



SAPIENZA  
UNIVERSITÀ DI ROMA

# Cardiac arrhythmias classification in a low-power processor with TensorFlow Lite

Department of Computer, Control, and Management Engineering  
Master of Science in Engineering in Computer Science

Candidate

Lorella Landi

ID number 1912333

Thesis Advisor

Prof. Ioannis Chatzigiannakis

Academic Year 2021/2022

# Contents

|   |    |
|---|----|
| Contents.....   | i  |
| Introduction .....  | 1  |
| 1. Cardiological fundamentals .....   | 3  |
| 1.1 The heart activity .....  | 3  |
| 1.2 ECG's history .....   | 5  |
| 1.3 The ECG Waveform .....  | 8  |
| 1.4 The 12-leads ECG .....  | 10 |
| 1.5 Arrhythmia types .....  | 11 |
| 1.5.1 Sinus .....   | 12 |
| 1.5.2 Atrial .....  | 16 |
| 1.5.3 Junctional.....   | 19 |
| 1.5.4 Ventricular.....  | 22 |
| 1.5.5 AV Blocks .....   | 25 |
| 2. Datasets.....  | 27 |
| 2.1 MIT-BIH Arrhythmia Database.....  | 28 |
| 2.2 AHA Database for Evaluation of Ventricular Arrhythmia<br>Detectors..... | 33 |
| 2.3 European ST-T database.....   | 34 |
| 2.4 PhysioNet .....   | 38 |
| 2.4.1 Moody PhysioNet Challenges.....                                       | 38 |
| 2.4.2 2020 PhysioNet Challenge: Classification of 12-lead ECGs ...          | 38 |
| 2.5 Conclusion.....   | 44 |
| 3. Previous related work .....  | 46 |
| 3.1 Arrhythmia detection.....   | 46 |
| 3.2 2020 PhysioNet Challenge papers.....                                    | 48 |
| 4. Techniques and tools .....   | 51 |
| 4.1 Classification techniques and tools.....                                | 51 |
| 4.1.1 Data Wrangling (DW).....  | 51 |
| 4.1.2 Feature Engineering (FE) .....  | 53 |

|       |   |    |
|-------|---|----|
| 4.1.3 | Challenge features.....                               | 53 |
| 4.1.4 | Spectral features .....                               | 53 |
| 4.1.5 | Exploratory Data Analysis (EDA).....                  | 55 |
| 4.1.6 | Unbalanced classes .....                              | 56 |
| 4.1.7 | Machine Learning models.....                          | 58 |
| 4.1.8 | Metrics.....  | 63 |
| 4.2   | TensorFlow Lite for Microcontrollers.....             | 67 |
| 4.2.1 | The importance of microcontrollers .....              | 67 |
| 4.2.2 | TensorFlow.....                                       | 68 |
| 4.2.3 | TensorFlow Lite.....                                  | 69 |
| 4.2.4 | Model optimization .....                              | 70 |
| 5.    | Arrhythmia classification on a low-power device ..... | 73 |
| 5.1   | Arrhythmia classification .....                       | 73 |
| 5.1.1 | Dataset .....   | 73 |
| 5.1.2 | Data wrangling.....                                   | 73 |
| 5.1.3 | EDA .....   | 74 |
| 5.1.4 | Feature selection and normalization .....             | 78 |
| 5.1.5 | Balancing classes .....                               | 80 |
| 5.1.6 | Fitted models and results.....                        | 81 |
| 5.2   | Deploy on a microcontroller .....                     | 86 |
| 5.2.1 | ESP-EYE board .....                                   | 86 |
| 5.2.2 | Model conversion.....                                 | 87 |
| 5.2.3 | Building, flashing, and monitoring .....              | 91 |
|       | Conclusions.....                                      | 93 |
|       | References.....                                       | 94 |

# Introduction

The World Health Organization estimated that 17.9 million people die each year from cardiovascular diseases (CVDs), an estimated 32% of all deaths worldwide. 85% of all CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age [1]. One of these diseases is arrhythmia, which results from an electrical malfunction in the heart's cardiac signal. Arrhythmias may occur in a healthy heart and be of minimal consequence, but they can also be a signal of a significant issue that could lead to a stroke or sudden cardiac death.

The electrocardiogram (ECG), which measures the electrical activity of the heart, is a common technique for diagnosing arrhythmias. Given that this test is non-invasive and painless for the patients, it may be used to gather enormous amounts of data, which can then be analyzed to detect arrhythmias. Cardiologists use their knowledge and experience to carefully evaluate the ECG; however, their activity is time-consuming and prone to error: an automatic approach can support them during their decision.

Furthermore, automatic arrhythmia detection may be essential in clinical cardiology, particularly when carried out in real-time, since it might identify early heart problems and prevent dangerous repercussions. A Machine Learning (ML) approach may be helpful to achieve this goal.

With the intention of implementing them on wearable devices in the future, I proposed some ML algorithms capable of automatically classifying arrhythmias starting from 12-lead ECGs on a microcontroller. I put the focus on fitting ML methods on microcontrollers because the latter are low-power devices with long battery life, and in our scenario is important that the real-time classification is not constrained by power but can last as long

as possible. Additionally, because microcontrollers are typically small, it is possible to incorporate them into wearable devices. Again, because they don't have to share the data externally, they can also meet the requirements of privacy. Lastly, they are inexpensive, which is crucial because, according to the WHO, more than 75% of CVD deaths occur in low- and middle-income countries [1].

In order to accurately explain the characteristics of the project, the thesis is structured as follow:

- In Chapter 1 is described the essential background to understand the cardiological basics: the human heart activity, the concepts associated with ECG monitoring, and the most common arrhythmia types.
- The Chapter 2 introduces the available and most used ECG datasets.
- The Chapter 3 focuses on the state-of-the-art regarding the research topic.
- In Chapter 4 are explained in detail the techniques and tools used.
- The Chapter 5 describes the procedure and the results of the arrhythmia classification on a low-power device.

# 1. Cardiological fundamentals

[2] "An electrocardiogram (ECG) is a measure of how the electrical activity of the heart changes over time, as action potentials within each myocyte propagate throughout the heart as a whole during each cardiac cycle. In other words, the ECG is the recording of the cumulative signals produced by populations of cells eliciting changes in their membrane potentials at a given point in time. The ECG provides specific waveforms of electrical differences when the atria and ventricles depolarize and repolarize."

## 1.1 The heart activity

The human heart circulates blood throughout the body as a result of electrical impulses and mechanical responses caused by conductive and mechanical cells. Positive and negative electrical charges are balanced at the start of the cycle when the heart is in a resting state. There is no electrical flow in this condition, which is known as the polarized state. A difference in potential between the inside and outside of the heart is required for the organ to receive the stimulus and begin beating. When the charges inside and outside the earth trade places, electricity flows through the heart in a wave-like motion. This wave is known as depolarization, which refers to the process of electrical discharge and flow of electrical activity. Repolarization is the process that occurs after depolarization and returns the heart to its initial state.

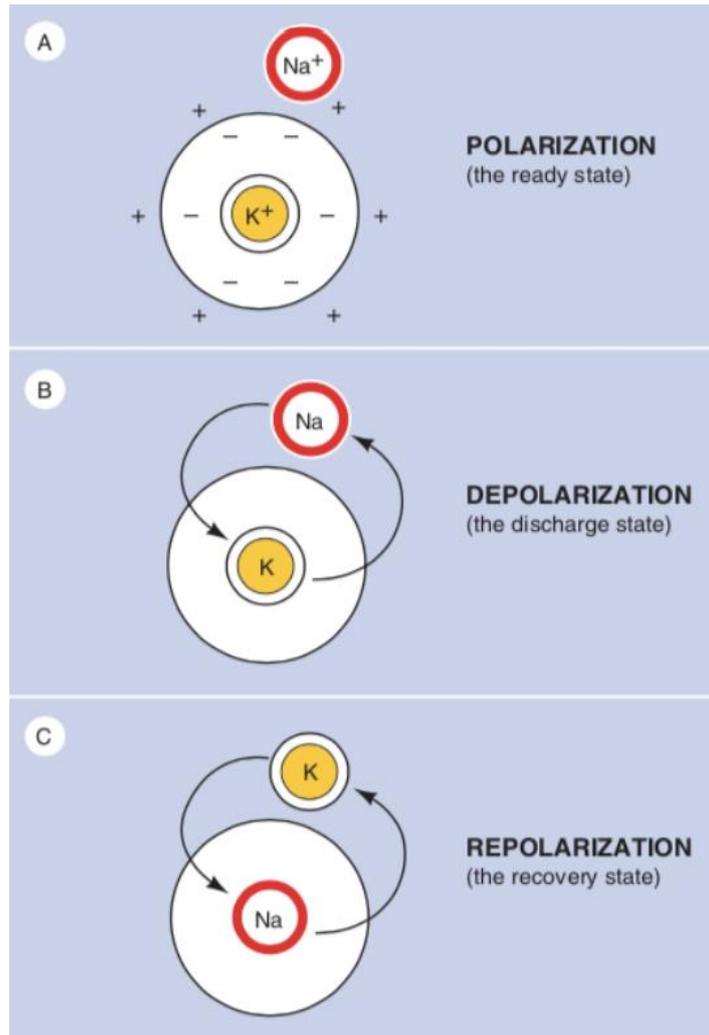


Figure 1.1 – Electrical cell transactions

The system which causes the contraction of the heart muscle is called the electrical conduction system of the heart (Figure 1.2). Normally, the electrical impulse originates in the Sinoatrial (SA) node and travels to the ventricles by way of the AV node; after leaving it, the impulses go through the Bundle of His to reach the right and left bundle branches, located within the right and left ventricles. At the terminal ends of the bundle branches, small fibers, called Purkinje fibers, distribute the electrical impulses to muscle cells to stimulate contraction.

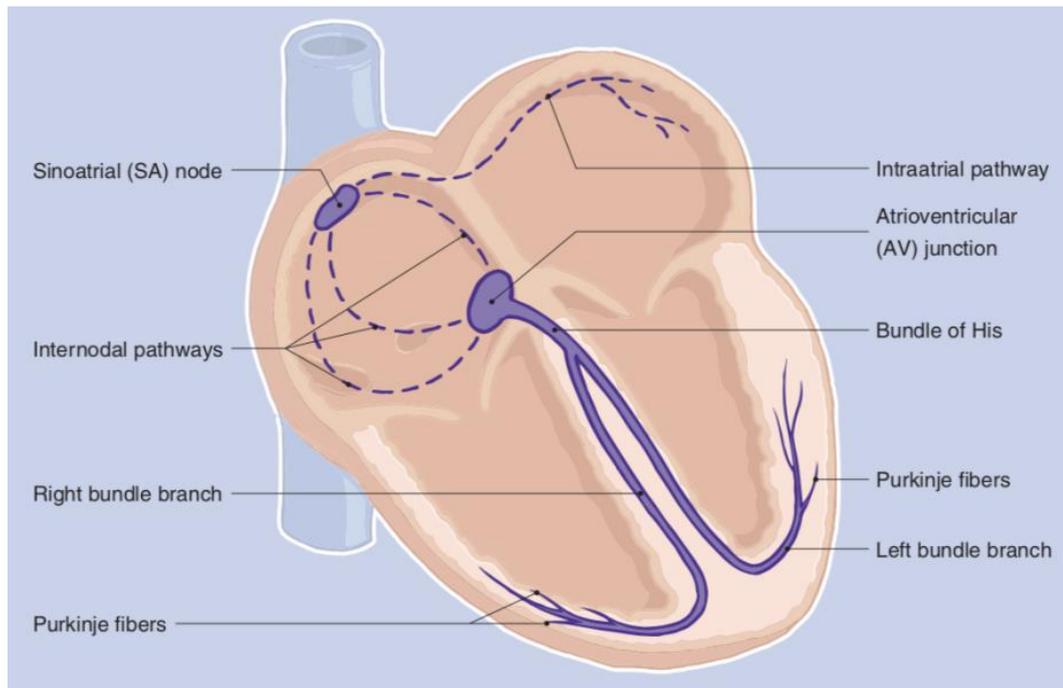


Figure 1.2 - Electrical conduction system

The electrical patterns of the heart can be picked up from the surface of the skin by attaching an electrode and connecting it to a machine that will display the electrical activity on graph paper, giving rise to the ECG [2].

## 1.2 ECG's history

The discovery of intrinsic electrical activity within the heart dates all the way back to the 1840s. Carlo Matteucci, an Italian physicist, was the first to discover that each heartbeat is accompanied by an electrical current in 1842. Emil DuBoisReymond, a German scientist, published the first action potential associated with muscular contraction not long after. In 1856, Rudolph von Koelliker and Heinrich Miller used a galvanometer to record the first cardiac action potential. Following that, Augustus D. Waller recorded the first human ECG after Gabriel Lippmann invented the

capillary electrometer in the early 1870s. That first device is shown in Figure 1.3.

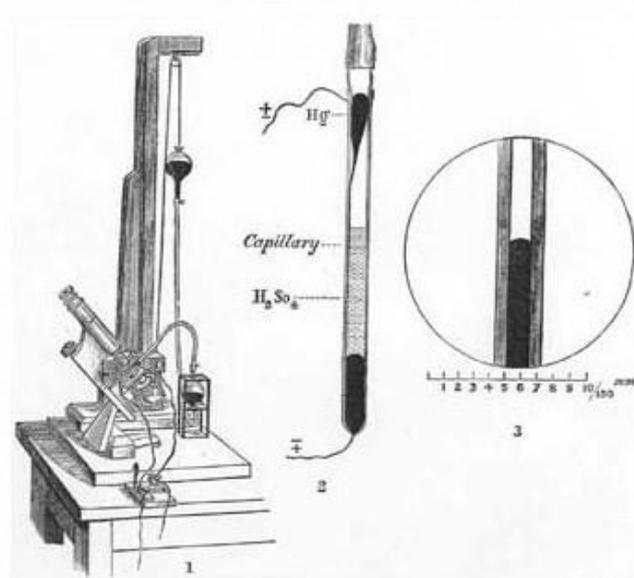


Figure 1.3 - Lippmann electrometer

Willem Einthoven's creation of the string galvanometer in 1901 was a key milestone in cardiac electrocardiography. The next year, he published the first ECG using his string galvanometer. Einthoven's string galvanometer consisted of a huge electromagnet with a thin silver-coated string stretched across it; electric currents passing through the thread caused the string to move from side to side in the electromagnet's magnetic field.

Einthoven made yet another significant addition to cardiac electrophysiology in 1912, when he discovered a mathematical link between the direction and size of the deflections recorded by the three limb leads. Einthoven's triangle is the name for this hypothesis. Before Frank Wilson described unipolar leads and the precordial lead configuration, the typical three-limb leads were used for three decades. The traditional Einthoven

limb leads, as well as the precordial and unipolar limb leads based on Wilson's work, make up the 12-lead ECG layout now in use.

This instrument was initially manufactured in 1905 by the Cambridge Scientific Instrument Company in London. Electrical impulses were sent from a hospital over a mile away to Einthoven's laboratory via a telephone cable. Bedside machines, on the other hand, were not available until the 1920s. The Sanborn Company produced a smaller version of the unit in 1935 that weighed only about 25 pounds.



*Figure 1.4 – Holter-Edan ECG device*

With Norman Jeff Holter's invention of the Holter monitor in 1949, the use of ECG in a nonclinical context became viable. The first iteration of this device was a 75-pound backpack that could record the ECG continually and send the signals via radio. The size of subsequent iterations of such devices has been drastically decreased, and the signal is now recorded digitally. Miniaturized devices now allow patients to be monitored for longer periods (typically 24 hours) to aid in the diagnosis of any rhythm or ischemic heart

disease concerns. One of the latest versions of the ECG is the one appearing in Figure 1.4.

### 1.3 The ECG Waveform

Signals of voltage versus time are created during the recording of an ECG, which are generally shown in millivolts (mV) vs seconds. Figure 1.5 depicts a typical Lead II ECG waveform. The negative electrode was placed on the right wrist and the positive electrode on the left ankle for this Lead II ECG recording. As a result, a series of peaks and waves can be seen, each of which corresponds to ventricular or atrial depolarization and repolarization, with each segment of the signal indicating a separate event in the cardiac cycle.

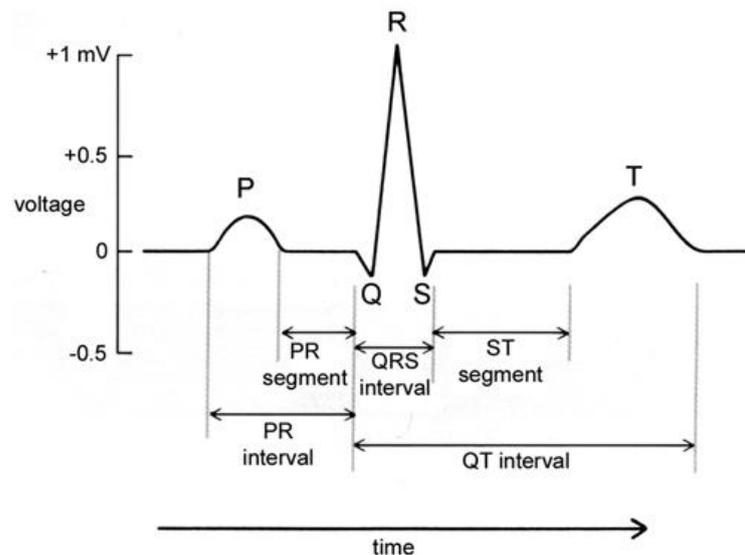


Figure 1.5 – A typical ECG waveform for one cardiac cycle, measured from Lead II position [2].

Three principal waveforms are recorded by the ECG (Figure 1.5):

- The P-wave.
- QRS complex.
- T-wave.

The P-wave is created by depolarization of the atria, the QRS by depolarization of the ventricles, and the T-wave by repolarization of the ventricles. In most people, these waveforms occur in a repeating rhythm called sinus rhythm, so-called because it originates in the sinus node. In some people, a fourth waveform (not shown in the previous image) called a U-wave can be seen. This is usually seen at slower heart rates. The significance of the U-wave remains uncertain. Some authors think that it represents the late stages of ventricular repolarization, while others describe it as a post-repolarization phenomenon. U-wave abnormalities have been described in various disease states including ischemic heart disease [3].

The depolarization of the sinoatrial node, which is positioned within the right atrium, starts the typical cardiac cycle. A conventional ECG will not detect this early firing because the node does not have enough cells to provide a measurable electrical potential. The right and left ventricles continue to depolarize after the P wave, resulting in the recordable QRS complex, which lasts about 100 milliseconds. The Q-wave is the initial negative deflection (if present), the R-wave is the largest positive deflection, and the S-wave is the smallest positive deflection [4].

The T-wave is usually the last potential in a cardiac cycle, followed by the P-wave of the next cycle, and so on. The ECG signal returns to baseline near the conclusion of ventricular contraction, and the ventricles repolarize after contraction. Atrial contractions have stopped, and the atria are repolarizing at the same time as the QRS complex. Because the effects of this widespread atrial repolarization are obscured by the much larger volume of tissue engaged in ventricular depolarization, it is not generally detectable in an ECG [2].

## 1.4 The 12-leads ECG

An ECG lead is a recording of the heart's electrical activity as seen from one side. As a result, when we take a 12-lead ECG, we're recording cardiac electrical activity from 12 different angles [4]. Assume you're visiting a historic structure and taking images of it. If you snap 12 photos from different angles around the structure, each one will depict a distinct element, such as the front, sides, and back. They work together to provide a three-dimensional record of the structure's shape and appearance. Similarly, a 12-lead ECG creates a three-dimensional depiction of the heart's electrical activity.

