SAPIENZA
UNIVERSITÀ DI ROMA

Heartbeat detection and Arrhythmia classification from the ECG signal using Machine Learning techniques

Dipartimento di Informatica, Automazione e Gestionale

Corso di Laurea Magistrale in Master of Science in Engineering in Computer Science

Candidate

Alessandro Scirè
ID number 1770594

Thesis Advisor

Prof. Ioannis Chatzigiannakis

Co-Advisor

Prof. Aris Anagnostopoulos

Academic Year 2017/2018

Thesis defended on 16 Gennaio 2019
in front of a Board of Examiners composed by:

Prof. Alessandro De Luca (chairman)
Prof. Ioannis Chatzigiannakis
Prof. Roberto Beraldi
Prof. Giorgio Grisetti
Prof. Massimo Mecella
Prof. Umberto Nanni
Prof. Fabio Patrizi

Heartbeat detection and Arrhythmia classification from the ECG signal using Machine Learning techniques
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: January 6, 2019

Author's email: alescire94@gmail.com

Introduction

In the following document is described the usage of machine learning techniques for the purpose of **arrhythmia detection and classification** from the ECG signal. In the last 50 years, electrocardiogram (ECG) have been a powerful tool in the exploration and diagnostic of heartbeat diseases. This approach gained a lot of popularity due to the fact that the acquisition of data is not invasive for patients and requires simple and cheap devices. Its wide employement allowed the medical community to afford a huge amount of data to be analyzed.

The analysis of the ECG record is a **time consuming process** even for an expert cardiologist, since it is a repetitive procedure applied to a long sequence of heartbeats. Therefore, a lot of efforts have been employed in order to automate this process. Furthermore, arrhythmias can take place in a healty heart and be of minimal consequence but they may also indicate a serious problem that may lead to stroke or sudden cardiac death. For this reason, automatic arrhythmia detection is crucial in clinical cardiology especially when performed in **real time**, and that is the point in which the field of machine learning comes in hand. The problems of **Heartbeat detection** and **Arrhythmia classification** will be addressed in a **Supervised learning** framework: the analysis of ECG tracings annotated from expert cardiologists allows to extract useful knowledge applicable to real world scenarios.

ECG devices are capable of recording the heart's patterns are available only in hospitals and are bulky for usage in outside place. To overcome this problem, in the last years, wearable devices have been developed such as *Wahoo Fitness* and *Polar H10 Heart Rate Sensor*. Currently available portable ECG devices rely on 1-3 leads which have limited accuracy and reliability, as electromagnetic noise may be either of environmental (e.g., mobile phones, electrical wires, appliances) or internal origin (e.g., produced by muscular contractions) and remains a frequent source of abnormally appearing ECGs. This problem is addressed by introducing **more than one recording points** (three, six, twelve ECG leads) in order to acquire more information from more than one heart regions. Considering additional recording points **increase the power consumption** of the device and produce a high-definition signal that requires bigger memory capacity to be stored [1]. This creates a trade-off between accuracy and longevity of the wearable device. This problem lead to the necessity of an efficient approach based on 1-2 Leads without loosing accuracy for what concerns the R peak detection and the arrhythmia analysis. In this study we demonstrate that such an approach based on Supervised Learning is possible.

A **real time agorithm** based on KNN has been developed for ECG peak detection, we evaluated the performance with respect to the state of the art and, in addition, we measured the computational time on *Internet of Things (IoT)* devices such as *Raspberry Pi 3* and two *Android* smartphones. This algorithm relies on a single ECG lead, that means less battery consumption and memory usage, achieving good results in terms of beat detection accuracy. Furthermore, we demonstrated that they can operate in real-time on *IoT* devices which is a real improvement beacuse there is no need to store the entire sequence of ECG data.

Use Cases

An automated healthcare application [2] allows patients to share information to their physicians, monitor their health status independently and notify the authorities rapidly in emergency situations. Any advanced healthcare monitoring system must be used from both patients and doctors, offline or in real-time, and under all circumstances. In general, a comprehensive healthcare monitoring system should support patient self-monitoring, physician offline monitoring through accessing obtained datasets from various user devices, physician online monitoring and patient monitoring within healthcare infrastructures such as hospitals and day-care centers. This analysis leads to the accurate definition of four use cases:

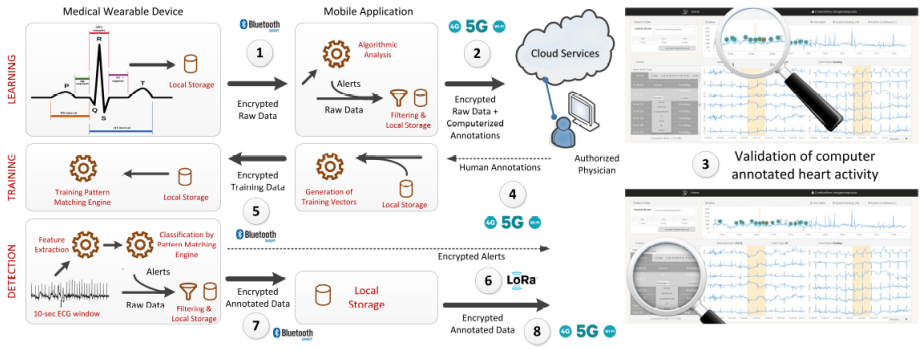
1. **Patient self-monitoring**: This use case refers to end-users that would like to perform self-monitoring of their medical conditions.

