataset. The relatively high F1 score indicates that the model achieved a good balance between precision and recall in classifying normal and atrial fibrillation (AFib) beats.

The confusion matrix reveals that the model performed well in identifying normal beats, with a high percentage of correctly classified normal beats (89.58%). However, there was a relatively higher misclassification rate for AFib beats as normal (12.65%) compared to normal beats misclassified as AFib (10.42%). This suggests that the model might have slightly more difficulty in correctly identifying AFib beats, potentially due to the complexity and variability of AFib patterns in the dataset.

These results could be attributed to several factors. First, the external dataset may contain variations in AFib patterns that differ from those present in the training dataset. Additionally, the model's performance may be influenced by the quality and

quantity of the data in the external dataset.

Overall, while the model demonstrated robustness and generalization capability across different data sources, further fine-tuning and optimization may be required to enhance its performance, particularly in accurately classifying AFib beats. Continued refinement of the model architecture and training strategies could lead to improved classification accuracy and reliability in real-world applications.

3.6.5 Experiment 04: Data Augmentation

Data augmentation is a technique used to artificially increase the size of a training dataset by applying various transformations to the existing data samples. These transformations introduce variations that help improve the robustness and generalization ability of machine learning models.

3.6.5.1 The 1D-CNN Proposed Model

In Experiment 3, we applied data augmentation to the model to observe its effect. Three common techniques were implemented:

- Random Scaling: Adjusts the signal's amplitude randomly.
- Random Shift: Shifts the signal along the time axis randomly.
- Random Noise Injection: Adds random noise to the signal.

These techniques aim to diversify the training data, potentially enhancing the model's ability to generalize and perform well on unseen data. Each data augmentation function is applied independently to each ECG signal sample in the dataset, resulting in multiple augmented versions of each original sample. By incorporating these augmented samples during training, the model becomes more resilient to variations in input data and is better able to generalize to unseen data.

Table 3.8: Evaluation Metrics Before and After Data Augmentation

Experiment 03	Accuracy (%)	F1 (%)
Before Data Augmentation	***93.26***	***94.57***
After Data Augmentation	***95.00***	***96.81***

```
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 13ms/step - accuracy: 0.9493 - loss: 0.2023
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9431 - loss: 0.1588
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.9583 - loss: 0.2179
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.9595 - loss: 0.1411
30/30 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.9504 - loss: 0.1813
Average Test loss: 0.1660364419221878
Average Test accuracy: 0.9499910831451416
30/30 ━━━━━━━━━━━━━━━━━━━━ 1s 22ms/step
F1 Score: 0.9680511182108626
```