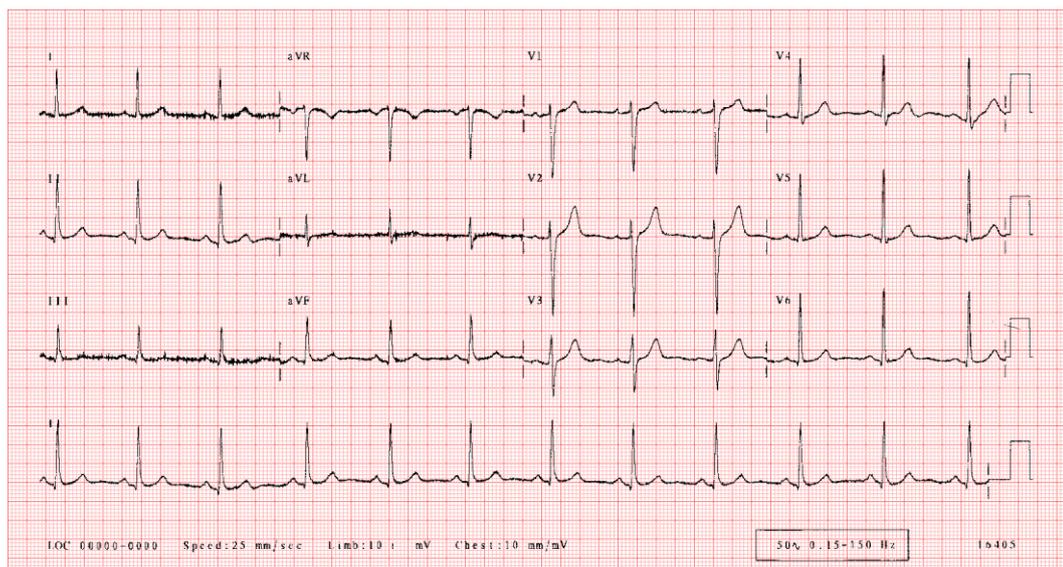


Figure 1.6 - 12-lead normal ECG

Multiple images of the heart's electrical activity can be recorded depending on the type of machine utilized and the number of electrodes inserted. The usage of 12-lead ECG devices is common among healthcare professionals. Twelve separate electrical images of the heart are measured and recorded via a 12-lead ECG (Figure 1.6). In other words, it records the electrical

activity of the heart as observed from 12 various angles. For example, Lead II monitors electrical activity as observed from the heart's inferior (diaphragmatic) surface. This lead is frequently used to measure heart rate [5].

## **1.5 Arrhythmia types**

Analyzing arrhythmias is a difficult undertaking since every person on the planet has a unique ECG that differs from everyone else's, and one person's ECG can change dramatically from one second to the next. Memorizing some of the most common ECG patterns and attempting to recognize them in the future is insufficient. Pattern identification is a popular yet unintentional way of approaching arrhythmias via ECG analysis [6]. Often the Arrhythmias are divided into two global categories:

1. Rhythmic: determined as a sequence of uneven beats.
2. Morphological: made of an abnormal single beat.

The work presented in the current thesis is focused on the first type of classification. Those arrhythmias have a categorization provided by SNOMED CT. The latter is the world's most complete and precise terminology package, with widespread acceptance around the world. It provides a common language for clinical IT systems, making data exchange between them easier, safer, and more accurate. It covers everything from processes and symptoms to clinical measurements, diagnosis, and drugs, and it's all in one place.

Arrhythmias are often divided into groups based on where the rhythm is initiated by the pacemaker. The following are the most prevalent sites, and consequently the primary arrhythmia categories:

1. Sinus
2. Atrial
3. Junctional
4. Ventricular
5. AV Blocks

### 1.5.1 Sinus

It is necessary to comprehend the 'benchmark' rhythm, or hemodynamically perfect rhythm, which is referred to as **Normal Sinus Rhythm (NSR)**, in order to assess cardiac rhythms (Figure 1.7). The following features must be present for a rhythm to be classified as Normal Sinus Rhythm.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular   |
| Rate           | 60-100/minute                                       |
| p waves        | Present, upright, symmetrical, one before every QRS |
| pri            | .12-.20 seconds                                     |
| QRS            | .06-.10 seconds                                     |

*Table 1.1 - Characteristics of Normal Sinus Rhythm*



*Figure 1.7 - Normal Sinus Rhythm [6]*

**Sinus Bradycardia (SB):** when a patient’s heart rate falls below 60 beats per minute, they are said to be bradycardic. Slow heart rates can be seen in fit and active people who are usually asymptomatic. When a patient’s heart rate falls below 60 beats per minute, critical care nurses must be ready to assess for decreasing cardiac output right away.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular   |
| Rate           | < 60/minute   |
| p waves        | Present, upright, symmetrical, one before every QRS |
| pri            | .12-.20 seconds                                     |
| QRS            | .06-.10 seconds                                     |

*Table 1.2 - Characteristics of Sinus Bradycardia*



*Figure 1.8 - Sinus Bradycardia [6]*

**Sinus Tachycardia (STach):** when a patient’s heart rate exceeds 100 beats per minute, they are labeled tachycardic, though most people don’t notice symptoms until their heart rate exceeds 150 beats per minute. At this point, a critical care nurse should look for signs and symptoms of decreased cardiac output (such as hypotension or a loss of consciousness).

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular   |
| Rate           | > 100/minute  |
| p waves        | Present, upright, symmetrical, one before every QRS |
| pri            | .12-.20 seconds                                     |
| QRS            | .06-.10 seconds                                     |

Table 1.3 - Characteristics of Sinus Tachycardia



Figure 1.9 - Sinus Tachycardia [6]

**Sinus Arrhythmia (SA):** this arrhythmia is typically benign and do not require any sort of treatment. It is seen in children and in mechanically ventilated patients.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular   |
| Rate           | 60-100/minute                                       |
| p waves        | Present, upright, symmetrical, one before every QRS |
| pri            | .12-.20 seconds                                     |
| QRS            | .06-.10 seconds                                     |

Table 1.4 - Characteristics of Sinus Arrhythmia



Figure 1.10 - Sinus Arrhythmia [6]

**Wandering Atrial Pacemaker (WAP):** It can be a normal aberration associated to ischemia. There is no treatment required.

| Characteristic | Status                         |
|----------------|--------------------------------|
| Rhythm         | Regular                        |
| Rate           | 60-100/minute                  |
| p waves        | P waves vary in shape and size |
| pri            | .12-.20 seconds                |
| QRS            | .06-.10 seconds                |

Table 1.5 - Characteristics of Wandering Atrial Pacemaker



Figure 1.11 - Wandering Atrial Pacemaker [6]

## 1.5.2 Atrial

The rhythms that originate in the atrial will be examined in the following section. Premature atrial contractions, atrial flutter, atrial fibrillation, and supraventricular tachycardia are examples of these arrhythmias. The key characteristics of cardiac rhythms will be outlined, as well as nursing consequences and useful advice to help critical care nurses correctly interpret atrial arrhythmias.

**Premature Atrial Contractions (PAC):** it can be a normal aberration, ischemia, or a signal of atrial irritability. It can lead to more serious atrial rhythms.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Early beat (PAC) causes rhythm to be irregular              |
| Rate           | Underlying rhythm usually 60-100/minute                     |
| p waves        | P waves have different configuration than underlying rhythm |
| pri            | .12-.20 seconds in underlying rhythm                        |
| QRS            | .06-.10 seconds in underlying rhythm                        |

*Table 1.6 - Characteristics of Premature Atrial Contractions*

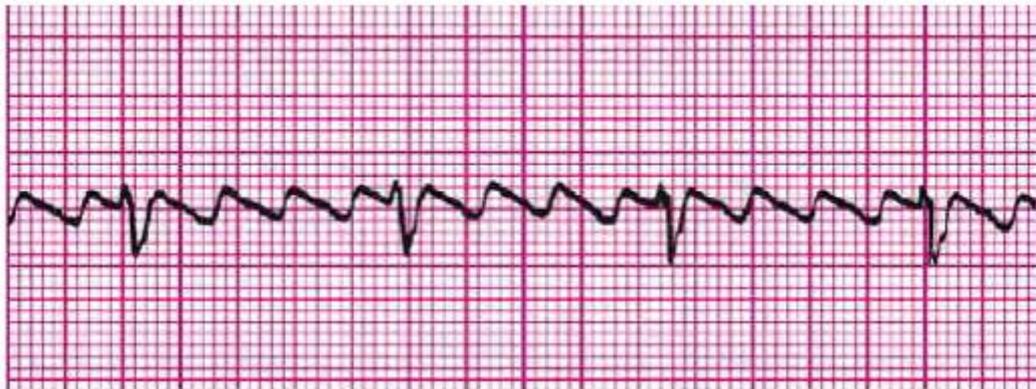


*Figure 1.12 - Premature Atrial Contractions [6]*

**Atrial Flutter (AFL):** it is caused by electrolyte imbalance, hypertension, ischemic heart disease, congenital heart disease, and rheumatic valve disease. Also, could appear after a cardiac surgery.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular or irregular  |
| Rate           | 60-100/minute (ventricular rate) 250-400 (atrial rate)          |
| p waves        | No p waves present. Flutter waves (F waves) or 'sawtooth' waves |
| pri            | No pri since no p waves   |
| QRS            | .06-.10 seconds   |

*Table 1.7 - Characteristics of Atrial Flutter*



*Figure 1.13 - Atrial Flutter [6]*

**Atrial Fibrillation (AF):** it is caused by electrolyte imbalance, hypertension, ischemic heart disease, congenital heart disease, rheumatic valve disease. Also following a cardiac surgery.

| Characteristic | Status    |
|----------------|-----------|
| Rhythm         | Irregular |

|         |  |
|---------|--|
| Rate    | 60-100/minute (ventricular rate) >400/minute (atrial rate) |
| p waves | No p waves. Fibrillatory waves (f waves)                   |
| pri     | No pri since no p waves                                    |
| QRS     | .06-.10 seconds  |

Table 1.8 - Characteristics of Atrial Fibrillation

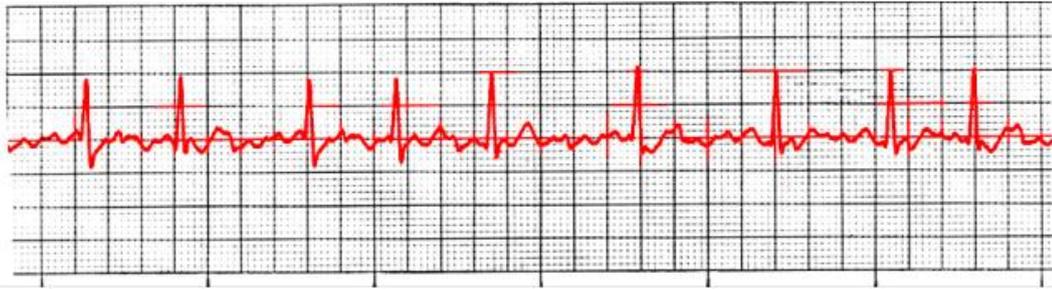


Figure 1.14 - Atrial Fibrillation [6]

**Supraventricular Tachycardia (SVT):** it is caused by congenital heart disease, emotional stress, physical stress or exertion, illegal drugs (i.e. cocaine or ecstasy), alcohol, caffeine.

| Characteristic | Status                                  |
|----------------|---|
| Rhythm         | Regular                                 |
| Rate           | 150-250/minute (atrial rate)            |
| p waves        | P waves may not be seen at higher rates |
| pri            | .12-.20 seconds (if seen)               |
| QRS            | .06-.10 seconds                         |

Table 1.9 - Characteristics of Supraventricular Tachycardia

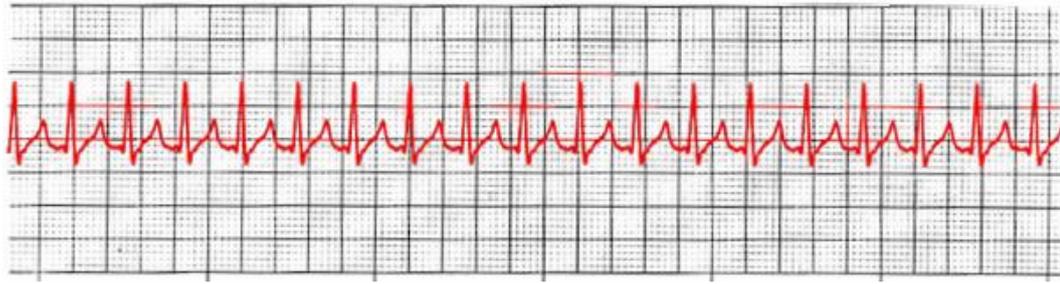


Figure 1.15 - Supraventricular Tachycardia [6]

### 1.5.3 Junctional

Junctional rhythms are temporary and non-lethal rhythms that originate in the AV node or junctional region. Inverted p waves are a typical feature of all junctional rhythms.

**Premature Junctional Contraction (JPC):** it is caused by medication toxicity (i.e. digoxin), ischemia. There is not treatment required. Continue to observe for increasing number of JPCs since this indicates increasing AV node irritability.

| Characteristic | Status   |
|----------------|--|
| Rhythm         | Early beat (PJC) causes the rhythm to be irregular |
| Rate           | 60-100/minute (underlying rhythm)                  |
| p waves        | P waves inverted or not seen in JPC                |
| pri            | Not applicable                                     |
| QRS            | .06-.10 seconds (in underlying rhythm)             |

Table 1.10 - Characteristics of Premature Junctional Contraction



Figure 1.16 - Premature Junctional Contraction [6]

**Junctional Rhythm (AVJR):** it is caused by medication toxicity (i.e. digoxin) or ischemia. It is necessary to treat causes.

| Characteristic | Status                     |
|----------------|----------------------------|
| Rhythm         | Regular                    |
| Rate           | <60/minute                 |
| p waves        | P waves inverted or absent |
| pri            | .12-.20 seconds            |
| QRS            | .06-.10 seconds            |

Table 1.11 - Characteristics of Junctional Rhythm

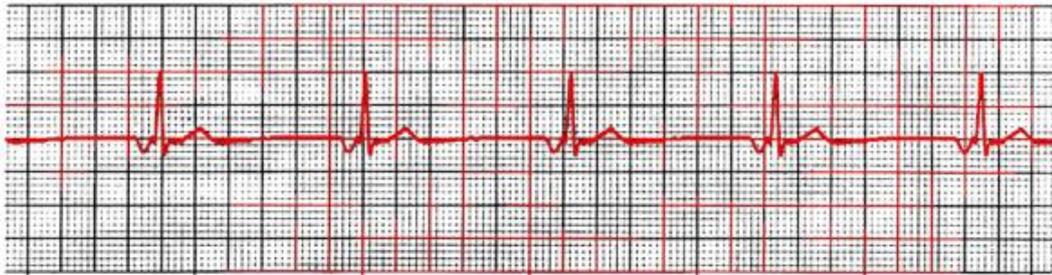


Figure 1.17 - Junctional Rhythm [6]

**Accelerated Junctional Rhythm (AJR):** it is caused by medication toxicity (i.e. digoxin) or ischemia. It is necessary to treat causes.

| Characteristic | Status  |
|----------------|---------|
| Rhythm         | Regular |

|         |                            |
|---------|----------------------------|
| Rate    | 60-100/minute              |
| p waves | P waves inverted or absent |
| pri     | Not applicable             |
| QRS     | .06-.10 seconds            |

*Table 1.12 - Characteristics of Accelerated Junctional Rhythm*



*Figure 1.18 - Accelerated Junctional Rhythm [6]*

**Paroxysmal Junctional Tachycardia (JT):** It is caused by ischemia. Its treatment is the same than SVT.

| <b>Characteristic</b> | <b>Status</b>                        |
|-----------------------|--------------------------------------|
| Rhythm                | Regular                              |
| Rate                  | 150-250/minute                       |
| p waves               | P waves inverted or absent (if seen) |
| pri                   | Not applicable                       |
| QRS                   | .06-.10 seconds                      |

*Table 1.13 - Characteristics of Paroxysmal Junctional Tachycardia*



Figure 1.19 - Paroxysmal Junctional Tachycardia [6]

### 1.5.4 Ventricular

Premature ventricular contractions, ventricular tachycardia, and ventricular fibrillation are examples of ventricular rhythms that can be induced by irritability, as well as those that result from the failure of higher-level pacemakers. Irritability patients have significantly various treatment options and consequences.

**Premature Ventricular Contractions (PVC):** it is caused by ventricular irritability (i.e. hypoxemia, acid-base imbalance, medications, electrolyte imbalance).

| Characteristic | Status   |
|----------------|--|
| Rhythm         | Early beat (PVC) causes the rhythm to be irregular |
| Rate           | 60-100/minute (underlying rhythm)                  |
| p waves        | None (in PVC)                                      |
| pri            | None (in PVC)                                      |
| QRS            | > .12 seconds (wide and bizarre)                   |

Table 1.14 - Characteristics of Premature Ventricular Contractions

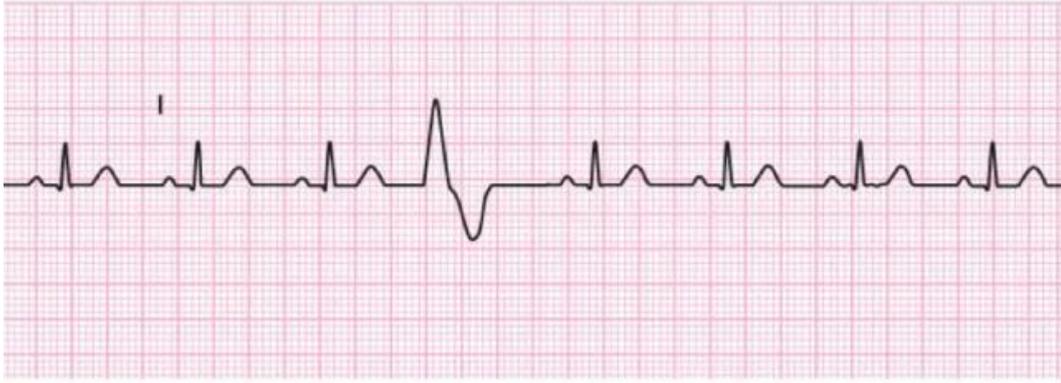


Figure 1.20 - Premature Ventricular Contractions [6]

**Ventricular Tachycardia (VTach):** it is caused by ventricular irritability (i.e. hypoxemia, acid-base imbalance, medications, electrolyte imbalance).

| Characteristic | Status                           |
|----------------|----------------------------------|
| Rhythm         | Regular                          |
| Rate           | 150-250/minute                   |
| p waves        | None                             |
| pri            | None                             |
| QRS            | > .12 seconds (wide and bizzare) |

Table 1.15 - Characteristics of Ventricular Tachycardia

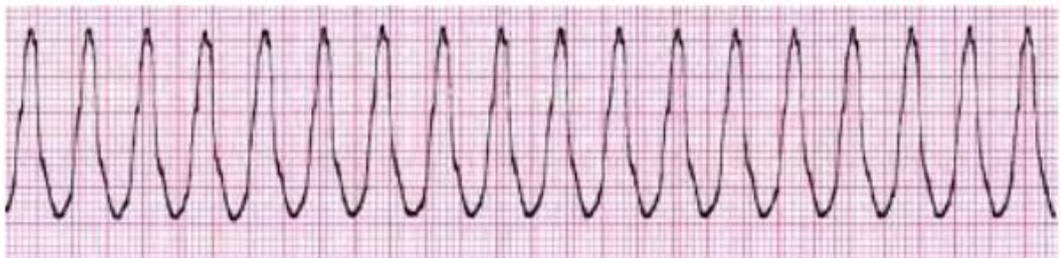


Figure 1.21 - Premature Ventricular Tachycardia [6]

**Ventricular Fibrillation (VF):** it is caused by ventricular irritability (i.e. hypoxemia, acid-base imbalance, medications, electrolyte imbalance).

| Characteristic | Status                |
|----------------|-----------------------|
| Rhythm         | Irregular and chaotic |
| Rate           | Cannot calculate      |
| p waves        | None                  |
| pri            | None                  |
| QRS            | None                  |

Table 1.16 - Characteristics of Idioventricular Rhythm



Figure 1.22 - Ventricular Fibrillation [6]

**Idioventricular Rhythm (IR):** it is caused by ischemia, reperfusion post thrombolytics.

| Characteristic | Status                           |
|----------------|----------------------------------|
| Rhythm         | Regular                          |
| Rate           | <40/minute                       |
| p waves        | No p waves                       |
| pri            | No pri                           |
| QRS            | > .12 seconds (wide and bizarre) |

Table 1.17 - Characteristics of Idioventricular Rhythm

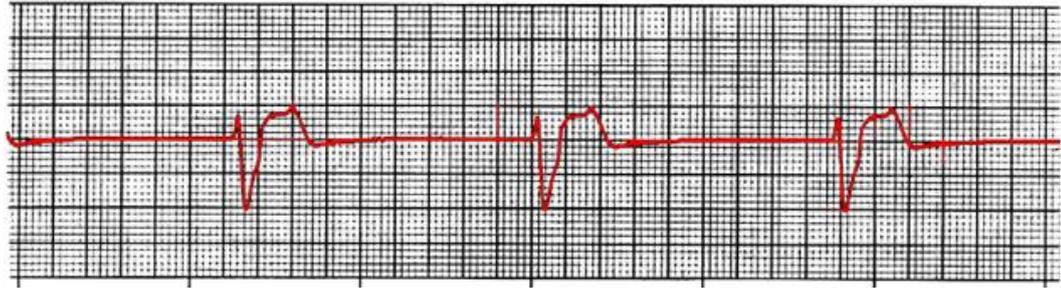


Figure 1.23 - Idioventricular Rhythm [6]

### 1.5.5 AV Blocks

Electrical conduction failure via the myocardium is characterized by atrioventricular (AV) blockages. Because AV blockages are linked to severe risk worsening or hemodynamic impairment, the critical care nurse must recognize and treat them as soon as possible.

**First Degree AV Block (IAVB):** it is caused by AV nodal disease, enhanced vagal tone (i.e. athletes), myocarditis, following myocardial infarction, electrolyte disturbances, medications (i.e. calcium channel blockers, beta blockers).

| Characteristic | Status          |
|----------------|-----------------|
| Rhythm         | Regular         |
| Rate           | 60-100/minute   |
| p waves        | P waves normal  |
| pri            | >.20 seconds    |
| QRS            | .06-.10 seconds |

Table 1.18 - Characteristics of First Degree AV Block

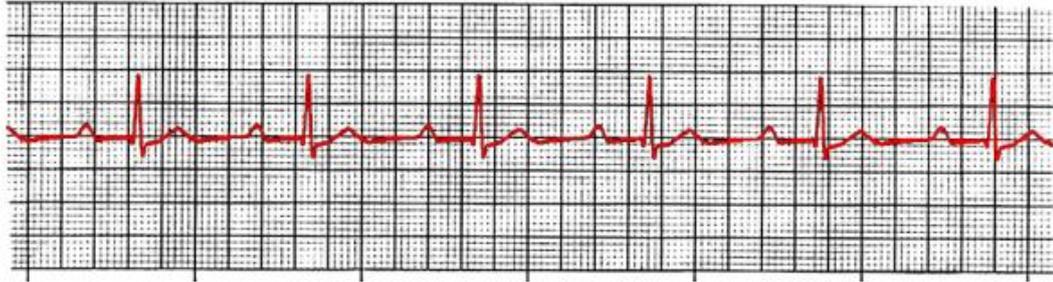


Figure 1.24 - First Degree AV Block [6]

**Second Degree Type I (IIAVB):** it is caused by ischemia. Usually benign, with no treatment required. If patient becomes hemodynamically compromised interventions for bradycardia should be considered.

| Characteristic | Status  |
|----------------|---|
| Rhythm         | Regular or slightly irregular                     |
| Rate           | 60-100/minute                                     |
| p waves        | P waves normal                                    |
| pri            | Progressively gets longer until a beat is dropped |
| QRS            | .06-.10 seconds                                   |

Table 1.19 - Characteristics of Second Degree Type I

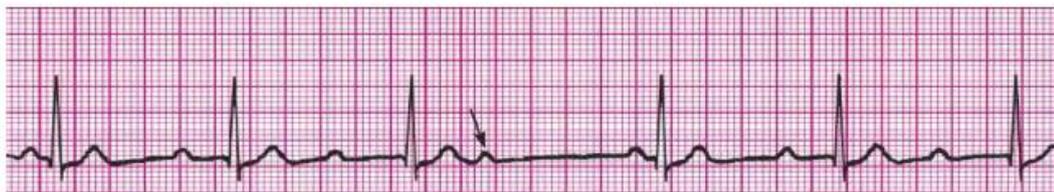


Figure 1.25 - Second Degree Type 1 [6]

## 2. Datasets

The advent of Artificial Intelligence (AI) based solutions to a wide range of real-world problems including industry, healthcare, security, etc. has turned the spotlight on the problem of data quality. “High-quality datasets are essential for developing machine learning models” according to an article in the International Journal on Advances in Software [7], and, in fact, data plays a critical role in ensuring that good ML models generate good quality predictions.

The following are some of the most relevant data quality components:

- **Accuracy:** accuracy is the degree to which data correctly reflects the real-world object or scenarios.
- **Completeness:** completeness denotes the percent of all required data currently available in a dataset.
- **Consistency:** data consistency refers to the uniformity of data. Consistency is achieved when data values do not conflict with other values within a record or across different data sets.
- **Validity:** data should be conformed to a specific format or to defined business rules and parameters.
- **Uniqueness:** uniqueness ensures that there are no duplications or overlapping of values across all data sets.
- **Timeliness:** timeliness refers to the time expectation for accessibility and availability of information. Timeliness can be measured as the time between when information is expected and when it is readily available for use. Timely data is data that is available when it is required.

Critical decisions based on poor-quality data can have very serious consequences. Overestimation of data quality implies underestimation of the implications of poor-quality data. Poor decision-making, business inefficiencies, reputational harm, missed opportunities and lost revenue are all serious repercussions for a company that does not prioritize data quality [8]. To give an idea, in 2017 the Data Warehousing Institute (TDWI) [9] estimates that poor data quality costs businesses in the United States more than \$700 billion per year.

The MIT-BIH Arrhythmia Database is the most commonly used and accredited dataset in the field of cardiologic diseases and ECG signals, with a high level of data quality assurance. The AHA Database for Evaluation of Ventricular Arrhythmia Detectors and the European ST-T databases, as well as the aforementioned MIT-BIH Arrhythmia database, have long been the most widely used ECG databases. The PhysioNet [10] team recently shared a new dataset for the 2020 Challenge that is gaining ground in the literature.

## **2.1 MIT-BIH Arrhythmia Database**

The Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) Arrhythmia database was the first generally available set of standard test material for evaluating arrhythmia detectors, and it has been utilized for that purpose as well as for basic research into cardiac dynamics at approximately 500 sites worldwide since 1980 [11].

The ECGs in the database come from a collection of over 4000 long-term Holter recordings [12] taken between 1975 and 1979 by the Beth Israel Hospital Arrhythmia Laboratory (BIH; now the Beth Israel Deaconess Medical Center). Approximately 60% of the subjects were inpatients. It

includes 48 half-hour recordings from 25 men aged 32 to 89 years and 22 women aged 23 to 89 years (records 201 and 202 came from the same male subject). The first 23 records are chosen at random (but with sufficient quality for human experts to analyze) from the entire collection, while the additional 25 are selected from the same set to include a variety of rare but clinically important phenomena such as complex ventricular, junctional, and supraventricular arrhythmias and conduction abnormalities, that would not be well-represented by a small random sample. The ECG leads differed from subject to subject, as would be expected in clinical practice, because surgical dressings and anatomical variations do not allow for the use of the same electrode placement in all cases. However, in most records, the upper signal is a modified limb lead II (MLII), obtained by placing the electrodes on the chest, and the lower signal is a modified lead V1 (occasionally V2, V4 or V5), which is also obtained by placing the electrodes on the chest. The digitization rate is of 360 samples per second per channel. Two or more cardiologists independently annotated each record for a total of 110,000 annotations included with the database. The codes and descriptions of the annotations contained in the database are shown in Table 2.1, Table 2.2, and Table 2.3, while Table 2.4 and Table 2.5 show the heartbeat and rhythm distributions, respectively.

| <b>Code</b> | <b>Description</b>   |
|-------------|--|
| N           | Normal beat (displayed as "." by the PhysioBank ATM, LightWAVE, pschart, and psfd) |
| L           | Left bundle branch block beat  |
| R           | Right bundle branch block beat   |
| B           | Bundle branch block beat (unspecified)   |
| A           | Atrial premature beat  |

|   |  |
|---|--|
| a | Aberrated atrial premature beat                              |
| J | Nodal (junctional) premature beat                            |
| S | Supraventricular premature or ectopic beat (atrial or nodal) |
| V | Premature ventricular contraction                            |
| r | R-on-T premature ventricular contraction                     |
| F | Fusion of ventricular and normal beat                        |
| e | Atrial escape beat   |
| j | Nodal (junctional) escape beat                               |
| n | Supraventricular escape beat (atrial or nodal)               |
| E | Ventricular escape beat                                      |
| / | Paced beat   |
| f | Fusion of paced and normal beat                              |
| Q | Unclassifiable beat  |
| ? | Beat not classified during learning                          |

Table 2.1 - MIT-BIH beat annotations.

| Code | Description                               |
|------|---|
| [    | Start of ventricular flutter/fibrillation |
| !    | Ventricular flutter wave                  |
| ]    | End of ventricular flutter/fibrillation   |
| x    | Non-conducted P-wave (blocked APC)        |
| (    | Waveform onset                            |
| )    | Waveform end                              |
| p    | Peak of P-wave                            |
| t    | Peak of T-wave                            |
| u    | Peak of U-wave                            |
| `    | PQ junction                               |
| '    | J-point                                   |
| ^    | (Non-captured) pacemaker artifact         |

|   |                            |
|---|----------------------------|
|   | Isolated QRS-like artifact |
| ~ | Change in signal quality   |
| + | Rhythm change              |
| s | ST segment change          |
| T | T-wave change              |
| * | Systole                    |
| D | Diastole                   |
| = | Measurement annotation     |
| " | Comment annotation         |
| @ | Link to external data      |

Table 2.2 - MIT-BIH non-beat annotations.

| Code  | Description                      |
|-------|----------------------------------|
| (AB   | Atrial bigeminy                  |
| (AFIB | Atrial fibrillation              |
| (AFL  | Atrial flutter                   |
| (B    | Ventricular bigeminy             |
| (BII  | 2° heart block                   |
| (IVR  | Idioventricular rhythm           |
| (N    | Normal sinus rhythm              |
| (NOD  | Nodal (A-V junctional) rhythm    |
| (P    | Paced rhythm                     |
| (PREX | Pre-excitation (WPW)             |
| (SBR  | Sinus bradycardia                |
| (SVTA | Supraventricular tachyarrhythmia |
| (T    | Ventricular trigeminy            |
| (VFL  | Ventricular flutter              |
| (VT   | Ventricular tachycardia          |

Table 2.3 - MIT-BIH most commonly used rhythm annotation strings.

| <b>Class</b> | <b>Number of instances</b> |
|--------------|----------------------------|
| N            | 90585                      |
| S            | 2781                       |
| V            | 7235                       |
| F            | 802                        |

*Table 2.4 - MIT-BIH distributions of heartbeats according to the AAMI recommendations.*

| <b>Class</b>                          | <b>Number of observations</b> |
|---------------------------------------|-------------------------------|
| Normal sinus rhythm                   | 283                           |
| Atrial premature beat                 | 66                            |
| Atrial flutter                        | 20                            |
| Atrial fibrillation                   | 135                           |
| Supraventricular tachyarrhythmia      | 13                            |
| Pre-excitation (WPW)                  | 21                            |
| Premature ventricular contraction     | 133                           |
| Ventricular bigeminy                  | 55                            |
| Ventricular trigemini                 | 13                            |
| Ventricular tachycardia               | 10                            |
| Idioventricular rhythm                | 10                            |
| Ventricular flutter                   | 10                            |
| Fusion of ventricular and normal beat | 11                            |
| Left bundle branch block beat         | 103                           |
| Right bundle branch block beat        | 62                            |
| Second-degree heart block             | 10                            |
| Pacemaker rhythm                      | 45                            |

*Table 2.5 - MIT-BIH distributions of rhythms.*

## 2.2 AHA Database for Evaluation of Ventricular Arrhythmia Detectors

The American Heart Association (AHA) sponsored the development of the AHA Database for Evaluation of Ventricular Arrhythmia Detectors with funding from the National Heart, Lung, and Blood Institute (NHLBI) [13]. A group led by G. Charles Oliver of Washington University in St. Louis created the database between 1977 and 1985.

This database has many features in common with the MIT-BIH Arrhythmia Database [11]. Particularly, both databases comprise two-channel Holter recordings, each having 30 minutes of signals meticulously hand-annotated beat-by-beat.

Recordings included in the AHA Database were chosen to satisfy one of eight sets of stringently defined selection criteria based on the degree of ventricular ectopy. The sets are:

1. no ventricular ectopy
2. isolated unifocal PVCs
3. isolated multifocal PVCs
4. ventricular bi- and trigeminy
5. R-on-T PVCs
6. ventricular couplets
7. ventricular tachycardia
8. ventricular flutter/fibrillation

For each of the eight sets, twenty recordings from actual patients were picked. Each of them was divided into two equal parts, one for algorithm development and the other for performance evaluation. Both the “development” and “evaluation” databases consisted of 80 two-channel

extracts of analog ambulatory ECG recordings, which were digitized at 250 Hz per channel with 12-bit resolution over a 10 mV range. Two versions of the database are available: the short version includes five minutes of unannotated ECG signals prior to the 30 minutes annotated segment of each recording, and the long version includes 2.5 hours of unannotated ECG signals prior to each annotated segment [14] [10].

The first portions of the “development” set were released in 1982, and all 80 records have been distributed since 1985 by ECRI. No revisions or updates were made subsequently. In 2000, ECRI made the “evaluation” recordings available for the first time.

As a result, although the AHA Database has an excellent representation of the most severe types of ventricular ectopy, it has relatively few examples of supraventricular ectopy, conduction defects, and noise-contaminated waveforms, all of which are common in clinical practice.

### **2.3 European ST-T database**

In 1986, a group led by Carlo Marchesi at the Institute of Clinical Physiology of the National Research Council (CNR) in Pisa decided to take on the challenge of creating a database for the development and evaluation of abnormalities in the ST segment and the T wave indicative of myocardial ischemia [11]. In the years following the creation of the MIT-BIH and AHA Databases, improvements in ambulatory ECG recorders permitted accurate reproduction of components of the ECG in the 0.01-0.10 Hz frequency range, which was required to observe these changes. The initiation of a standard approach to the detection and interpretation of ST-T changes was made by the “Concerted Action” on Ambulatory Monitoring, set up by the European Community in 1985. The development of the full database was

coordinated by the CNR Institute and the Thoraxcenter of the Erasmus University in Rotterdam. In 1989, the European Society of Cardiology agreed to sponsor the remainder of the project, providing both financial and scientific backing so as to enable the completion of the database. Finally, with the cooperation of the Biomedical Engineering Centre of MIT, production of the first edition of the European ST-T database on CD-ROM went ahead [15]. The source of the ECGs included in the European ST-T Database is a set of 24-h Holter recordings which were provided by the participating research groups from eight countries. This database consists of 90 continuous two-channel records, each 2 h in duration, taken from ambulatory ECG recordings from 79 patients: 70 men aged 30 to 84, and 8 women aged 55 to 71 (information is missing for one patient and some records come from the same patient). Each record contains at least one ST or T episode. 372 ST segment change instances and 423 T-wave change episodes are included in the database, with durations ranging from 30 seconds to several minutes and peak displacements ranging from 100 mV to more than 1 mV.

For each case, the two leads were recorded. The leads which were used included modified leads V1, V2, V3, V4, and V5, and modified limb leads I and III (MLI, MLIII), obtained by placing the electrodes on the chest.

In cooperation with the developers of MIT-BIH Arrhythmia Database, the annotation scheme was revised to be consistent with both MIT-BIH and AHA formats [16]. The codes and descriptions of the annotations contained in the database are shown in Table 2.6, Table 2.7, and Table 2.8, while Table 2.9, Table 2.10, and Table 2.11 Table 2.5 show the total counts of main events, heartbeat distribution, and rhythm distribution, respectively.

| <b>Code</b> | <b>Description</b>                         |
|-------------|--|
| N           | Normal beat                                |
| a           | Aberrated atrial premature beat            |
| J           | Nodal (junctional) premature beat          |
| S           | Supraventricular premature or ectopic beat |
| V           | Premature ventricular contraction          |
| F           | Fusion of ventricular and normal beat      |
| Q           | Unclassifiable beat                        |
| I           | Isolated QRS-like artifact                 |

Table 2.6 - ST-T database beat annotations.

| <b>Code</b> | <b>Description</b>               |
|-------------|----------------------------------|
| (SB         | Supra ventricular bigeminy       |
| (B          | Ventricular bigeminy             |
| (B3         | Third degree heart block         |
| (N          | Normal sinus rhythm              |
| (SBR        | Sinus bradycardia                |
| (SVTA       | Supraventricular tachyarrhythmia |
| (T          | Ventricular trigeminy            |
| (VT         | Ventricular tachycardia          |

Table 2.7 - ST-T database rhythm annotations.

| <b>Code</b> | <b>Description</b>      |
|-------------|-------------------------|
| (ST..       | Beginning of ST episode |
| AST...      | Peak of ST episode      |
| ST..)       | End of ST episode       |
| (T...       | Beginning of T episode  |
| AT...       | Peak of T episode       |
| T..)        | End of T episode        |

Table 2.8 - ST-T database ST and T change annotations.

|                         |        |                 |                  |          |
|-------------------------|--------|-----------------|------------------|----------|
| Beats                   | 790565 |                 |                  |          |
| SVPC                    | 1093   |                 |                  |          |
| PVC                     | 4467   |                 |                  |          |
| Fusion PVC              | 354    |                 |                  |          |
| Supraventricular ectopy |        |                 |                  |          |
| Isolated beats          | 819    |                 |                  |          |
| Couplets                | 103    |                 |                  |          |
| Runs                    | 22     |                 |                  |          |
| Ventricular ectopy      |        |                 |                  |          |
| Isolated beats          | 3771   |                 |                  |          |
| Couplets                | 122    |                 |                  |          |
| Runs                    | 191    |                 |                  |          |
| ST-T episodes           | No.    | Peak ( $\mu$ V) | Duration (h:m:s) |          |
|                         |        | mean            | mean             | total    |
| (first channel)         |        |                 |                  |          |
| Positive ST deviation   | 65     | + 250           | 2:32             | 2:45:33  |
| Negative ST deviation   | 121    | - 200           | 8:52             | 17:53:40 |
| Positive T deviation    | 141    | + 300           | 5:12             | 12:15:27 |
| Negative T deviation    | 94     | - 300           | 5:02             | 7:54:19  |
| (second channel)        |        |                 |                  |          |
| Positive ST deviation   | 53     | + 300           | 4:04             | 3:35:44  |
| Negative ST deviation   | 133    | - 200           | 9:40             | 21:25:52 |
| Positive T deviation    | 89     | + 350           | 4:21             | 6:28:27  |
| Negative T deviation    | 99     | - 300           | 6:02             | 9:58:42  |

Table 2.9 - ST-T database total counts of main events.

| Class | Number of instances |
|-------|---------------------|
| N     | 784633              |
| S     | 1095                |
| V     | 4467                |
| F     | 354                 |

Table 2.10 - ST-T database distributions of heartbeats according to the AAMI recommendations.

| Class                            | Number of observations |
|----------------------------------|------------------------|
| Ventricular bigeminy             | 37                     |
| Third degree heart block         | 1                      |
| Normal sinus rhythm              | 426                    |
| Sinus bradycardia                | 10                     |
| Supraventricular tachyarrhythmia | 22                     |

|                         |     |
|-------------------------|-----|
| Ventricular trigeminy   | 39  |
| Ventricular tachycardia | 191 |

*Table 2.11 - ST-T database distributions of rhythms.*

## **2.4 PhysioNet**

### **2.4.1 Moody PhysioNet Challenges**

PhysioNet presents an annual series of biomedical 'Challenges' that focus on unsolved clinical and basic science challenges in collaboration with the annual Computing in Cardiology (CinC) conferences. The National Institutes of Health (NIH), Google, MathWorks, and the Gordon and Betty Moore Foundation have all lent their support to these challenges. George Moody, of the Laboratory for Computational Physiology (LCP), directed these Challenges for the first 15 years (from 2000 to 2014), before retiring due to ill health. Gari Clifford of Emory University and the Georgia Institute of Technology has been leading the Challenges since 2015. In 2021, the 'PhysioNet/Computing in Cardiology Challenges' were renamed the 'George B. Moody PhysioNet Challenges' to honor George's lifetime contributions to the discipline, particularly his seminal work on the Challenges [17].