Figure 3-30: Results after Data Augmentation

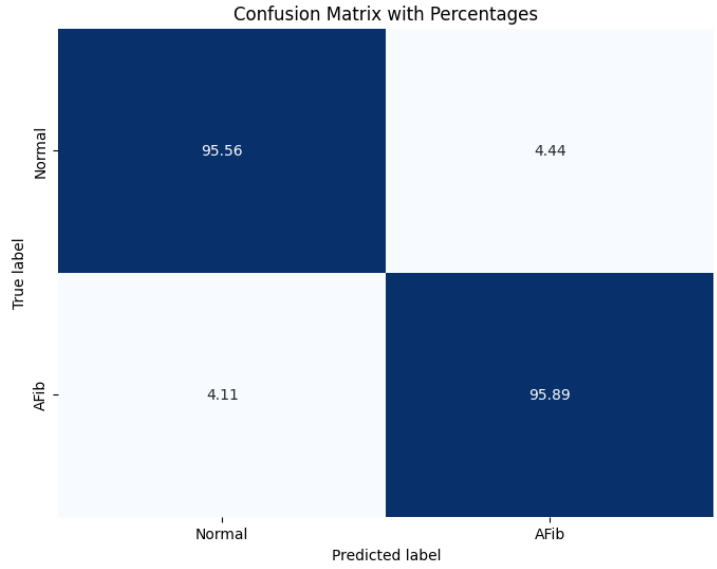


Figure 3-31: Confusion Matrix after Data Augmentation

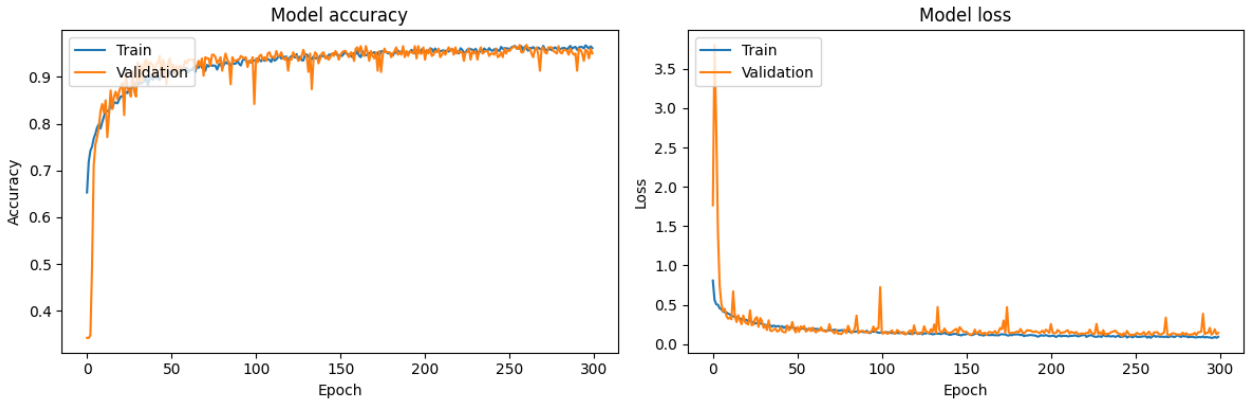


Figure 3-32: Convergence Loss and Accuracy Graphs after Data Augmentation

In our study, we employed a 5-fold stratified cross-validation strategy to ensure robust evaluation of our model's performance. This approach involves dividing the dataset into five equally sized folds while maintaining consistent class distributions across folds.

During each iteration of the cross-validation loop: (72% of the data was allocated for training, 8% for validation, and 20% for testing).

This partitioning scheme enabled us to train, validate, and test the model on different subsets of the data, facilitating thorough assessment of its performance and generalization ability.

Before applying data augmentation, our model achieved an accuracy of 93.26%. Following the introduction of data augmentation techniques, this accuracy notably improved to 95.00%. This enhancement indicates that data augmentation contributed significantly to the model's capacity to accurately classify instances, resulting in an overall higher accuracy rate.

Similarly, the F1 score, which considers both precision and recall, experienced a boost post data augmentation. Prior to augmentation, the F1 score stood at 94.57%, climbing to 96.81% afterward. This indicates that data augmentation not only improved the model's accuracy but also its ability to strike a balance between precision and recall, thereby enhancing its overall performance.

Our findings underscore the effectiveness of data augmentation in bolstering the

model's performance. The augmented data provided additional diverse examples for the model to learn from, resulting in improved generalization and robustness in classification tasks.

Testing The Model on Unseen Data After Data Augmentation Testing the model on unseen data after applying data augmentation yielded promising results. The F1 score, a metric that balances precision and recall, stood at 0.9019. This score indicates that the model's performance on the unseen data was robust, with a good balance between correctly identifying normal instances (N) and atrial fibrillation instances (A).

Examining the confusion matrix (shown in Figure 3-33) provides additional insights into the model's performance. The matrix reveals that the model correctly classified the majority of both normal (NN) and atrial fibrillation (AA) instances, with accuracies of 92.18% and 90.43%, respectively. However, there were instances of misclassification, as evidenced by the presence of NA and AN entries, representing cases where the model misclassified normal instances as atrial fibrillation and vice versa.

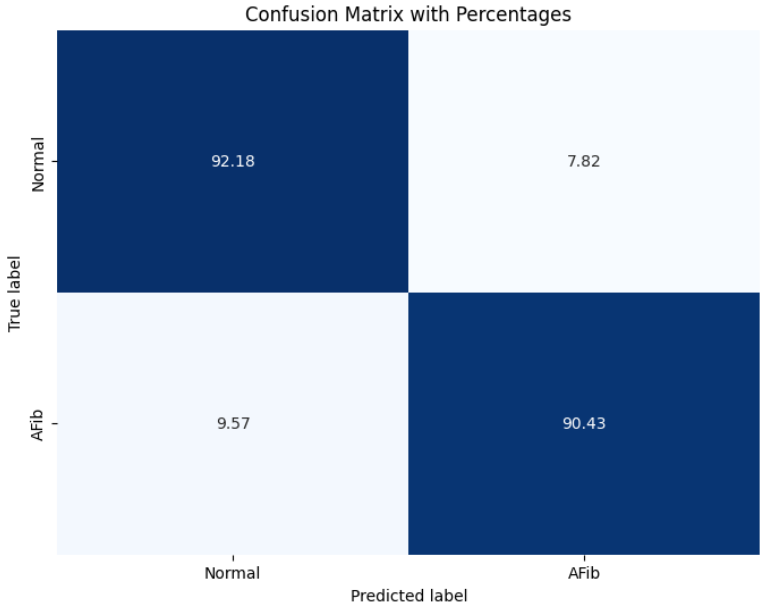


Figure 3-33: Confusion Matrix of The Test on unseen Data After Training with Data Augmentation