### **2.4.2 2020 PhysioNet Challenge: Classification of 12-lead**

#### **ECGs**

#### **2.4.2.1 The Challenge**

The 2020 Challenge's purpose is to use 12-lead ECG records to detect clinical diagnosis. Starting from the clinical data provided, the participants must

implement an open-source algorithm that can automatically classify the cardiac abnormality or abnormalities present in each 12-lead ECG recording and provide a probability or confidence score for each of them, with an emphasis on 27 diagnoses (Table 2.12) To determine the winner, the trained models of the participants are run on hidden validation and test sets and their performance is evaluated using a novel, expert-based evaluation metric designed specifically for the 2020 Challenge. The team whose algorithm achieves the highest score is the winner of the Challenge.

| <b>Diagnosis</b>                                 | <b>Code</b> | <b>Abbreviation</b> |
|--|-------------|---------------------|
| 1st degree AV block                              | 270492004   | IAVB                |
| Atrial fibrillation                              | 164889003   | AF                  |
| Atrial flutter                                   | 164890007   | AFL                 |
| Bradycardia                                      | 426627000   | Brady               |
| Complete right bundle branch block               | 713427006   | CRBBB               |
| Incomplete right bundle branch block             | 713426002   | IRBBB               |
| Left anterior fascicular block                   | 445118002   | LAnFB               |
| Left axis deviation                              | 39732003    | LAD                 |
| Left bundle branch block                         | 164909002   | LBBB                |
| Low QRS voltages                                 | 251146004   | LQRSV               |
| Nonspecific intraventricular conduction disorder | 698252002   | NSIVCB              |
| Pacing rhythm                                    | 10370003    | PR                  |
| Premature atrial contraction                     | 284470004   | PAC                 |
| Premature ventricular contractions               | 427172004   | PVC                 |
| Prolonged PR interval                            | 164947007   | LPR                 |
| Prolonged QT interval                            | 111975006   | LQT                 |
| Q wave abnormal                                  | 164917005   | QAb                 |
| Right axis deviation                             | 47665007    | RAD                 |

|                                  |           |       |
|----------------------------------|-----------|-------|
| Right bundle branch block        | 59118001  | RBBB  |
| Sinus arrhythmia                 | 427393009 | SA    |
| Sinus bradycardia                | 426177001 | SB    |
| Sinus rhythm                     | 426783006 | NSR   |
| Sinus tachycardia                | 427084000 | STach |
| Supraventricular premature beats | 63593006  | SVPB  |
| T wave abnormal                  | 164934002 | Tab   |
| T wave inversion                 | 59931005  | TInv  |
| Ventricular premature beats      | 17338001  | VPB   |

Table 2.12 – Diagnoses, SNOMED CT codes and abbreviations for the 27 diagnoses that were scored for the Challenge.

### 2.4.2.2 The Dataset

The data are from five different sources:

1. CPSC Database and CPSC-Extra Database
2. INCART Database
3. PTB and PTB-XL Database
4. The Georgia 12-lead ECG Challenge (G12EC) Database
5. Undisclosed Database

These sources are described below and summarized in Table 2.13.

The first source consists of three databases from the China Physiological Signal Challenge 2018 (CPSC2018), which took place in Nanjing, China at the 7th International Conference on Biomedical Engineering and Biotechnology: the original public training dataset (CPSC), an unused dataset (CPSC-Extra), and the test dataset (the hidden CPSC set). The first two were shared as training sets, while the last one was split into validation and test set for the 2020 Challenge. This training set consists of two sets of

6,877 (male: 3,699; female: 3,178) and 3,453 (male: 1,843; female: 1,610) of 12-lead ECG recordings lasting from 6 seconds to 60 seconds. Each recording was sampled at 500 Hz.

The second source is the public dataset from the St. Petersburg Institute of Cardiological Technics (INCART) 12-lead Arrhythmia Database. The dataset was shared as a training set. This database consists of 74 annotated recordings extracted from 32 Holter records. Each record is 30 minutes long and contains 12 standard leads, each sampled at 257 Hz.

The third source from the Physikalisch Technische Bundesanstalt (PTB) includes two public databases which were shared as training sets: the PTB Diagnostic ECG Database and the PTB-XL, a large publicly available electrocardiography dataset. The first PTB database contains 516 records (male: 377, female: 139). Each recording was sampled at 1000 Hz. The PTB-XL contains 21,837 clinical 12-lead ECGs (male: 11,379 and female: 10,458) of 10 second length with a sampling frequency of 500 Hz.

The fourth source is the Georgia 12-lead ECG Challenge (G12EC) Database. This is a new database, representing a large population from the Southeastern United States, and is split between the training, validation, and test sets. The validation and test set comprised the hidden G12EC set. This training set contains 10,344 12-lead ECGs (male: 5,551, female: 4,793) of 10 second length with a sampling frequency of 500 Hz.

The fifth source is a dataset from an undisclosed American institution that is geographically distinct from the other dataset sources. This dataset has never been posted publicly and contains 10,000 ECGs all retained as test data [18].

| Database    | Total Patients | Recordings in Training Set | Recordings in Validation Set | Recordings in Test Set | Total Recordings |
|-------------|----------------|----------------------------|------------------------------|------------------------|------------------|
| CPSC        | 9458           | 10330                      | 1463                         | 1463                   | 13256            |
| INCART      | 32             | 74                         | 0                            | 0                      | 74               |
| PTB         | 19175          | 22353                      | 0                            | 0                      | 22353            |
| G12EC       | 15742          | 10344                      | 5167                         | 5167                   | 20678            |
| Undisclosed | Unknown        | 0                          | 0                            | 10000                  | 10000            |
| Total       | Unknown        | 43101                      | 6630                         | 16630                  | 66361            |

Table 2.13 - Numbers of patients and recordings in the training, validation, and test databases for the Challenge.

Table 2.14 summarizes the Challenge databases' age, sex, and recording information, revealing variances amongst the populations, whereas Figure 2.1 summarizes the diagnoses for the training and validation data.

| Dataset                    | Number of Recordings | Mean Duration (seconds) | Mean Age (years) | Sex (male/female) | Sample Frequency (Hz) |
|----------------------------|----------------------|-------------------------|------------------|-------------------|-----------------------|
| CPSC (all data)            | 13256                | 16.2                    | 61.1             | 53%/47%           | 500                   |
| <i>CPSC Training</i>       | 6877                 | 15.9                    | 60.2             | 54%/46%           | 500                   |
| <i>CPSC-Extra Training</i> | 3453                 | 15.9                    | 63.7             | 53%/46%           | 500                   |
| <i>Hidden CPSC</i>         | 2926                 | 17.4                    | 60.4             | 52%/48%           | 500                   |
| INCART                     | 72                   | 1800.0                  | 56.0             | 54%/46%           | 257                   |
| PTB                        | 516                  | 110.8                   | 56.3             | 73%/27%           | 1000                  |
| PTB-XL                     | 21837                | 10.0                    | 59.8             | 52%/48%           | 500                   |
| G12EC (all data)           | 20678                | 10.0                    | 60.5             | 54%/46%           | 500                   |
| <i>G12EC Training</i>      | 10344                | 10.0                    | 60.5             | 54%/46%           | 500                   |
| <i>Hidden G12EC</i>        | 10344                | 10.0                    | 60.5             | 54%/46%           | 500                   |
| Undisclosed                | 10000                | 10.0                    | 63.0             | 53%/47%           | 300                   |

Table 2.14 - Number of recordings, mean duration of recordings, mean age of patients in recordings, sex of patients in recordings, and sample frequency of recordings for each data set.

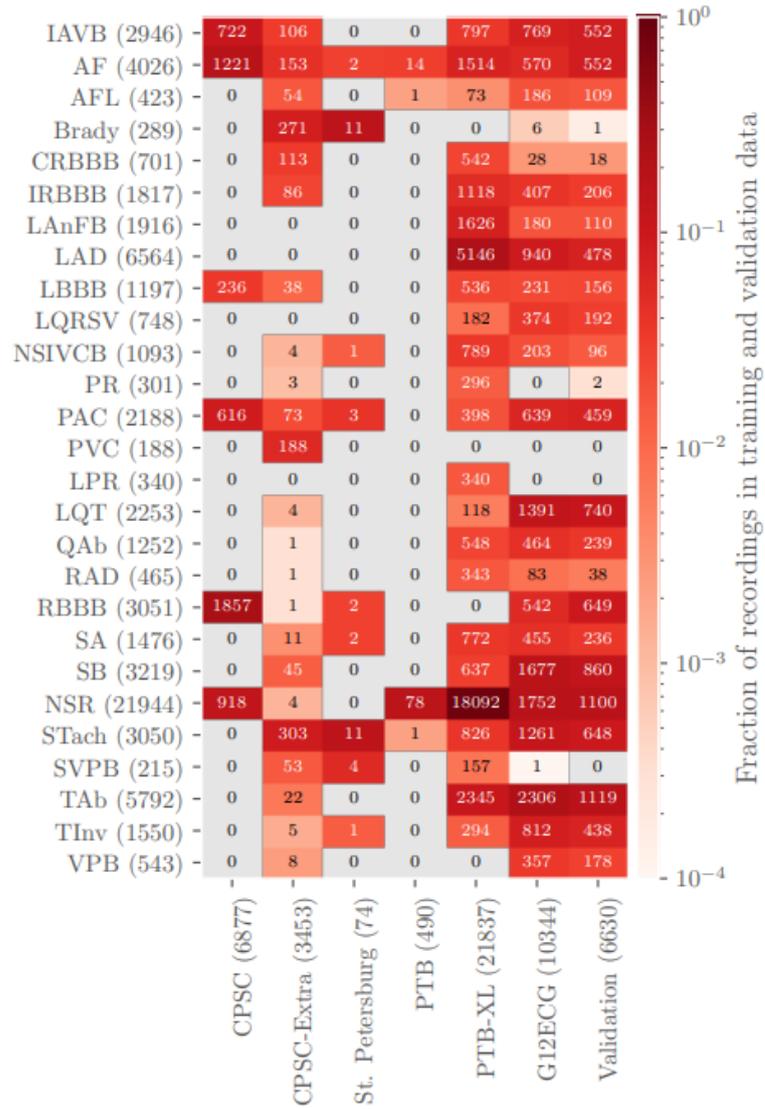


Figure 2.1 - Numbers of recordings with each scored diagnosis in the training and validation sets.

All data is provided in WFDB format. Each ECG recording has a binary MATLAB v4 file for the ECG signal data and a text file in WFDB header format describing the recording and patient attributes, including the diagnosis [18] [10].

## 2.5 Conclusion

Probably the most diffused dataset for ECG signal analysis and arrhythmia detection among all those freely available is the MIT-BIH Arrhythmia database. This is because, first the dataset includes recordings of all five arrhythmia classes suggested by the AAMI standards [50] and second, as all the recordings included are annotated by two cardiologists, it makes it suitable for both supervised and unsupervised learning techniques. On the other hand, the limitations of the MIT-BIH dataset in terms of (a) signal quality since only two ECG leads were used during the recordings; and (b) dataset size, since the different types of arrhythmias identified on the 48 patients, inevitably restrict the generality of the trained AI-models.

As I previously said, the AHA Database has an excellent representation of the most severe types of ventricular ectopy, but it has relatively few examples of supraventricular ectopy, conduction defects, and noise-contaminated waveforms, all of which are common in clinical practice. In the other hand the European ST-T Database is developed specifically for abnormalities in the ST segment and the T wave indicative of myocardial ischemia.

So, I decided to carry out the experiment on the training set of the more recent and well-structured 2020 PhysioNet Challenge dataset. The choice was mostly conducted based on the fact that the PhysioNet database has a more precise focus on the rhythmic arrhythmias, instead of the morphological ones like the MIT-BIH database do. Moreover, the PhysioNet dataset includes 12 leads ECG recordings from 43,059 patients instead of the two-channel ones obtained from 47 subjects in the other database. Finally, the data of the chosen dataset come from different

hospitals, and this suits well also with the purpose to present a federated learning approach to deal with privacy-related issues.

### 3. Previous related work

#### 3.1 Arrhythmia detection

Along the literature I could find that a huge number of diverse techniques have been applied when classifying ECG' arrhythmias. As an example [19], [20], and [21] focused their efforts on using Deep Neural Networks (Artificial Neural Networks and Multi-layer Perceptron) to get a model that predicts the abnormality given the ECG signal. In comparison, the author of [22] combined the use of Naïve Bayes, Adaboost, Random Forest and Support Vector Machines to get the best classifier for his work. Finally, [23], [24] and [25] employed in their research some Convolutional Neural Network approaches. Among them the highlighted Squeezenet, Attention mechanism and Resnet as the champion methods to deal with the ECG detection.

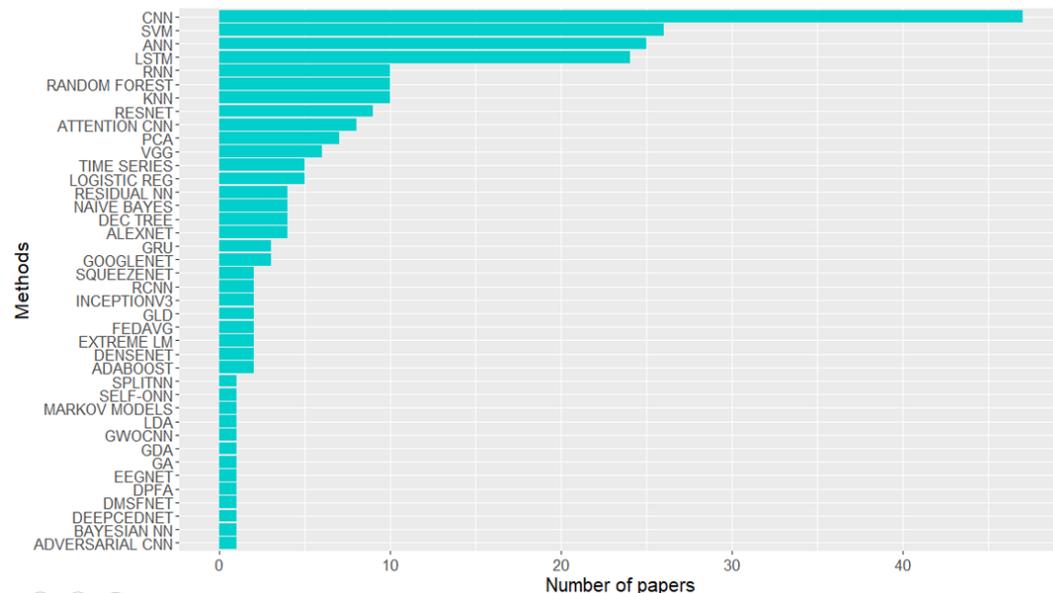


Figure 3.1 - Most used classification methods for ECG

As depicted in Figure 3.1, the most used technique is the Convolutional Neural Network (CNN). That one includes also some self-made Deep Neural Networks (DNN). On the second place we find Support Vector Machines (SVM) and Artificial Neural Networks (ANN). And very close to them most of the author also used Long Short-Term Memory (LSTM) algorithms. On the opposite, a few papers contributed with techniques like GWOCNN, DFPA, DEEPCETNET, etc., which are also CNN but that have specific alterations adapted to by the papers' authors.

With respect to ECG arrhythmia classification, there are plenty of measurements employed in the literature. In Figure 3.2 are shown the most relevant metrics used in this aim, gathered from the available articles and papers that I found in the literature.

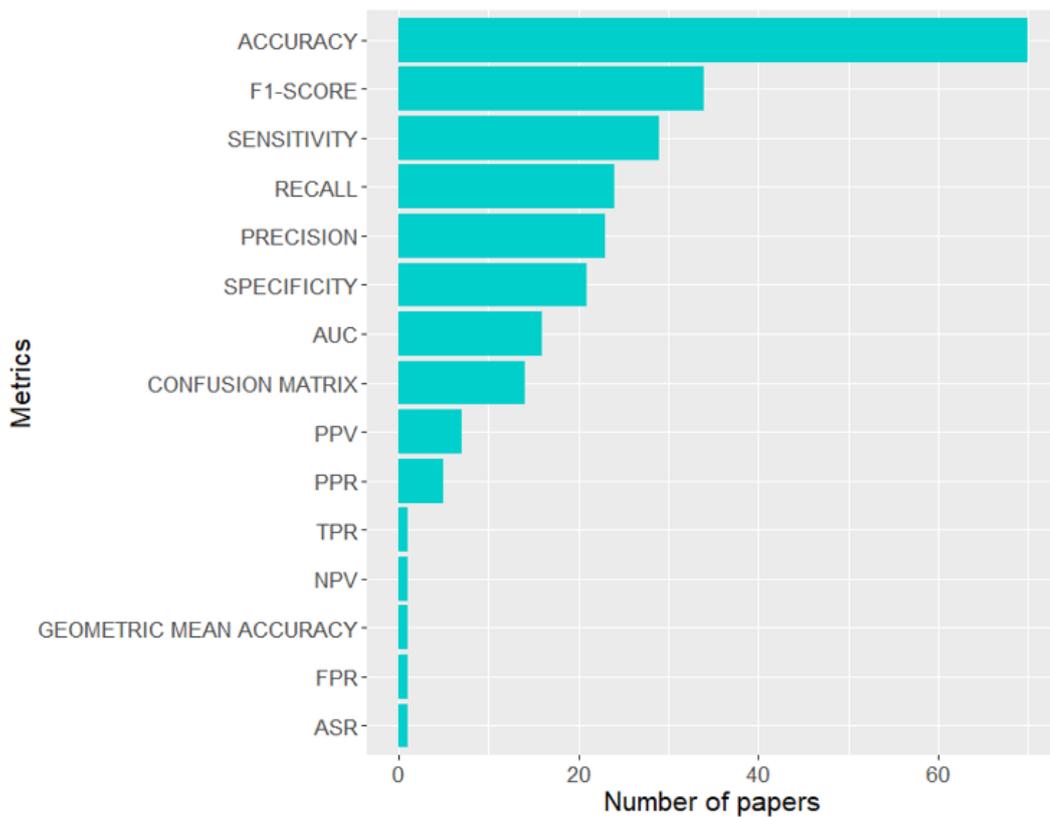


Figure 3.2 - Most used metrics for ECG classification

From the previous chart it is evident that the most used measure is the Accuracy. In the second place, the F1-Score is often used. It is important to clarify that the latter is preferred when dealing with unbalanced data, since it takes into account both Recall and Precision for its calculation. Some papers that consider the 4 mentioned measures at the same time can be found in [26], [27] and [28].

### **3.2 2020 PhysioNet Challenge papers**

Since I decided to use the 2020 PhysioNet Challenge dataset, it is also useful to consider and finding out what papers has been published in the challenge and what kind of architecture the participants were following.

41 teams got selected during this challenge and others are simply discarded because the method did not work on the hidden set, the team failed to submit a preprint of a final article on time, or the team was absent during CinC conference. Looking at the 41 teams' papers published, one can observe that some techniques are used by the majority of the teams. Among these techniques, signal processing, DNNs, Convolutional Neural Networks, end-to-end and multi-binary classifications are used by all of the top 10 teams. In addition, there are several important points that emerged:

1. Deep-learning methods were more popular than traditional methods in Challenge 2020.
2. All the teams that employed deep-learning methods used CNNs.
3. None of the top-10 teams used hand-labeled features (except demographic features); they all adopted end-to-end models instead.

To investigate the techniques applied by each team, I considered five aspects of the methods that formed a solution pipeline: data preprocessing,

feature engineering, machine-learning models, training strategy, and applications to the real world (Table 3.1 – Details of employed techniques. Table 3.1).

| Aspect                         | Inclusion                           | Usage(%) | # in top 10 methods | p-value |
|--------------------------------|-------------------------------------|----------|---------------------|---------|
| Data preprocessing             | Signal processing                   | 95.12    | 10                  | N.A.    |
|                                | Data augmentation                   | 31.70    | 6                   | 0.071   |
|                                | Imbalance handling                  | 53.66    | 7                   | 0.252   |
| Features engineering           | Hand features                       | 36.59    | 0                   | 0.983   |
|                                | Demographic features                | 29.27    | 5                   | 0.109   |
| ML models                      | DNN                                 | 82.93    | 10                  | 0.116   |
|                                | CNN                                 | 82.93    | 10                  | 0.116   |
|                                | RNN/transformer                     | 31.71    | 4                   | 0.317   |
|                                | Attention                           | 24.39    | 6                   | 0.006   |
| Training strategy              | Model ensemble                      | 36.59    | 4                   | 0.878   |
|                                | End-to-end                          | 80.49    | 10                  | 0.139   |
|                                | Multi-binary classification         | 58.54    | 10                  | 0.002   |
| Applications to the real world | Post-processing                     | 2.38     | 1                   | N.A.    |
|                                | Interpretability                    | 4.76     | 0                   | N.A.    |
|                                | Unknown classes and unseen patients | 0        | 0                   | N.A.    |

N.A. means that the hypothesis test is not conducted.

Table 3.1 – Details of employed techniques.

One can also notice that the three highest-ranking teams used the model ensemble [29], [30] and [31], but only 14 out of 41 teams employed this strategy [32]. It is also important to note that model ensemble only helps if

used for single model rather than models that are structurally different. Most of teams also used only age and sex as features rather other using demographic features or 12-lead ECG based features. The training data in Challenge 2020 suffer from heavy class imbalance, so most teams used threshold optimization [33] [31] [34] and weighted loss [35] [36] to handle imbalance class issue. In addition, oversampling [37], down-sampling [38], and other methods have been employed in Challenge 2020's submissions.

## 4. Techniques and tools

### 4.1 Classification techniques and tools

This section goes over the most important analysis and approaches for studying, modeling, and predicting ECG arrhythmia diagnosis.

#### 4.1.1 Data Wrangling (DW)

Data wrangling is the act of cleaning and combining chaotic and difficult data sets for easy access and analysis. With the amount of data and data sources growing all the time, it's more vital than ever to arrange massive volumes of data for analysis [39].

To facilitate data consumption and organization, this method normally requires manually transforming and mapping data from one raw format to another. The most relevant Data Wrangling's objectives are [39]:

- Collect data from a variety of sources to uncover "deeper intelligence."
- As soon as feasible, get reliable, actionable data into the hands of business analysts.
- Reduce the amount of time it takes to collect and organize jumbled data before it can be used.
- Allow data scientists and analysts to focus on data analysis instead of data manipulation.
- Encourage senior executives in a company to improve their decision-making skills.



Figure 4.1 - Main steps in Data Wrangling

The data wrangling approach typically consists of six iterative steps, as seen in Figure 4.1, as mentioned by [40]:

1. **Publishing:** Data wranglers prepare data for downstream usage - whether by a specific user or program - and identify any special actions or logic that were employed to do so.
2. **Discovering:** Before delving into the data, it's important to first have a better knowledge of what's there, since this will influence how you examine the data.
3. **Validating:** These are recurrent programming sequences that verify data quality, consistency, and security. Validation can include things like ensuring that qualities that should be distributed on a regular basis are distributed uniformly.
4. **Enrichment:** "What more types of data can be obtained from what already exists?" one can question during the data wrangling stage, or "What further information could assist me in making better selections based on the current data?" .
5. **Structuring:** The data must be structured in this step of data wrangling because raw data arrives in a range of formats and sizes.

6. **Cleaning:** By altering null values and establishing standard formats, data wrangling aims to improve data quality.

### 4.1.2 Feature Engineering (FE)

The act of choosing, altering, and transforming raw data into features that may be utilized in supervised learning is known as feature engineering. It may be necessary to build and train better features in order for machine learning to perform well on new datasets.

### 4.1.3 Challenge features

Within the PhysioNet 2020, the organizers provided a code that calculated 14 features leveraged on the recordings. Those variables were based on the R-Peaks and the RR interval:

- **R-Peaks:** It refers to the R wave's highest amplitude (as seen in Figure 1.5).
- **RR-Interval:** On an ECG, it is the period between two consecutive R-waves of the QRS signal. The former is determined by the sinus node's inherent features as well as autonomic factors.

Then, with the previous measures, the competence calculated the mean, median, standard deviation, variance, skewness, and kurtosis only for the first lead. In addition, they used the age and sex provided with the initial raw data.

### 4.1.4 Spectral features

Leveraged on the solution developed by [41], I implemented 636 features that deals with the spectral part of the signals provided in the ECG. Spectral

analysis (where the spectral features were derived) is a frequently utilized tool for exploring biomedical data. The waveform component forms, their time positions within the cardiac cycle, and the regularity of the heart period all influence the ECG signal's spectrum [42]. Usually the Fourier Transform (FT) is used to extract information from signals like ECG. Nevertheless, the Fourier Transform has the drawback of capturing global frequency information, or frequencies that are present throughout a whole signal. This type of signal decomposition may not be appropriate for many applications, such as electrocardiography, which involves signals with short periods of distinctive oscillation. The Wavelet Transform, which decomposes a function into a set of wavelets, is another option that corrects the FT approach [43].

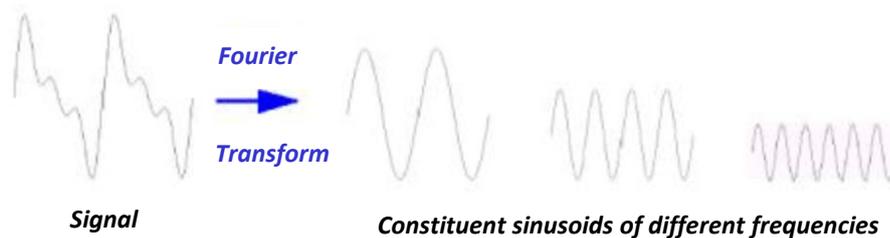


Figure 4.2 - Wavelet representation

A Wavelet is a time-localized wave-like oscillation; an example is shown in Figure 4.2. Scale and location are the two most basic features of wavelets. The scale (or dilation) of a wavelet determines how "stretched" or "squished" it is. This attribute has to do with how waves are characterized in terms of frequency. The wavelet's position in time is defined by its location (or space).

Then, the schema of features calculated is as follows. For each lead calculate:

1. **Statistics:** Percentiles (5, 25, 50, 75, 95), mean, standard deviation and variance for the complete signals.
2. Calculate **coefficients** of **Discrete Wavelet Transform (DWT)**. DWT gets local frequencies for the signals. The Coefficients are calculated using the function `wavedec` from the Python's library *pywt*.
3. For each **coefficient** of DWT calculate:
  - **Statistics:** Percentiles (5, 25, 50, 75, 95), mean, standard deviation and variance.
  - **Shannon's entropy** (same that entropy): It's related to the "amount of information" of a variable. In other words, it measures information of the distribution.

#### 4.1.5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data to identify patterns, spot anomalies, test hypotheses, and check assumptions using summary statistics and graphical representations. It's important to initially comprehend the data before attempting to get as many insights as possible. EDA is all about making sense of data before getting their hands dirty with it. The major steps commonly examined in an EDA are shown in Figure 4.3.

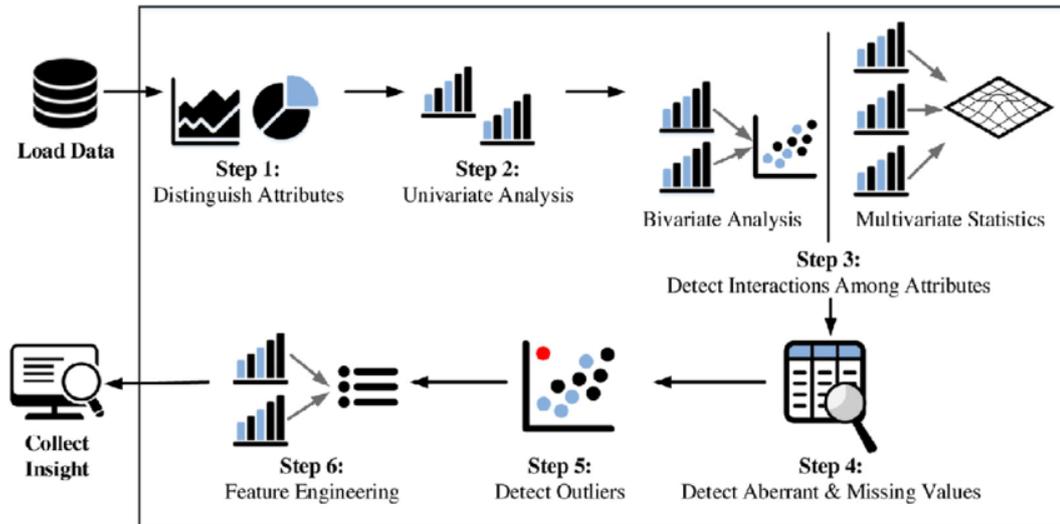


Figure 4.3 - Schema of an EDA

#### 4.1.6 Unbalanced classes

One of the most difficult issues when training a model is modeling imbalanced data [44]. When dealing with classification problems, the intended class balance is quite important. When a dataset has an uneven distribution of classes, the models attempt to learn only the dominant class, resulting in biased predictions.

One approach for addressing this issue is random sampling. Random resampling can be accomplished in two ways, each with its own set of benefits and drawbacks:

- **Oversampling:** Replicating examples from the minority class.
- **Undersampling:** Deleting examples from the majority class.

To put it another way, both over-sampling and under-sampling include creating bias by selecting more instances from one class than from another. The prior is used to compensate for an imbalance that is already present in the data or that is likely to occur if a perfectly random sample is obtained [45]. Because it makes no assumptions about the data, random sampling is

a naive strategy. To minimize the data's influence on the Machine Learning algorithm, a fresh adjusted version of the data with a new class distribution is generated.

Random Oversampling (ROS) and Synthetic Minority Oversampling Technique (SMOTE) were the two oversampling techniques chosen for this project. SMOTE is a technique for creating synthetic samples for the minority class. Overcoming the problem of overfitting produced by random oversampling is easier with this method. It focuses on the feature space in order to generate new examples by interpolating between positive occurrences that are near in proximity.

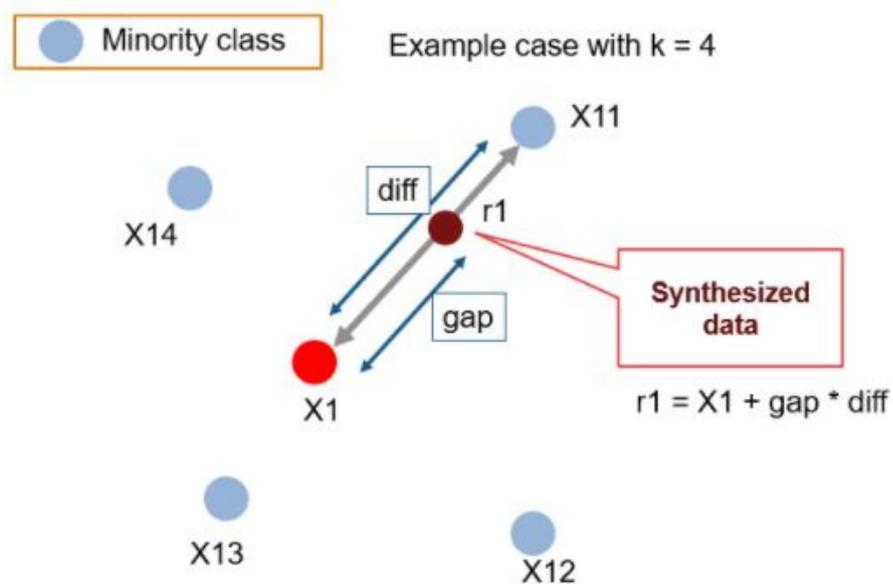


Figure 4.4 - SMOTE process

SMOTE uses the k-nearest neighbor technique to create synthetic data. To make them, it follows the instructions below [44]:

1. Find the nearest neighbors of the feature vector.
2. Determine the distance between the two sample sites.
3. At random, the distance is multiplied by an integer between 0 and 1.

4. Find a new point on the line segment at the calculated distance.
5. Rep the procedure for each of the feature vectors that were discovered.

### **4.1.7 Machine Learning models**

Classifiers are the models provided in the following sections. These tools were created with the goal of determining which behaviors are more likely to be associated with various arrhythmia patterns. Each of these methods is widely utilized in various data-driven systems, and they have demonstrated useful behavior in a variety of classifying tasks, including ECG classification. The various versions of the dataset were created using Python Notebooks in Google Colab. This section details the key models that were tested and evaluated.

#### **4.1.7.1 XG-Boost (XGB)**

The XG-Boost technique, which has proven to be effective in a variety of classification and regression problems, is the first attempt to classify the ECG signals. The algorithm has been used to a variety of sectors, including economics, credit rating, and health-related difficulties. The preceding are reasons to expect that such a strategy will be effective in the field of arrhythmia detection today.

XG-Boost is a decision-tree-based ensemble Machine Learning approach that uses gradient boosting [46] [47]. When it comes to unstructured data prediction, Artificial Neural Networks outperform all other algorithms or frameworks (text, audio, pictures, etc.). However, for small-to-medium

tabular data, such as the one utilized in this challenge, decision tree-based algorithms are now rated best-in-class.

XG-Boost minimizes a loss function to provide an additive expansion of the objective function, similar to gradient boosting. Because XG-Boost is only interested in decision trees as base classifiers, the complexity of the trees is controlled using a variation of the loss function.

$$L = \sum_{i=1}^n L(y_i, \hat{y}_i) + \sum_{k=1}^K \Theta(p_k) \quad (5.1)$$

$$\Theta(w) = \gamma Z + \frac{1}{2} \lambda \|w\|^2 \quad (5.2)$$

The number of leaves on the tree is  $Z$ , and the leaf output scores are  $w$  [47]. This loss function can be included into the split criterion of decision trees, resulting in a pre-pruning strategy. Trees with a greater  $\gamma$  value are easier to understand. The amount of loss reduction gain required to separate an internal node is determined by  $\gamma$  [46]. Shrinkage is a regularization parameter in XG-Boost that decreases the step size in the additive expansion. Finally, other techniques such as tree depth can be utilized to keep the trees from becoming too complex. As a result of lowering tree complexity, the models are trained faster and need less storage space.

### 4.1.7.2 CatBoost

The second candidate in predicting the arrhythmia type for ECG is the CatBoost algorithm. The latter is a decision tree gradient boosting technique. It was created by Yandex (with its final version in 2017) researchers and engineers and is used by Yandex and other firms such as CERN, Cloudflare, and Careem taxi for search, recommendation systems, personal assistant, self-driving cars, weather prediction, and many other activities. It is open source.

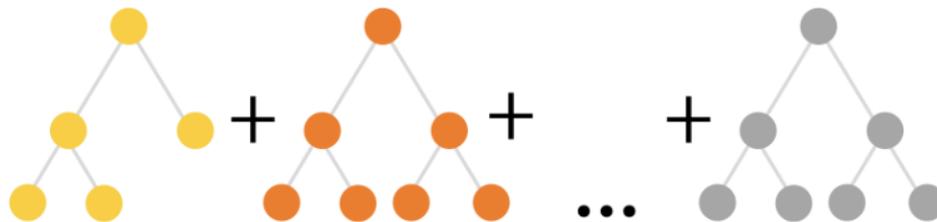


Figure 4.5 - Catboost (decision trees)

The implementation of ordered boosting [48], a permutation-driven alternative to the conventional approach, and a novel technique for processing category characteristics are two key algorithmic innovations offered in CatBoost. Both strategies were developed in order to combat a prediction shift induced by a specific type of target leakage found in all current gradient boosting algorithm implementations.

### 4.1.7.3 Deep Neural Network (DNN)

A Deep Neural Network is another method that could be used for predicting ECG diagnosis. A DNN is a set of algorithms that attempts to

recognize relationships in a batch of data by mimicking how the human brain functions.

In this context, deep neural networks refer to organic or artificial systems of neurons [49]. Deep neural networks can adapt to changing input and produce the best possible result without requiring the output criteria to be modified because they can adapt to changing input. Neural networks, an artificial intelligence-based concept, are swiftly gaining popularity in the development of trading systems.

Neural networks aid in time-series forecasting, algorithmic trading, securities classification, credit risk modeling, and the generation of proprietary indicators and price derivatives in the financial world [50] [51]. The deep neural network of the human brain is akin to a neural network. A "neuron" in a deep neural network is a mathematical function that collects and categorizes data according to a set of rules. The network closely resembles curve fitting and regression analysis, two statistical methods.

Perceptrons are grouped in interconnected layers in a multi-layered perceptron (MLP) [51], as indicated in Figure 4.6. The input layer is responsible for collecting input patterns. In the output layer, input patterns can be mapped to classifications or output signals. Hidden layers fine-tune the input weightings until the neural network's margin of error is as little as possible. Hidden layers are supposed to deduce salient elements from input data that have the ability to predict outcomes. This is how feature extraction works, and it's similar to how statistical methods such as principal component analysis function [51].

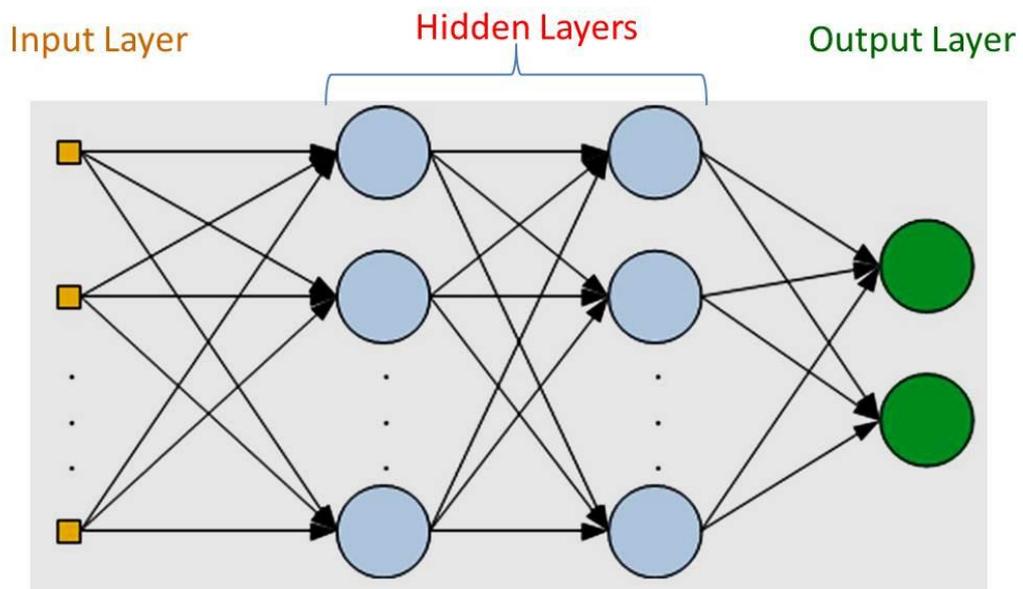


Figure 4.6 – Deep Neural Network (Multi-layer Perceptron) schema

#### 4.1.7.4 Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) networks are a type of Deep Learning network. It's a class of recurrent neural networks (RNNs) that can learn long-term dependencies, which is useful for solving sequence prediction issues. Apart from single data points like photos, LSTM has feedback connections, which means it can process the complete sequence of data.

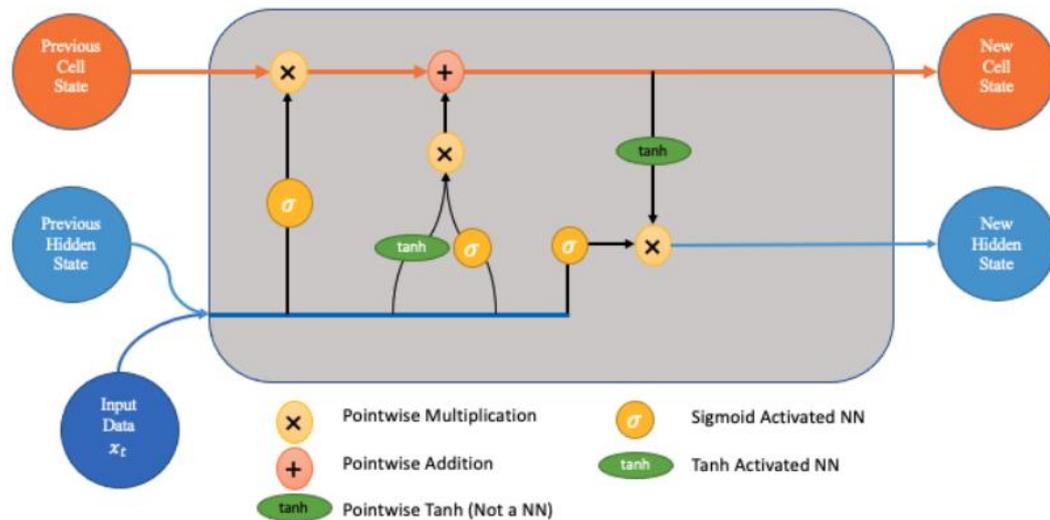


Figure 4.7 - Long-Short Term Memory schema

An LSTM model's primary role is played by a memory cell called a "cell state" which maintains its state across time. The horizontal line that runs through the top of the diagram below (Figure 4.7) represents the cell state. It can be compared to a conveyor belt on which data just passes, unmodified [52].

#### 4.1.8 Metrics

It is vital to create metrics that will assist in determining whether a model is better than others. There are explanations for each of the metrics used in the following subparagraphs.

##### 4.1.8.1 Confusion matrix

A confusion matrix, like the one shown in Table 4.1, demonstrates how well a classification model works on test data for which the true values are known [53]. The confusion matrix is simple in itself, but the related nomenclature can be confusing. In the following examples, I've created a

hypothetical target variable called "Diagnose A" with the values "Yes" (if the recording belongs to that diagnose) and "No" (if the recording does not belong to that diagnose).

|                    |                       | Actual Class          |                      |
|--------------------|-----------------------|-----------------------|----------------------|
|                    |                       | Diagnose A<br>YES = 1 | Diagnose A<br>NO = 0 |
| Predicted<br>Class | Diagnose A<br>YES = 1 | True Positives (TP)   | False Positives (FP) |
|                    | Diagnose A<br>NO = 0  | False Negatives (FN)  | True Negatives (TN)  |

Table 4.1 - Confusion matrix

Here is an explanation for each of the matrix's elements to understand the preceding terminology [53] [54].