3.8 Comparison of Results with Existing Literature

Table 3.10: Comparison of Results with Literature

Model	**Feature**	**Methods**	**Se (%)**	**Sp (%)**	**Acc (%)**
[46]	5s ECG Records	1D-CNN+LSTM	84.89	84.89	80.23
[47]	30-min ECG	SVM	81.1	79.3	80.2
[48]	10s ECG Record	1D CNNs, LiteVGG-11	86.53	71.69	77.91
[48]	501×936 grayscale image	2D CNNs, EfficientNet-B2	75.74	74.76	75.20
Proposed Method 01:	Single beat of 80 samples	1D-CNN	**95.87**	**95.56**	**95.00**
Proposed Method 02:	Single beat of 80 samples	1D-CNN	**90.06**	**92.43**	**90.19**
Proposed Method 03:	Single beat of 80 samples	2D-CNN	**100**	**99.69**	**99.84**

3.9 Conclusion

In this chapter, we presented our 1D-CNN model, which showed impressive performance with 95% accuracy and an F1 score of 96.81%. We then converted beats into 8x10 grayscale images and applied them to a 2D-CNN, yielding competitive results comparable to previous studies.

General Conclusion

Our project titled "Atrial Fibrillation Analysis by Deep Learning " aimed to explore the potential of deep learning techniques in identifying atrial fibrillation (AFib).

In our research, we initially focused on preprocessing the ECG signals by evaluating various QRS detection algorithms. Ultimately, we selected XQRS due to its superior performance and ability to filter ECG signals effectively. Following this, we employed beat segmentation, utilizing an 80-sample window around the R peaks, and normalization to enhance data quality.

For the classification phase, we introduced a novel 1D-CNN architecture designed to accurately extract crucial features from the beats. Our model achieved impressive results, with an overall accuracy of 95.00% and an F1 score of 96.81% on the MIT-BIH AFDB dataset. Furthermore, it demonstrated robust performance on totally unseen data from The PhysioNet CinC 2017 Challenge for AFib, achieving an F1 score of 90.19%. This success validates the effectiveness of our model on unseen data and fulfills our primary objective of utilizing an inter-subject scheme.

Subsequently, to further enhance our model's performance, we proposed a 2D-CNN model. By converting beats into grayscale images as features, this approach yielded remarkable results, with an F1 score of 99.85% and an accuracy of 99.84%. These findings challenge many state-of-the-art and published articles in the field.

Future Work

Due to time constraints, we were unable to deploy the model as originally intended. For future endeavors, we recommend initially testing the proposed 2D-CNN model on unseen data, ensuring that the preprocessing stage and reshaping of beats into images are appropriately done to match the model's input requirements. Afterward, exploring algorithms to compress the model's size would be beneficial for hardware implementation. Furthermore, integrating electrodes and a compatible architecture on a Raspberry Pi could aid in deploying the model, aligning with the goal of reducing expenses linked to monitoring devices.

Bibliography

[1] World Health Organization. *Cardiovascular diseases (CVDs)*. https://www.who.int/. [Online; accessed 04-Feb-2024]. 2020.

[2] Mayo Clinic. *Atrial Fibrillation*. https://www.mayoclinic.org/. [Online; accessed 30-April-2024]. 2024.

[3] Mayo Clinic. *Chambers and Valves of the Heart*. https://www.mayoclinic.org/. [Online; accessed 30-March-2024]. 2024.

[4] Arthur C. Guyton and John E. Hall. Textbook of Medical Physiology. Saunders Elservier, 2006.

[5] *L'électrocardiogramme*. http://entraide-esi-ide.com/lelectrocardiogramme/. [Online; accessed 02-Feb-2024]. 2020.

[6] pfizer. *Flashback: The First ECG*. URL: https://www.pfizer.com/news/articles/flashback_the_first_ecg. (accessed: 22.04.2024).

[7] American heart association. *Arrhythmia*. www.heart.org. [Online; accessed 24-Apr-2024]. 2020.