- **True negatives (TN):** The model predicted that they wouldn't have the diagnose A, and they don't.
- **True positives (TP):** These are examples when the model predicted yes (those recording has the diagnose A), and they actually don't.
- **False positives (FP):** The model projected that they would have the diagnose A, but they don't. (This is often referred to as a "Type I error").
- **False negatives (FN):** The model anticipated that they would not have diagnose A, yet they do. (This is often referred to as a "Type II error".)

### 4.1.8.2 Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5.3)$$

The most basic performance metric is accuracy, which is defined as the proportion of correctly predicted observations to all observations. If a model is correct, one would assume it is the best. Accuracy is a relevant measure when the datasets are symmetric, and the number of false positives and false negatives is about equal.

Imagining, for example, the case when the training set contains 98% samples of class A and 2% samples of class B. The model may thus easily attain a 98% training accuracy by simply guessing every training sample that belongs to class A. When the same model is tested on a test set that contains 60% class A samples and 40% class B samples, the test accuracy reduces to 60%. As a result, classification accuracy is poor, but it gives the image of great accuracy. Then, when the cost of misclassification of minor class samples becomes significant, [54] the true issue appears. The cost of failing to diagnose, for example, a sick person's ailment is significantly greater than the expense of submitting a healthy person to additional tests when dealing with a rare but lethal disorder.

### 4.1.8.3 Precision

$$Precision = \frac{TP}{TP + FP} \quad (5.4)$$

Precision [53] is the ratio of accurately predicted positive observations to total expected positive observations. This measure answers the question of how many of the drivers who were identified as drowsy actually drove. Precision is linked to a low false-positive rate. Precision is a good statistic to employ when the costs of false positive are high. Take, for example, the identification of email spam. In email spam detection, a false positive happens when an email that is not spam (actual negative) is wrongly identified as spam (predicted spam). If the precision of the spam detection model is low, the email user may miss important emails.

#### 4.1.8.4 Recall

$$Recall = \frac{TP}{TP + FN} \quad (5.5)$$

Recall [53] is the ratio of successfully predicted positive observations to all observations in the actual class. It's meant to answer the question of how many drivers who actually slept were labeled as such. In the case of identifying sick patients, for example, if a sick patient (Actual Positive) conducts the test and is predicted to be healthy (Predicted Negative). The cost of false negative will be quite high if the condition is infectious.

#### 4.1.8.5 F1 Score

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \quad (5.6)$$

The F1-Score is the weighted average of Precision and Recall. As a result, both false positives and false negatives are taken into account in this score. F1-Score is often more valuable than accuracy, despite the fact that it is less intuitive [54] [53]. This is especially true if the class distribution is unequal. When the costs of false positives and false negatives are equal, accuracy works well. If the cost of false positives and false negatives differs significantly, it is best to evaluate both Precision and Recall.

#### 4.1.8.6 Overall index

As a final metric, I introduced the use of the mean value of Accuracy, Precision, Recall and F1-Score, that I called "Overall index". The Overall index is really useful because it allows to check with only one metric the overall behavior of the classifiers.

## 4.2 TensorFlow Lite for Microcontrollers

### 4.2.1 The importance of microcontrollers

Microcontrollers are typically small, low-powered computing devices that are embedded within hardware that requires basic computation. By bringing machine learning to tiny microcontrollers, we can increase the intelligence of the billions of everyday objects we use, such as home

appliances and Internet of Things devices, without having to invest in costly hardware or dependable internet connections, which are frequently constrained by bandwidth and power and produce significant latency. Since no data leaves the device, this can also help protect privacy [55].

Summing up, some of the pros to fit ML on microcontrollers are the following:

- **Function:** wanting a smart device to act quickly and locally.
- **Cost:** accomplishing this with simple, lower cost hardware.
- **Privacy:** not wanting to share all sensor data externally.
- **Efficiency:** smaller device form-factor, energy-harvesting or longer battery life.

## 4.2.2 TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications [56].

TensorFlow's high-level APIs are based on the Keras API standard for defining and training neural networks. Keras enables fast prototyping, state-of-the-art research, and production.

### 4.2.2.1 History

TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was made available in 2015 under the Apache License 2.0. In September 2019, Google published

TensorFlow 2.0, an upgraded version of the software. A large number of programming languages, including Python, Javascript, C++, and Java, are compatible with TensorFlow and this adaptability makes it suitable for a variety of uses across numerous sectors [57].

### **4.2.3 TensorFlow Lite**

TensorFlow Lite has APIs for mobile apps or embedded devices to generate and deploy TensorFlow models. These models are compressed and optimized in order to be more efficient and have higher performance on smaller capacity devices. TensorFlow Lite uses FlatBuffers as the data serialization format for network models, eschewing the Protocol Buffers format used by standard TensorFlow models [57].

#### **4.2.3.1 TensorFlow Lite for Microcontrollers**

In particular, TensorFlow Lite for Microcontrollers (TFLM) is designed to run machine learning models on microcontrollers and other devices with only few kilobytes of memory. The core runtime just fits in 16 KB on an Arm Cortex M3 and can run many basic models. It doesn't require operating system support, any standard C or C++ libraries, or dynamic memory allocation.

TFLM is written in C++ 11 and requires a 32-bit platform. It has been tested extensively with many processors based on the Arm Cortex-M Series architecture and has been ported to other architectures including ESP32. It is open source and can be included in any C++ 11 project [55].

The following development boards are supported:

- Arduino Nano 33 BLE Sense
- SparkFun Edge
- STM32F746 Discovery kit
- Adafruit EdgeBadge
- Adafruit TensorFlow Lite for Microcontrollers Kit
- Adafruit Circuit Playground Bluefruit
- Espressif ESP32-DevKitC
- Espressif ESP-EYE
- Wio Terminal: ATSAMD51
- Himax WE-I Plus EVB Endpoint AI Development Board
- Synopsys DesignWare ARC EM Software Development Platform
- Sony Spresense

TensorFlow Lite for Microcontrollers is designed for the specific constraints of microcontroller development, and there are some limitations to be considered:

- Support for a limited subset of TensorFlow operations.
- Support for a limited set of devices.
- Low-level C++ API requiring manual memory management.
- On device training is not supported.

#### **4.2.4 Model optimization**

Often, the memory or processing power of edge devices is constrained. To enable models to operate within these restrictions, various optimizations might be used.

[58] The benefits of the model optimization are:

- **Size reduction:** some forms of optimization can be used to reduce the size of a model. Smaller models utilize less RAM when they are executed, take up less memory on the devices, and take less time and bandwidth to download to users' devices. It can translate to better performance and stability.
- **Latency reduction:** some forms of optimization can reduce the latency, amount of time required to run a single inference using a model, resulting in lower latency. It can also have an impact on power consumption.
- **Accelerator compatibility:** with properly optimized models, some hardware accelerators can perform inference very quickly.

TensorFlow Lite currently supports optimization via:

- **Quantization** that works by reducing the precision of the numbers used to represent a model's parameters, which by default are 32-bit floating point numbers. This results in a smaller model size and faster computation. Figure 4.8 schematized all post-training quantization methods.
- **Pruning:** works by removing parameters within a model that have only a minor impact on its predictions. Pruned models are the same size on disk, and have the same runtime latency, but can be compressed more effectively.
- **Clustering:** works by grouping the weights of each layer in a model into a predefined number of clusters, then sharing the centroid values for the weights belonging to each individual cluster. This reduces the number of unique weight values in a model, thus

reducing its complexity. As a result, clustered models can be compressed more effectively.

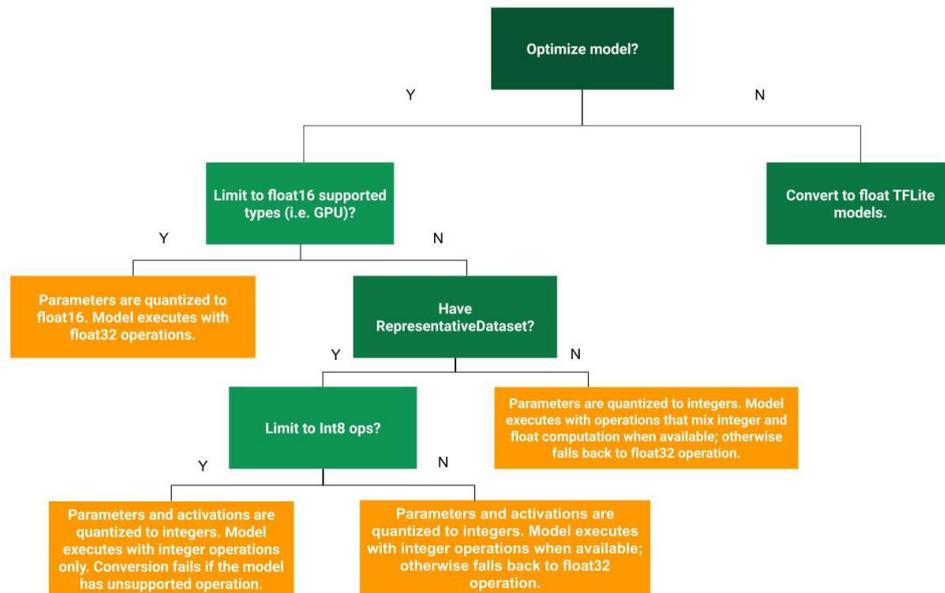


Figure 4.8 - Post-training quantization methods

# 5. Arrhythmia classification on a low-power device

## 5.1 Arrhythmia classification

### 5.1.1 Dataset

As mentioned in Section 2.5, I decided to conduct my experiments on the training set of the 2020 PhysioNet Challenge Dataset.

Refer to Section 2.4.2.2 to see the database's structure in detail.

### 5.1.2 Data wrangling

As mentioned in 4.1.1, the data wrangling process is usually the first step when dealing with a data-oriented problem. In this case, I placed the data in a Google Drive folder after downloading it from the official competition's website [59]. Afterwards, using Google Colaboratory (Google Colab) I extracted and organized the information in Python. The representation of the ECG along the 12-leads can be examined in Figure 5.1.

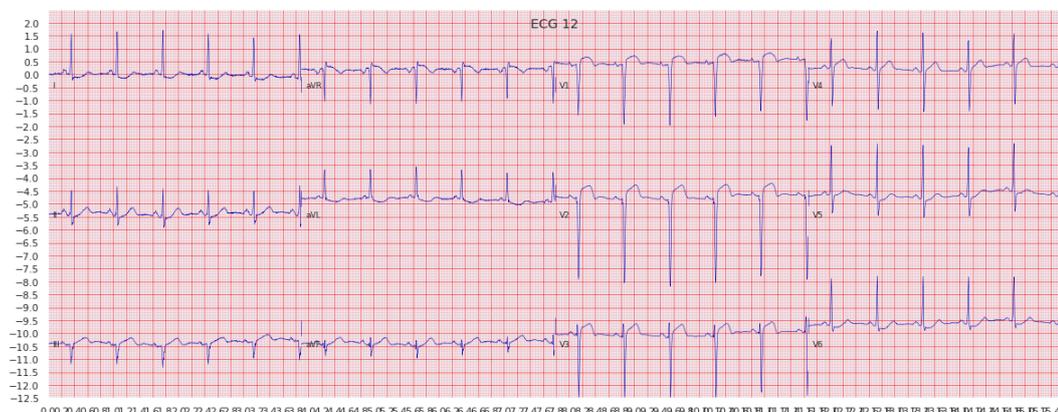


Figure 5.1 - 12-lead ECG for recording S0033 of PTB database

Then, the whole data (all the databases) and the features mentioned in 4.1.2 were calculated. That process took almost 1 hour to run in an using the default configuration of Google Colab. Besides, for each recording I selected the first diagnose (arrhythmia) that appeared as the label to be predicted. The latter process ran in about 2 minutes.

### 5.1.3 EDA

Once the big dataset was loaded, it contained a total of 43,101 recordings and about 764 variables. From the latter, 650 were the features created and the remaining 3 corresponded to the id of the recording, the database that it belongs to and the label (response variable to be predicted). Then, the first analysis need was to examine if the features contained any missing value. Using the function `bar` from the *missingno* library, I managed to explore the missing values. In Figure 5.2 are depicted only the 50 first features' missing counts and percentage. The features names are based on a syntax in which are represented: lead as *l*, their coefficient as *c* and operation applied on it for example: mean, coefficient percentiles, standard deviation etc. Also, `lead_i` represents ECG lead number from 00 to 11, `coefficient_j` represents a coefficient number from Discrete Wavelet Transform (there are 5 coefficients in total from 1 to 5) and `operation_k` represents the operation name like *mean* as average, standard deviation as *std*, variance as *var*, percentiles represented as *n5* (percentile 5), *n25*(percentile 25), *n50* (percentile 50), *n75* (percentile 75), *n95* (percentile 95) and entropy applied on coefficient get represented as *c1-c5*.



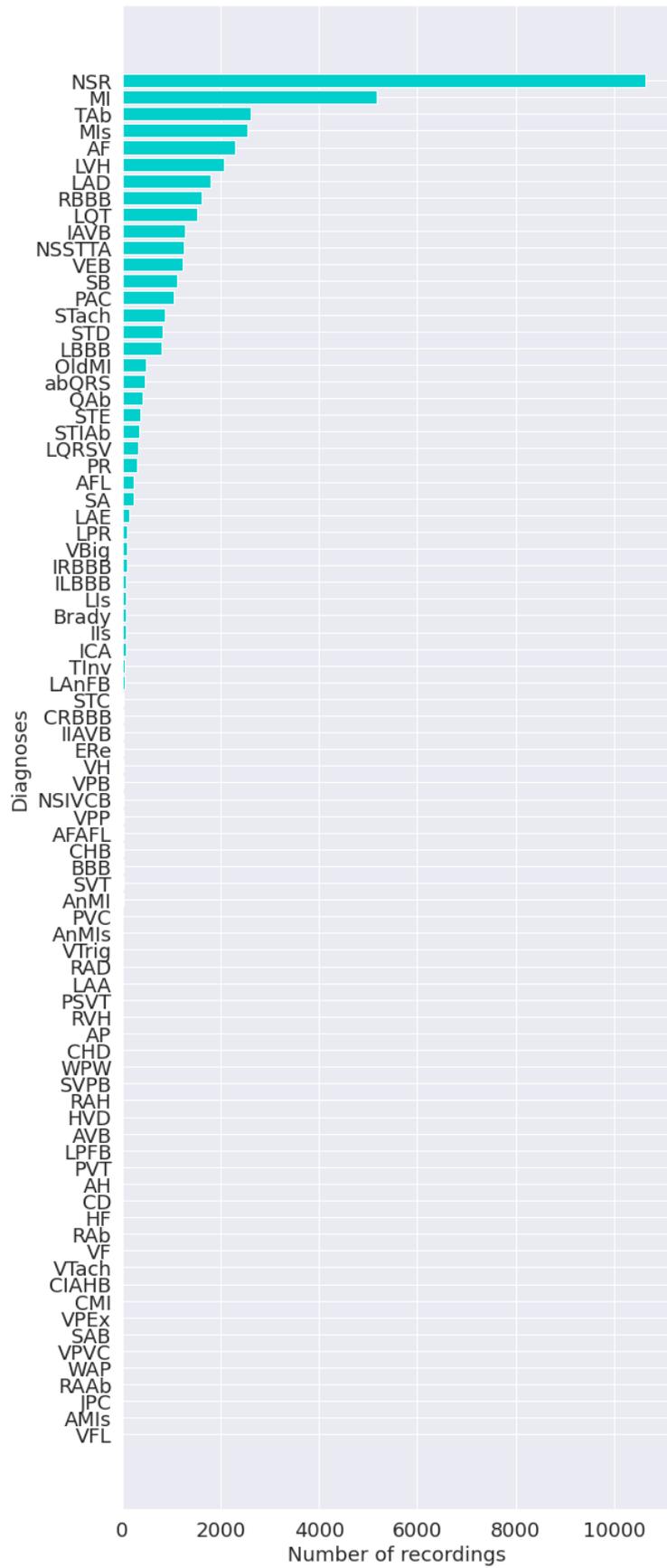


Figure 5.3 - Label distribution for the complete dataset

As evidenced in the previous chart, there are too many arrhythmias that don't have a big participation. That can lead to problems when trying to infer the predicted class of a recording, since there were not enough cases to learn the classifiers properly. That why those diagnoses with a participation smaller than 150 records were discarded from the analysis. With the previous filter, the selected data to work with got a size of 41,894 recordings distributed as shown in Figure 5.4.

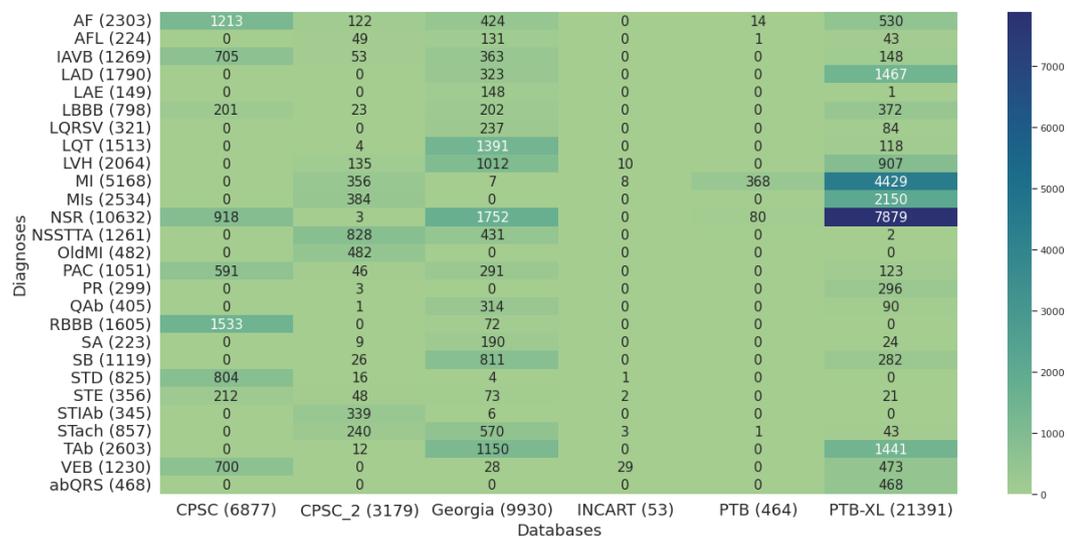


Figure 5.4 - Number of recordings for each diagnosis by database for the filtered data

Besides, the final distribution of the labels ended up as shown in Figure 5.5. As expected, the most common diagnose was Normal Sinus Rhythm (NSR), which is the normal status for an ECG. In addition, the arrhythmia with one of the smallest participations turned out to be Sinus Arrhythmia (SA).