[8] *The ultimate ECG book, ECG & ECHO*. https://ecgwaves.com/. [Online; accessed 02-Feb-2024]. 2020.

[9] Lara Ortiz-Martin et al. "Heartbeats do not make good pseudo-random number generators: An analysis of the randomness of inter-pulse intervals". In: *Entropy* 20.2 (2018), pp. 1–19.

[10] Leif Sörnmo and Pablo Laguna. *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. 2005.

[11] Rémi Dubois. "Application des nouvelles méthodes d'apprentissage à la détection précoce d'anomalies en électrocardiographie". In: *These de Doctorat de l'Universite Pierre et Marie Curie* (2004).

[12] Valentin Fuster et al. "ACC/AHA/ESC guidelines for the management of patients with atrial fibrillation: executive summary: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the European Society of Cardiology Committee for Practice Guidelines and Policy Conferences (Committee to Develop Guidelines for the Management of Patients With Atrial Fibrillation) Developed in collaboration with the North American Society of Pacing and Electrophysiology". In: *Journal of the American College of Cardiology* 38.4 (2001), pp. 1231–1265.

[13] Aakash Healthcare. `https://aakashhealthcare.files.wordpress.com/2019/03/heart-arrhythmia-treatment-in-delhi.jpg`. [Online; accessed 02-Feb-2024]. 2020.

[14] *Atrial Fibrillation, Cardiosmart.org*. `https://www.cardiosmart.org/heart-conditions/atrial-fibrillation`. [Online; accessed 14-Feb-2024]. 2020.

[15] Paulus Kirchhof et al. "2016 ESC Guidelines for the management of atrial fibrillation developed in collaboration with EACTS". In: *European journal of cardio-thoracic surgery* 50.5 (2016), e1–e88.

[16] Craig T January et al. "2014 AHA/ACC/HRS guideline for the management of patients with atrial fibrillation: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the Heart Rhythm Society". In: *Journal of the American College of Cardiology* 64.21 (2014), e1–e76.

[17] *"Persistent Atrial Fibrillation: Symptoms, Treatment, and Prognosis, Healthline*. `https://www.healthline.com/health/atrial-fibrillation/persistent#symptoms`. [Online; accessed 20-Feb-2024]. 2020.

[18] D George Wyse et al. "Lone atrial fibrillation: does it exist?" In: *Journal of the American College of Cardiology* 63.17 (2014), pp. 1715–1723.

[19] Paul A Rogers et al. "Current evidence-based understanding of the epidemiology, prevention, and treatment of atrial fibrillation". In: *Current problems in cardiology* 43.6 (2018), pp. 241–283.

[20] Massimo Zoni-Berisso et al. "Epidemiology of atrial fibrillation: European perspective". In: *Clinical epidemiology* 6 (2014), p. 213.

[21] Hooman Kamel et al. "Atrial fibrillation and mechanisms of stroke: time for a new model". In: *Stroke* 47.3 (2016), pp. 895–900.

[22] Francesca Pistoia et al. "The epidemiology of atrial fibrillation and stroke". In: *Cardiology clinics* 34.2 (2016), pp. 255–268.

[23] C. H. S. Scotland. ""Diagnosis and Treatment of Atrial Fibrillation,"". In: (2021).

[24] Jie Lian, Lian Wang, and Dirk Muessig. "A simple method to detect atrial fibrillation using RR intervals". In: *The American journal of cardiology* 107.10 (2011), pp. 1494–1497.

[25] Juan Benezet-Mazuecos, Camila S García-Talavera, and José Manuel Rubio. "Smart devices for a smart detection of atrial fibrillation". In: *Journal of Thoracic Disease* 10.Suppl 33 (2018), S3824.

[26] Michael V McConnell et al. "Mobile health advances in physical activity, fitness, and atrial fibrillation: moving hearts". In: *Journal of the American College of Cardiology* 71.23 (2018), pp. 2691–2701.

[27] Shreyanshu Sundaray. *History of Deep Learning*. Medium, Accesseced 11/06/2024. July 2023. URL: https://medium.com/@sreyan806/history-of-deep-learning-c176e2d3cddf#:~:text=The%20evolution%20of%20deep%20learning,the%20foundation%20for%20Deep%20Learning.

[28] Jeremy Norman. *Athena's Owl*. https://www.historyofinformation.com. Image of McCulloch (right) and Pitts (left) in 1949, retrieved from https://www.semanticscholar.org. 2023.