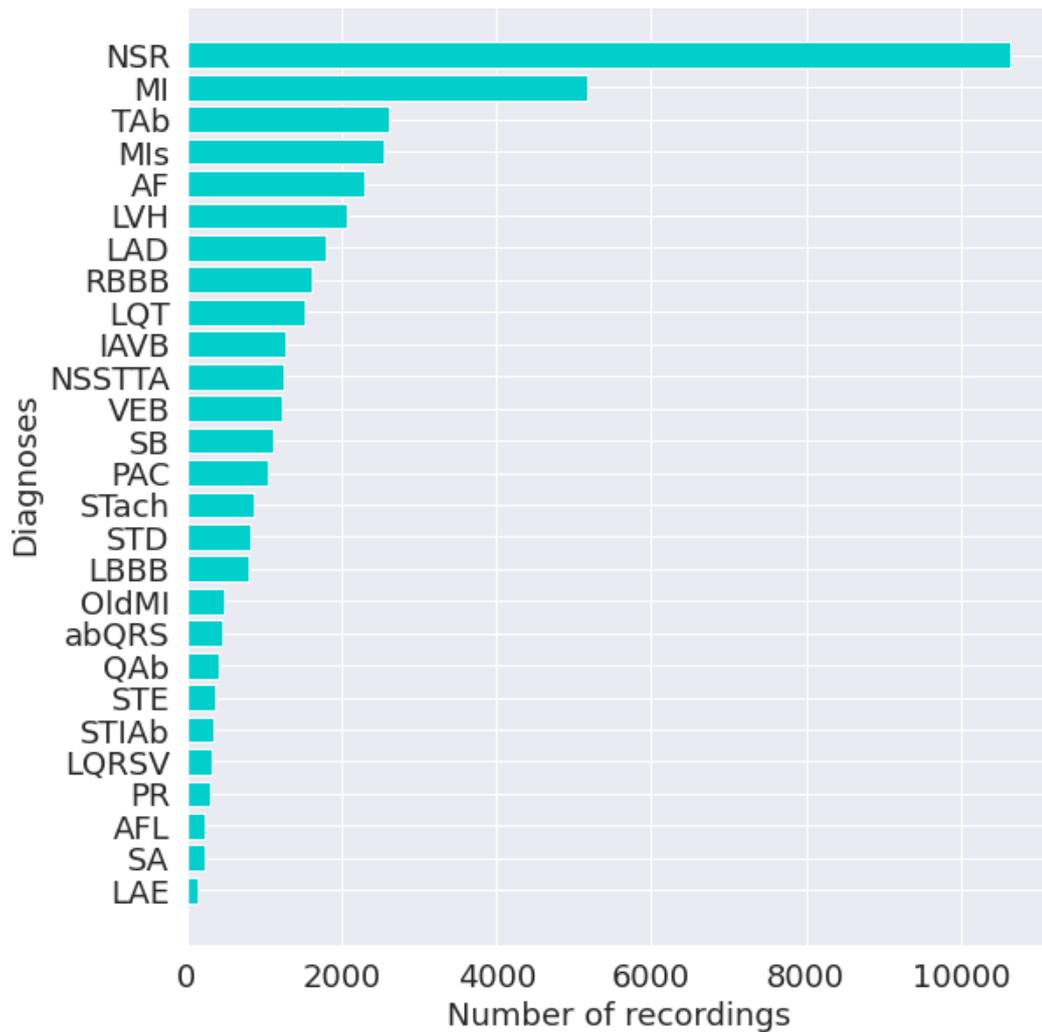


Figure 5.5 - Label distribution for the filtered dataset

### 5.1.4 Feature selection and normalization

In an analytical context, having a huge amount is a double-edged sword. On the one hand, the more information existing to predict a phenomenon, the better. On the other hand, the computational time required to process to much information may lead to training times that are not affordable. Regarding the latter I decided to perform a feature selection step in order to determine the most important features to predict the arrhythmias.

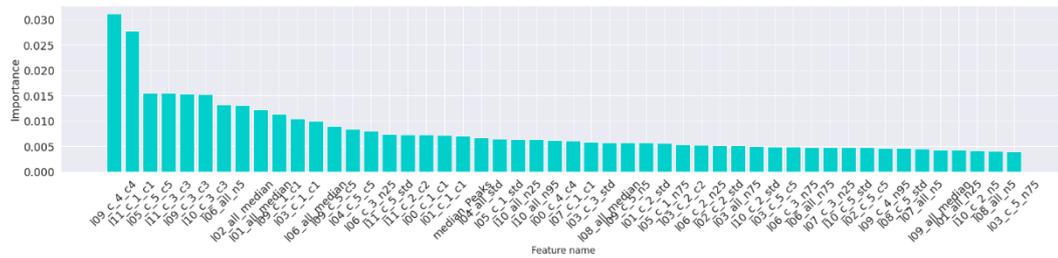


Figure 5.6 - Feature importance from XG-Boost algorithm (only the 50 best)

The bar-plot in Figure 5.6 depicted the most important features to predict the classes obtained by means of the XG-Boost method. The latter provides an automatic raking of the most relevant features to classifier the ECGs. I decided to take the best 120 variables since they managed to get a good enough accuracy, compared to the one obtained using all the features. The best features turned out to be the Entropy for leads: 9, 11, 10; the percentile 5% for lead 6; and the Median for leads 1, 2.

As an additional tool to enhance the performance of the models there was an implementation of features normalization. In this case I tried three different techniques to transform the features to the same scale. The approaches tried were provided by the *sklearn* library in Python. Those are: `StandardScaler`, `MaxMinScaler` and `RobustScaler`. In the end, the scenario that provided the best results was using `RobustScaler`. The latter uses statistics that are resistant to outliers to scale features. The median is removed, and the data is scaled according to the quantile range (defaults to IQR: Interquartile Range). The interquartile range (IQR) is the distance between the first and third quartiles (25th and 3rd quartiles) (75th quantile).

### 5.1.5 Balancing classes

As depicted in Figure 5.5, the diagnoses have an imbalanced characteristic. The latter means that each category has a different participation over the data. That could represent a problem in the performance of the classifiers that will be proposed. Then, two oversampling methods were proposed to deal with the imbalance issue. The first one is called Random Oversampling (ROS). In the latter the minority classes are replicated together with its features. Besides, a down-sampling was applied to have a number of recording similar to the filtered dataset. In the end, the ROS dataset had 43,200 recordings. And as depicted in Figure 5.7, the distribution of the labels is much more similar among the arrhythmia categories.

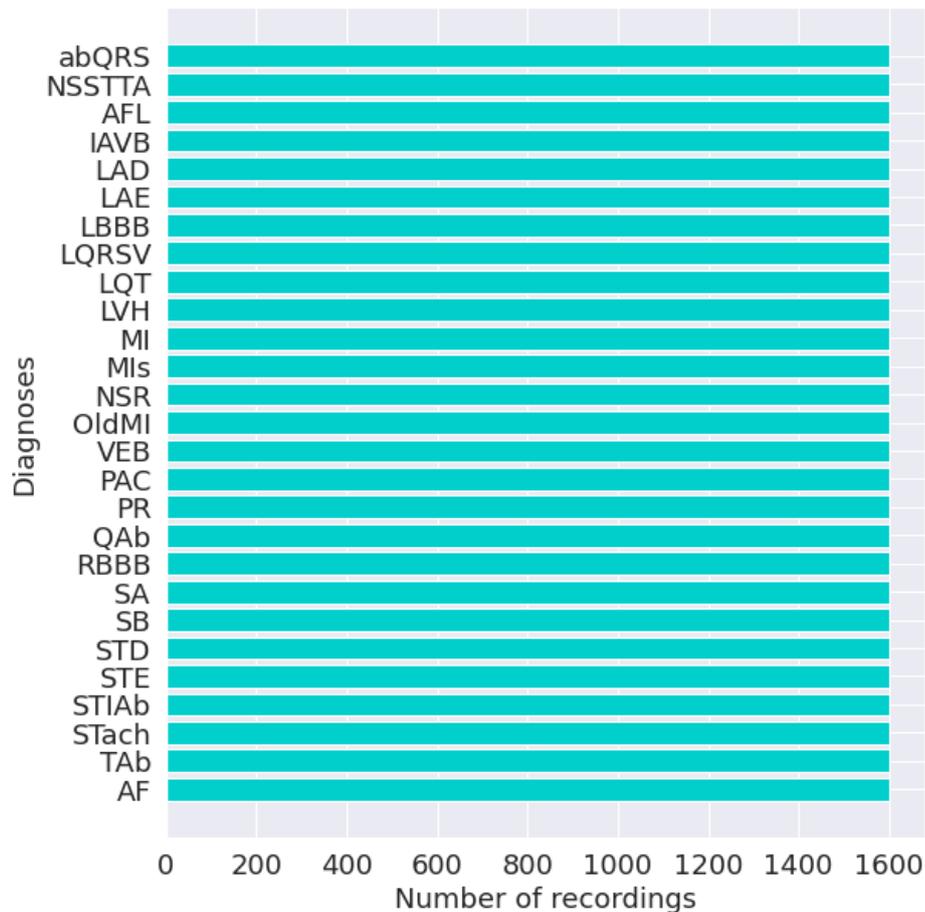


Figure 5.7 - Number of recordings for each diagnosis for the ROS oversampled data

The second oversampling technique used was SMOTE (SMT). In the end, the SMOTE dataset had also 43,200 recordings. And as shown in Figure 5.8, the distribution of the labels is also similar among the arrhythmia categories.

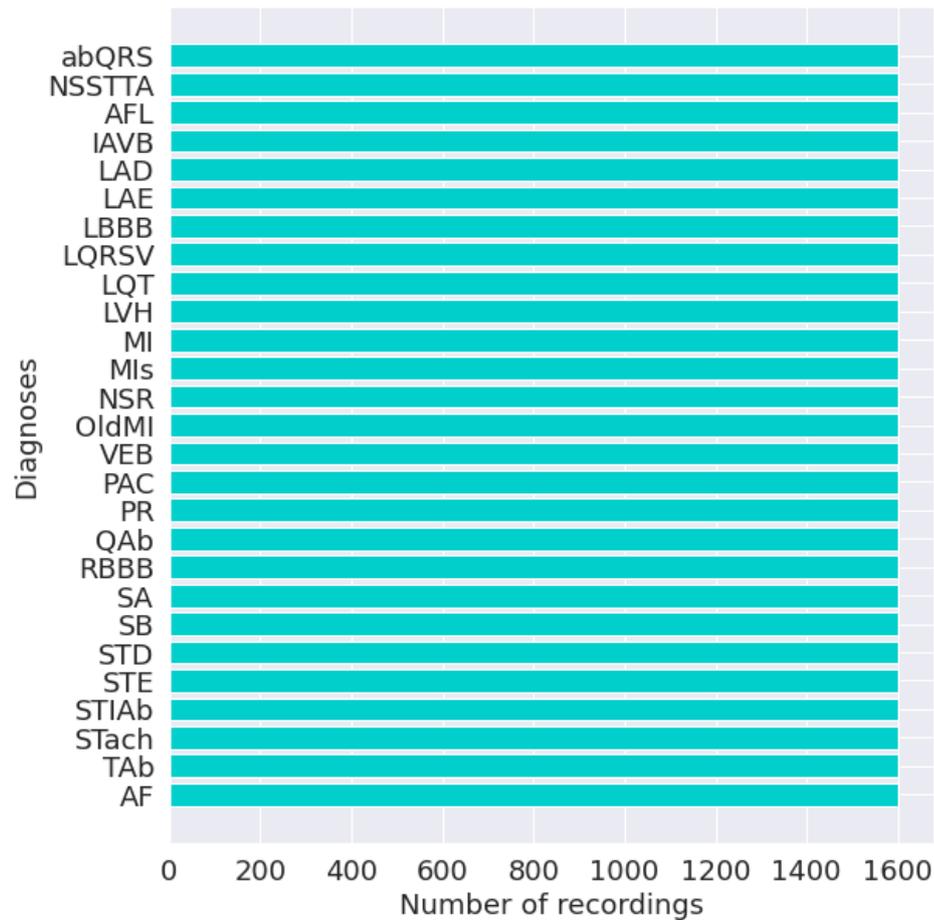


Figure 5.8 - Number of recordings for each diagnosis for the SMOTE oversampled data

### 5.1.6 Fitted models and results

With the previous pre-processing applied over the data, the following step was to adjust some Machine Learning models to the ECG's arrhythmias. During this step also other scenarios and considerations were employed [60]. A detailed explanation of the outlines is discussed in Table 5.1.

| <b>Characteristic</b>  | <b>Scenarios</b>  | <b>Best approach</b>                           |
|------------------------|---|--|
| Data split             | %Train-%Validation-%Test:<br>Option 1: 60%-20%-20%<br>Option 2: 70%-10%-10%<br>Option 3: 80%-10%-10%<br>Option 4: 90%-5%-5% | Option 4: 90%-5%-5%                            |
| Features normalization | Option 1: MinMaxScaler<br>Option 2: StandardScaler<br>Option 3: RobustScaler  | Option 3: RobustScaler                         |
| Sampling rate          | Option 1: 257Hz<br>Option 2: 500Hz  | Option 1: 257Hz                                |
| Features employed      | Option 1: Baseline features<br>Option 2: Baseline features + Spectral features  | Option 2: Baseline features + Spectral feature |

*Table 5.1 - Scenarios tried during modelling phase*

In the previous table are depicted the best approaches that managed to get the best performances (in terms of Accuracy, Recall, Precision, F1-Score and Overall Score). Moreover, in Figure 5.9 is shown the detailed behaviour of each one of the algorithms employed.

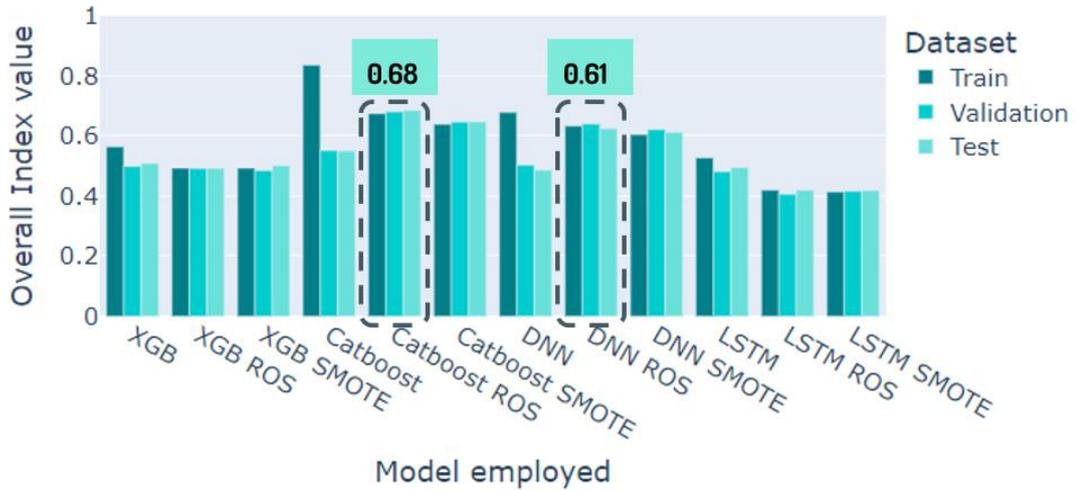


Figure 5.9 – Overall Index for methods employed

Within the previous results it is easy to realize that the best performing model is the one that applies Catboost over the ROS data. The latter obtained an overall index close to 0.68. Nevertheless, the Deep Neural Network applied over the ROS data has a close behaviour, having the mentioned metric in 0.61. Finally, LSTM and XG-Boost does not perform that well compared to the other models. An additional point to mention is that both Catboost and DNN used over the ROS dataset does not show any sign of overfitting since the metrics for train, validation and test sets are almost the same.

A similar analysis can be derived from Figure 5.10 and Figure 5.11, where the Accuracy and F1-Score are shown. In terms of Accuracy, Catboost got 0.67 and DNN got 0.61, both applied over the ROS dataset. It is important to clarify that using Accuracy is not the best metrics in this dataset since the labels are heavily unbalanced. For that reason, it is better to use the F1-Score which is more robust to the imbalanced datasets. Then, in terms of F1-Score, Catboost obtained a 0.67 and DNN 0.62.



Figure 5.10 - Accuracy for methods employed



Figure 5.11 - F1-Score for methods employed

Finally, a measurement of the time taken to train for each model was included. As shown in Figure 5.12

| Method         | Accuracy | F1-Score | Overall index | Mins |
|----------------|----------|----------|---------------|------|
| XG-Boost       | 0.52     | 0.49     | 0.50          | 9.4  |
| XG-Boost ROS   | 0.45     | 0.47     | 0.49          | 15.7 |
| XG-Boost SMOTE | 0.48     | 0.49     | 0.50          | 15.4 |
| CatBoost       | 0.55     | 0.53     | 0.55          | 34.5 |
| CatBoost ROS   | 0.67     | 0.67     | 0.68          | 36.4 |

|                |      |      |      |      |
|----------------|------|------|------|------|
| CatBoost SMOTE | 0.64 | 0.64 | 0.64 | 35.5 |
| DNN            | 0.50 | 0.47 | 0.48 | 2.8  |
| DNN ROS        | 0.61 | 0.62 | 0.61 | 3.8  |
| DNN SMOTE      | 0.60 | 0.60 | 0.61 | 4.3  |
| LSTM           | 0.51 | 0.39 | 0.49 | 4.3  |
| LSTM ROS       | 0.38 | 0.39 | 0.41 | 4.9  |
| LSTM SMOTE     | 0.39 | 0.39 | 0.41 | 2.7  |

, Catboost took almost 35 minutes to run. On the other hand, DNN and LSTM are algorithms that employed few time in training (close to 4 minutes in average). Finally, Catboost is the slowest method, although it generates the best results. In counter position, DNN is the fastest method, and the performance is not quite different from Catboost.

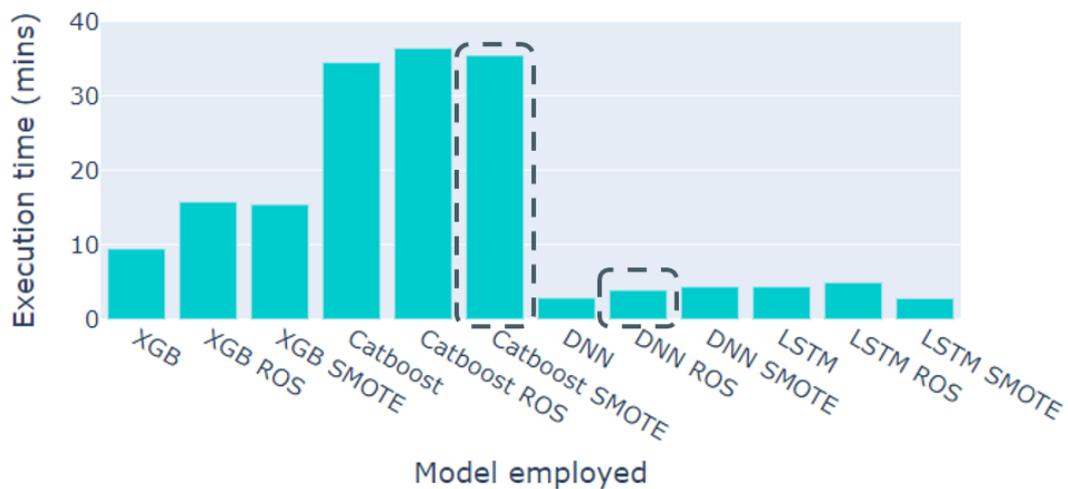


Figure 5.12 - Execution times for the methods used

| Method         | Accuracy | F1-Score | Overall index | Mins |
|----------------|----------|----------|---------------|------|
| XG-Boost       | 0.52     | 0.49     | 0.50          | 9.4  |
| XG-Boost ROS   | 0.45     | 0.47     | 0.49          | 15.7 |
| XG-Boost SMOTE | 0.48     | 0.49     | 0.50          | 15.4 |
| CatBoost       | 0.55     | 0.53     | 0.55          | 34.5 |

|                |      |      |      |      |
|----------------|------|------|------|------|
| CatBoost ROS   | 0.67 | 0.67 | 0.68 | 36.4 |
| CatBoost SMOTE | 0.64 | 0.64 | 0.64 | 35.5 |
| DNN            | 0.50 | 0.47 | 0.48 | 2.8  |
| DNN ROS        | 0.61 | 0.62 | 0.61 | 3.8  |
| DNN SMOTE      | 0.60 | 0.60 | 0.61 | 4.3  |
| LSTM           | 0.51 | 0.39 | 0.49 | 4.3  |
| LSTM ROS       | 0.38 | 0.39 | 0.41 | 4.9  |
| LSTM SMOTE     | 0.39 | 0.39 | 0.41 | 2.7  |

*Table 5.2 – Detailed metrics for the methods used*

## 5.2 Deploy on a microcontroller

In this section is described the process of uploading the trained model inside a low-power processor.

### 5.2.1 ESP-EYE board

The board used is the Espressif ESP-EYE, a 41 x 21 mm development board for image recognition and audio processing, which can be used in various Artificial Intelligence of Things (AIoT) applications. It has a microphone, a 2-Megapixel camera, and an ESP32 chip. With a 4 Mbyte flash and an 8 Mbyte PSRAM, ESP-EYE has plenty of storage. Additionally, it allows debugging via a Micro-USB connector and image transmission via Wi-Fi [61].