[29] Aston Zhang et al. *Dive into Deep Learning*. Interactive deep learning book with code, math, and discussions. Implemented with PyTorch, NumPy/MXNet, JAX, and TensorFlow. Adopted at 500 universities from 70 countries. d2l.ai, 2021. URL: https://d2l.ai.

[30] L. Alzubaidi, J. Zhang, and A.J. et al. Humaidi. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8.1 (2021), p. 53. DOI: 10.1186/s40537-021-00444-8.

[31] Muhammed Nizam. *Neural Networks: From Biological to Artificial*. https://www.linkedin.com/pulse/neural-networks-from-biological-artificial-muhammed-nizam-pcxwf/. Accessed: 2024-06-05. 2023.

[32] IBM. *What is a Neural Network?* Accessed: 2024-06-06. 2024. URL: https://www.ibm.com/topics/neural-networks#:~:text=Every%20neural%20network%20consists%20of,own%20associated%20weight%20and%20threshold..

[33] Vinay Ramachandran. *Activation Functions in Neural Networks*. Accessed: 2024-06-06. 2019. URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[34] P. De Chazal, M. O'Dwyer, and R.B. Reilly. "Automatic classification of heartbeats using ECG morphology and heartbeat interval features". In: *IEEE Transactions on Biomedical Engineering* 51.7 (2004), pp. 1196–1206. DOI: 10.1109/TBME.2004.827359.

[35] A. Sellami and H. Hwang. "A robust deep convolutional neural network with batch-weighted loss for heartbeat classification". In: *Expert Systems with Applications* 122 (2019), pp. 75–84. DOI: 10.1016/j.eswa.2018.12.037.

[36] F. Murat et al. "Review of Deep Learning-Based Atrial Fibrillation Detection Studies". In: *International Journal of Environmental Research and Public Health* 18.21 (2021), p. 11302. DOI: 10.3390/ijerph182111302.

[37] *Python (programming language) — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/wiki/Python_(programming_language). [Online; accessed 15-06-2024]. 2020.

[38] Wikipedia contributors. *Visual Studio Code*. https://en.wikipedia.org/wiki/Visual_Studio_Code. Accessed: 2024-05-27. 2024.

[39] *Anaconda (Python distribution) — Wikipedia, The Free Encyclopedia*. https://fr.wikipedia.org/wiki/Anaconda_(Python_distribution). [Online; accessed 15-06-2024]. 2020.

[40] Mohammad Hossin and Sulaiman M.N. "A Review on Evaluation Metrics for Data Classification Evaluations". In: *International Journal of Data Mining & Knowledge Management Process* 5 (Mar. 2015), pp. 01–11. DOI: 10.5121/ijdkp.2015.5201.

[41] DigitalOcean. *Confusion Matrix in R*. https://www.digitalocean.com/community/tutorials/confusion-matrix-in-r.

[42] A. Dutt. *Performance Evaluation Metrics for Classification*. Accessed on: Insert date accessed. 2020. URL: https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html.

[43] Ary L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals". In: *Circulation [Online]* 101.23 (2000), e215–e220.

[44] MF Issa et al. "Heartbeat classification based on single lead-II ECG using deep learning". In: *Heliyon* 9.7 (July 2023), e17974. DOI: 10.1016/j.heliyon.2023.e17974.

[45] Chengyu Xie et al. *Waveform Database Software Package (WFDB) for Python (version 4.1.0)*. PhysioNet. 2023. URL: https://doi.org/10.13026/9njx-6322.

[46] Yongjie Ping et al. "Automatic Detection of Atrial Fibrillation Based on CNN-LSTM and Shortcut Connection". In: *Computational and Mathematical Methods in Medicine* 2021 (2021), p. 8827670. DOI: 10.1155/2021/8827670. URL: https://doi.org/10.1155/2021/8827670.

[47] Kok-Hwee Boon et al. "Paroxysmal atrial fibrillation prediction method with shorter HRV sequences". In: *Computer Methods and Programs in Biomedicine* 134 (2016), pp. 187–196. DOI: 10.1016/j.cmpb.2016.07.015. URL: https://doi.org/10.1016/j.cmpb.2016.07.015.

[48] Estela Ribeiro et al. "Can Deep Learning Models Differentiate Atrial Fibrillation from Atrial Flutter?" In: *bioRxiv* (2023). Preprint. DOI: 10.1101/2023.08.08.23293815. URL: https://doi.org/10.1101/2023.08.08.23293815.