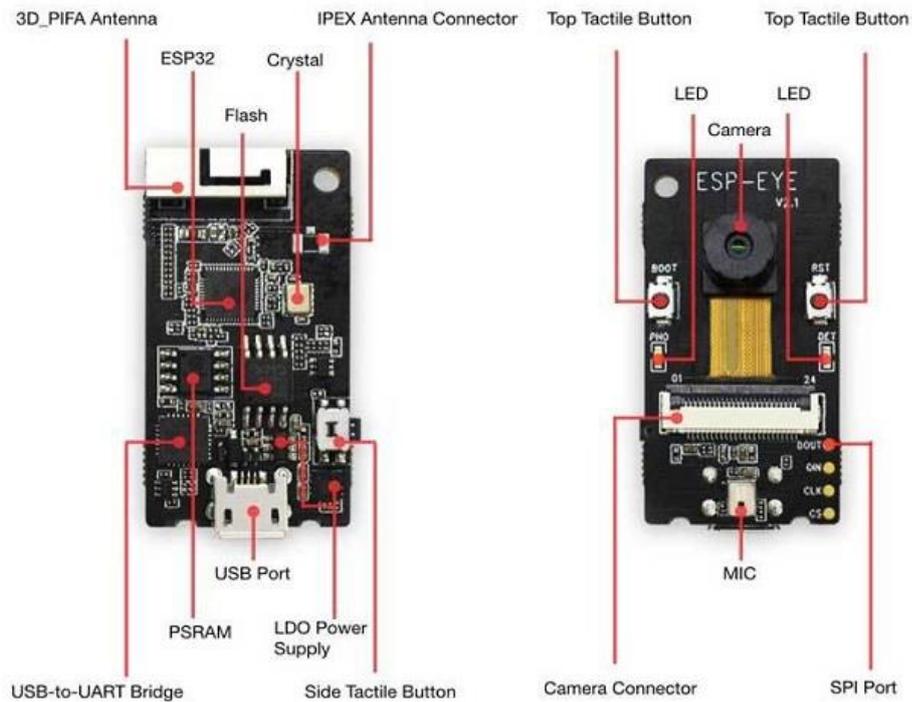


Figure 5.13 - Back view and front view of ESP-EYE board

## 5.2.2 Model conversion

The DNN ROS approach has the best performance numbers when F1-Score and execution time are considered, so I opted to move forward with my experiment utilizing this model.

First, I had to convert the trained model into a TensorFlow Lite (tflite) model in order to deploy the prediction model inside the board. To achieve this, I first used *keras* to save my model, and then I used the `tf.lite.TFLiteConverter.from_keras_model` function to create a converter that transformed the keras model into a tflite model.

I experimented a bit during the converting process because there are several parameters that may be supplied to the converter to determine optimization techniques.

Firstly, I realized that converting the model in a default way (without optimization) didn't work in my situation because the size of the converted model was too large with respect to the ESP-EYE's capacity memory.

Therefore, I chose a strategy that takes model optimization into account.

Quantization is the optimization technique that best meets my purposes because it always has the potential to reduce the size of a model while potentially losing some accuracy. Instead, pruning and clustering methods are not good since they are used to reduce the size of the model for download, making it easier to compress. Moreover, by making the mathematics involved in inference simpler, quantization can decrease latency, again potentially at the expense of some accuracy.

However, microcontrollers do not support hybrid models so a full quantization using a representative dataset was necessary.

Although full-integer quantization provides improved model size and latency, the quantized model won't always work as expected. It's usually expected for the model quality to be slightly lower than the original float model. However, there are cases where the model quality can go below the expectation or generated completely wrong results.

I tried to convert the model using full integer quantization, but it gave me totally incorrect results. Consequently, I chose to use float 32 operations, with the representative datasets (see Figure 4.8) for quantization, and the results were only somewhat worse than those of the original DNN ROS model (Table 5.3).

| <b>Method</b>                     | <b>Accuracy</b> | <b>Precision</b> | <b>Recall</b> | <b>F1-Score</b> | <b>Overall index</b> |
|-----------------------------------|-----------------|------------------|---------------|-----------------|----------------------|
| DNN ROS full integer quantization | 0.061           | 0.003            | 0.061         | 0.007           | 0.033                |
| DNN ROS float 32 quantization     | 0.590           | 0.640            | 0.590         | 0.597           | 0.604                |
| DNN ROS original                  | 0.614           | 0.664            | 0.614         | 0.619           | 0.628                |

*Table 5.3 – Metrics of the methods on test dataset*

Once I had the tflite model was important to check both the size and the operators used, since, as we know, TensorFlow Lite has some limitations. I utilized the open source Netron [62] viewer to look at model and detect the operators used (Figure 5.14), and then, I checked if all of them are in the list of compatible operators offered by TensorFlow. To see the complete list of operators it is possible to read the GitHub cited in [63].

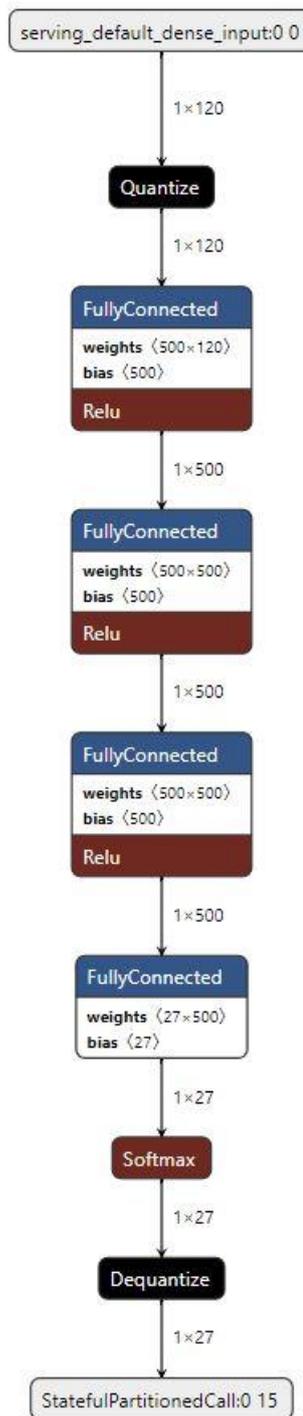


Figure 5.14 - Netron DNN ROS quantized model

### 5.2.3 Building, flashing, and monitoring

I created the project using Espressif IoT Development Framework (ESP-IDF), the official development framework for the ESP32, ESP32-S and ESP32-C Series SoCs, specifically I downloaded the extension available inside Visual Studio Code.

Now, I had to convert the DNN ROS model from a .tflite file to a .h header file in order to include it in the C program. The model's size rose following the conversion, but it was still within the bounds of the available memory. Then, using the TensorFlow Lite library I created a function that executes the model taking as input the ECG features and returns the code associated with the predicted diagnosis.

So, I connected the ESP-EYE board via USB to my computer, I selected the specific serial port inside ESP-IDF, and finally, after having built the code (Figure 5.15), I flashed the firmware inside the board (Figure 5.16) and looked at the results (Figure 5.17).

```
> Executing task: C:\Users\lore1\.espressif\python_env\idf4.4_py3.8_env\Scripts\python.exe C:\Users\lore1\esp\esp-idf\tools\idf_size.py c:\Users\lore1\Documents\espidf_projects\esp-idf-26\build\ecg_classifier.map <

Total sizes:
Used static DRAM: 18616 bytes ( 162120 remain, 10.3% used)
  .data size: 13856 bytes
  .bss size: 4760 bytes
Used static IRAM: 45574 bytes ( 85498 remain, 34.8% used)
  .text size: 44547 bytes
  .vectors size: 1027 bytes
Used Flash size : 747899 bytes
  .text : 123703 bytes
  .rodata : 623940 bytes
Total image size: 807329 bytes (.bin may be padded larger)
```

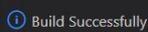


Figure 5.15 - Build successful log

```
Flash will be erased from 0x00008000 to 0x00008fff...
Compressed 807536 bytes to 583889...
Wrote 807536 bytes (583889 compressed) at 0x00010000 in 13.8 seconds (effective 469.6 kbit/s)...
Hash of data verified.
Flash params set to 0x0220
Compressed 25280 bytes to 15807...
Wrote 25280 bytes (15807 compressed) at 0x00010000 in 0.8 seconds (effective 239.2 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 105...
Wrote 3072 bytes (105 compressed) at 0x00008000 in 0.1 seconds (effective 301.7 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Flash Done ⚡
```

Figure 5.16 - Flash log

```
AllocateTensors() success
input_tensor size: 2
input_tensor first dim size: 1
input_tensor second dim size: 120
input_tensor third dim size: 29
output_tensor size: 2
output_tensor first dim size: 1
output_tensor second dim size: 27
For input: 0, predicted: 9 , correct: 9
For input: 1, predicted: 21 , correct: 21
For input: 2, predicted: 21 , correct: 0
For input: 3, predicted: 23 , correct: 23
For input: 4, predicted: 21 , correct: 13
For input: 5, predicted: 9 , correct: 9
For input: 6, predicted: 21 , correct: 21
For input: 7, predicted: 20 , correct: 20
For input: 8, predicted: 2 , correct: 8
For input: 9, predicted: 25 , correct: 11
```

Figure 5.17 - Output log

## Conclusions

When compared to the state-of-the-art methods in the literature on high-end processor architecture, the results show that arrhythmia detection within low-power devices can make predictions with a respectable high level of precision.

The COVID-19 pandemic has dramatically decreased the number of patients visiting hospitals and clinics, causing doctors and clinics to look towards online telemedicine and home monitoring alternatives. The monitoring of cardiac health problems is such an example and can be managed with the use of wearable cardiac monitoring equipment [64]. Moreover, during the quarantine period, when COVID-19 patients are residing at home, wearable technologies can help by enabling homecare monitoring so the development of such smart wearables is really important, and this is made even more so because it has been consistently shown that the presence of pre-existing cardiovascular disease and/or development of acute cardiac injury are associated with significantly worse outcome in these patients [65].

Future investigations in this area will involve the improvement of the F1-Score and Accuracy of the classification methods. To this end, may be interesting to add other morphological features related to all the twelve leads. It could be also important to evaluate the performance of the proposed technique using different datasets, and to investigate an intra-patient training methodology. Finally, another advancement will be to connect the microprocessor to a wearable device capable of revealing heart activity parameters, allowing for real-time arrhythmia classification.

## References

- [1] World Health Organization, "Cardiovascular diseases," [Online]. Available: <https://www.who.int/health-topics/cardiovascular-diseases>. [Accessed 01 07 2022].
- [2] P. Iaizzo, Handbook of cardiac anatomy, physiology, and devices, Springer International Publishing, 2015.
- [3] M. Sampson and A. McGrath, "Understanding the ECG. Part 1: Anatomy and physiology," *British Journal of Cardiac Nursing*, vol. 10, no. 11, pp. 548--554, 2015.
- [4] M. Sampson and A. McGrath, "Understanding the ECG Part 2: ECG basics," *British Journal of Cardiac Nursing*, vol. 10, no. 12, pp. 588--594, 2015.
- [5] G. Walraven, Basic Arrhythmias, Pearson Education, 2014.
- [6] Goldsworthy, Kleinpell and Williams, "International Best Practices in Critical Care," *Queensland: World Federation of Critical Care Nurses*, 2019.
- [7] V. Gudivada, A. Apon and J. Ding, "Data Quality Considerations for Big Data and Machine Learning: Going Beyond Data Cleaning and Transformations," *International Journal on Advances in Software*, vol 10 no 1 & 2, <http://www.iariajournals.org/software/>, 2017.
- [8] Forbes Press Releases Forbes Staff, "Poor-Quality Data Imposes Costs and Risks on Businesses, Says New Forbes Insights Report," *Forbes*, 2017.
- [9] TDWI, "The data warehousing institute," [Online]. Available: <https://tdwi.org/Home.aspx>.
- [10] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.," *Circulation [Online]*, vol. 101, no. 23, p. e215–e220, 2000.
- [11] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. vol. 20, no. no. 3, pp. pp. 45-50, May/June 2001.
- [12] N. J. Holter, "New methods for heart studies," *Science*, vol. vol. 134, p. p. 1214, 1961.

- [13] ECRI, "American Heart Association ECG Database USB," [Online]. Available: <https://www.ecri.org/american-heart-association-ecg-database-usb>.
- [14] PhysioNet, "AHA Database Sample Excluded Record," [Online]. Available: <https://physionet.org/content/ahadb/1.0.0/>.
- [15] A. Taddei, G. Distanto, M. Emdin, P. Pisani, G. B. Moody, C. Zeelenberg and C. Marchesi, "The European ST-T database: standard for evaluating systems for the analysis of ST-T changes in ambulatory electrocardiography," *European Heart Journal*, vol. 13, pp. 1164-1172, 1992.
- [16] A. Biagini, G. Distanto, M. Emdin, A. Taddei, M. G. Mazzei, P. Pisani, N. Roggero, M. Varanini, R. G. Mark, G. B. Moody, L. Braaksma, C. Zeelenberg and C. Marchesi, "The European ST-T database: development, distribution and use," *Proceedings Computers in Cardiology*, pp. 177-180, 1990.
- [17] PhysioNet, "Moody Challenge Overview," [Online]. Available: <https://physionet.org/about/challenge/moody-challenge-overview>.
- [18] E. A. Perez Alday, A. Gu, A. Shah, C. Robichaux, A.-K. Ian Wong, C. Liu, F. Liu, A. Bahrami Rad, A. Elola, S. Seyedi, Q. Li, A. Sharma, G. D. Clifford and M. A. Reyna, *Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020*, 2020.
- [19] S. Savalia and V. Emamian, "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks," *Bioengineering*, vol. 5, no. 2, p. 35, 2018.
- [20] A. Vishwa, M. K. Lal, S. Dixit and P. Vardwaj, "Clasification of arrhythmic ECG data using machine learning techniques," *IJIMAI*, vol. 1, no. 4, pp. 67--70, 2011.
- [21] F. Murat, O. Yildirim, M. Talo, U. B. Baloglu, Y. Demir and U. R. Acharya, "Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review," *Computers in biology and medicine*, vol. 120, p. 103726, 2020.
- [22] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Generation Computer Systems*, vol. 86, pp. 446--455, 2018.
- [23] A. Demonbreun and G. M. Mirsky, "Automated classification of electrocardiograms using wavelet analysis and deep learning," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [24] Z. He, P. Zhang, L. Xu, Z. Bai, H. Zhang, W. Li, P. Xia and X. Chen, "A Novel Convolutional Neural Network for Arrhythmia Detection From

- 12-lead Electrocardiograms," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [25] D. Borra, A. Andalò, S. Severi and C. Corsi, "On the application of convolutional neural networks for 12-lead ECG multi-label classification using datasets from multiple centers," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [26] A. Raza, K. P. Tran, L. Koehl and S. Li, "Designing ecg monitoring healthcare system with federated transfer learning and explainable ai," *Knowledge-Based Systems*, vol. 236, p. 107763, 2022.
- [27] V. Sujadevi and K. Soman, "Towards identifying most important leads for ECG classification. A Data driven approach employing Deep Learning," *Procedia Computer Science*, vol. 171, pp. 602--608, 2020.
- [28] M. Jiang, J. Gu, Y. Li, B. Wei, J. Zhang, Z. Wang and L. Xia, "HADLN: Hybrid attention-based deep learning network for automated arrhythmia classification," *Frontiers in Physiology*, vol. 12, 2021.
- [29] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij and J. Rubin, "A wide and deep transformer neural network for 12-lead ECG classification," *2020 Computing in Cardiology*, pp. 1--4, 2020.
- [30] Z. Zhao, H. Fang, S. D. Relton, R. Yan, Y. Liu, Z. Li, J. Qin and D. C. Wong, "daptive lead weighted resnet trained with different duration signals for classifying 12-lead ecgs," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [31] Z. Zhu, H. Wang, T. Zhao, Y. Guo, Z. Xu, Z. Liu, S. Liu, X. Lan, X. Sun and M. Feng, "Classification of cardiac abnormalities from ECG signals using SE-ResNet," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [32] S. Hong, W. Zhang, C. Sun, Y. Zhou and H. Li, "Practical Lessons on 12-Lead ECG Classification: Meta-Analysis of Methods From PhysioNet/Computing in Cardiology Challenge 2020," *Frontiers in Physiology*, p. 2505, 2022.
- [33] J. a. C. T. a. X. B. a. B. X. Chen, Y. Wang, H. Duan, W. Li, J. Zhang and X. Ma, "SE-ECGNet: multi-scale SE-Net for multi-lead ECG data," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [34] N. Fayyazifar, S. Ahderom, D. Suter, A. Maiorana and G. Dwivedi, "Impact of neural architecture design on cardiac abnormality classification using 12-lead ECG signals," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [35] S. Min, H.-S. Choi, H. Han, M. Seo, J.-K. Kim, J. Park, S. Jung, I.-Y. Oh, B. Lee and S. Yoon, "Bag of tricks for electrocardiogram classification

- with deep neural networks," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [36] M. N. Bos, R. R. van de Leur, J. F. Vranken, D. K. Gupta, P. van der Harst, P. A. Doevendans and R. van Es, "Automated comprehensive interpretation of 12-lead electrocardiograms using pre-trained exponentially dilated causal convolutional neural networks," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [37] C. Zisou, A. Sochopoulos and K. Kitsios, "Convolutional recurrent neural network and LightGBM ensemble model for 12-lead ECG classification," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [38] P.-Y. Hsu, P.-H. Hsu, T.-H. Lee and H.-L. Liu, "Multi-label Arrhythmia classification from 12-lead electrocardiograms," in *2020 Computing in Cardiology*, 2020, pp. 1--4.
- [39] T. Rattenbury, J. M. Hellerstein, J. Heer, S. Kandel and C. Carreras, *Principles of data wrangling: Practical techniques for data preparation*, O'Reilly Media, Inc., 2017.
- [40] R. Stefanski, "Data Wrangling in 6 Steps: An Analyst's Guide For Creating Useful Data," 21 September 2021. [Online]. Available: <https://hevodata.com/learn/data-wrangling/>. [Accessed 05 May 2022].
- [41] D. Zhang, "Interpretable Deep Learning for Automatic Diagnosis of 12-lead Electrocardiogram," 2021. [Online]. Available: <https://github.com/onlyzdd/ecg-diagnosis>. [Accessed 06 2022].
- [42] J. Surda, S. Lovas, J. Pucik and M. Jus, "Spectral Properties of ECG Signal," *2007 17th International Conference Radioelektronika*, pp. 1-5, 05 2007.
- [43] S. Talebi, "The Wavelet Transform," 12 2020. [Online]. Available: <https://towardsdatascience.com/the-wavelet-transform-e9cfa85d7b34>. [Accessed 06 2022].
- [44] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 06 2002.
- [45] A. Fernández, S. Garcia, F. Herrera and N. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863-905, 04 2018.
- [46] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.

- [47] C. Bentéjac, A. Csörgő and G. Martínez-Muñoz, "A Comparative Analysis of XGBoost," 2019.
- [48] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.
- [49] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," vol. 43, no. 1, pp. 3--31, 2000.
- [50] C. Gallo, "Artificial neural networks tutorial," in *Encyclopedia of Information Science and Technology, Third Edition*, IGI Global, 2015, pp. 6369--6378.
- [51] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Transactions on Circuits and Systems*, vol. 8, no. 7, pp. 579--588, 2009.
- [52] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735--1780, 1997.
- [53] G. Canbek, S. Sagiroglu, T. T. Temizel and N. Baykal, "Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights," in *2017 International Conference on Computer Science and Engineering (UBMK)*, IEEE, 2017, pp. 821--826.
- [54] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
- [55] TensorFlow, "TensorFlow Lite for Microcontrollers," [Online]. Available: <https://www.tensorflow.org/lite/microcontrollers>. [Accessed 01 06 2022].
- [56] TensorFlow, "TensorFlow," [Online]. Available: <https://www.tensorflow.org/>. [Accessed 01 06 2022].
- [57] Wikipedia, "TensorFlow - Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/TensorFlow>. [Accessed 01 06 2022].
- [58] TensorFlow, "Model optimization | TensorFlow Lite," [Online]. Available: [https://www.tensorflow.org/lite/performance/model\\_optimization](https://www.tensorflow.org/lite/performance/model_optimization). [Accessed 15 06 2022].
- [59] PhysioNet, "Classification of 12-lead ecgs: The physionet/computing in cardiology challenge 2020," [Online]. [Accessed 01 06 2022].

- [60] Q. H. Nguyen, H.-B. Ly, L. S. Ho, N. Al-Ansari, H. V. Le, V. Q. Tran, I. Prakash and B. T. Pham, "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Mathematical Problems in Engineering*, vol. 2021, 2021.
- [61] Espressif, "ESP-EYE AI Board I Espressif," [Online]. Available: <https://www.espressif.com/en/products/devkits/esp-eye/overview>. [Accessed 01 06 2022].
- [62] Netron, [Online]. Available: <https://netron.app/>.
- [63] [Online]. Available: [https://github.com/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/all\\_ops\\_resolver.cc](https://github.com/tensorflow/tflite-micro/blob/main/tensorflow/lite/micro/all_ops_resolver.cc). [Accessed 01 06 2022].
- [64] A. Catalani, I. Chatzigiannakis, A. Anagnostopoulos, G. Akrivopoulou, D. Amaxilatis and A. Antoniou, "Hardware-assisted and Deep-Learning techniques for Low-Power Detection of Cardiovascular Abnormalities in Smart Wearables," in *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2021, pp. 144--151.
- [65] M. Bansal, "Cardiovascular disease and COVID-19," *Diabetes & metabolic syndrome*, vol. 14, no. 3, pp. 247-250, 2020.