2. **Physician off-line monitoring**: This scenario targets doctors that will use healthcare products to monitor their patients either at their office or at a patients home during a visit. The patients carry the wearable device at home and then, after a period of time, return the device to their physician to acquire the traces and conduct the diagnosis.

3. **Physician on-line monitoring**: This particular use case is an extension of the previous one, where the patients use the device over an extended period of time and the physician can remotely monitor the acquired vital traces via the available cloud services.

4. **Patient monitoring within Healthcare Infrastructures**: Being the most advanced scenario, this can take place inside an ambulance, hospital or adult day-care center whereby the wearable devices, operated by a personal health assistant instantly acquire the traces of the patient and enable real-time monitoring of the data which are stored to the cloud services.

An example of such application is the *HEART* platform, that combines wearable embedded devices, mobile edge devices, and cloud services to provide on-the-spot, reliable, accurate, and instant monitoring of the heart. Initially, the wearable ECG goes through a learning phase in order to collect an adequate number of ECG recordings based on which we will train the pattern matching engine to match the needs of the particular patient. This is a necessary first step as the evaluation of and ECG depends on anthropometric data (body height and body weight) on



Information exchange during the three phases of Learning, Training and Detection, source:[1]

age and sex of a patient. During this phase, the wearable device is continuously storing the ECG recordings and periodically forwards them to the nearby edge device. This latter analyzes the received signal and forward it to the cloud services along with computerized annotations. The authorized physician may view reports, search traces and examine ECG alerts aggregated on the patient's health records remotely after they are synced with the cloud platform. The physician goes through the annotated recordings and validates or rejects the computerized interpretations depending on his expert assessment. When an adequate number of normal and abnormal sessions have been identified by the authorized physician, the system is ready to enter the training phase. The human-curated annotations are forwarded to the edge device where the corresponding sessions are extracted from the local storage, are analyzed to extract a carefully selected set of features. The features of each session along with the annotations constitute the training vectors for the pattern matching engine. When the training completes, the wearable device is ready to enter the detection phase. During the detection phase the signals collected from the ECG leads are analyzed and the features are extracted using the local processor. The resulting vector is passed to the pattern matching engine for classification. In case an abnormal event is detected, the wearable device is in a position to immediately notify the authorized physician via a short message exchange including only the alert type. As soon as the patient visits the physician, the complete ECG recordings are relayed to the nearby edge device. The ECG recordings are finally uploaded on the cloud platform where they become available to the authorized physician for examination and assessment. The above described cycle of learning, training and detecting is repeated periodically to re-evaluate the operation of the pattern matching of the wearable device and fine-tune its performance. A wearable device that is used for diagnosing and monitoring heart diseases needs to be capable of collecting high-quality ECG traces. During the learning phase, the wearable device is continuously storing the ECG traces on the internal memory. During this offline monitoring period, the device is expected to store a recorded session of 24 hours.

List of Contributions

1. **QRS KNN algorithm:** A **new algorithm** for *heartbeat detection* is pre-

sented, named *QRS KNN*, that exploits a different feature definition with respect to the one explained in [3].This new approach has the advantage of requiring a smaller amount of data for training, and resulted in performances($Precision = 0.988$, $Recall = 0.923$) comparable to the State of the Art approaches and lower computational time for both the training and classification steps with respect to the implementation of [3].

2. **KNN algorithm:** The procedure based on the KNN(K Nearest Neighbors) algorithm, described in [3], has been implemented in order to provide a baseline for comparison with the *QRS KNN*. Therefore, the algorithm performance has been evaluated and the computational time has been measured.

3. **First-order difference Algorithm:** The first-order difference approach is a naive algorithm designed to find the local maxima in a generic signal. The decision of implementing this approach was taken in order to have a comparison between ad-hoc methods for ECG analysis for heartbeat detection and a standard signal processing module.

4. **Recurrent Neural Network:** The heartbeat detection module aims to identify the heartbeats along the ECG tracing. The output of this module is then feed to the *Arrhythmia classification* stage. A Recurrent Neural Network has been designed and trained in order to discriminate normal heartbeats to the arrhythmic ones. The detected arrhythmic beats are then classified in more fine-grained classes, defined by a well-known standard for cardiac algorithm evaluation[4].

5. **Comparative Evaluation:** A comparative evaluation of the various approaches for both *heartbeat detection and arrhythmia classification* described in this study has been provided according to the correctness of the results and the computational time. The computational time has been measured executing the various algorithms on three different platforms: an Intel®Core™i3-3240 CPU @3.40GHz, an Arm Cortex-A53 CPU @2.1GHz and a Qualcomm SnapDragon 600 CPU @1.9 GHz.

Document Structure

- Chapter 1: introduction into the cardiological field, and the relationship between the ECG signal morphology and the cardiac cycle.
- Chapter 2: description of the approach followed for the heartbeat detection along an ECG record and comparisons with standard solutions.
- Chapter 3: arrhythmia classification procedure and description of the results.
- Chapter 4: Conclusions and future work

Acknowledgments

Ringrazio i miei genitori che da sempre mi sostengono e Sonia per essermi stata amorevolmente vicina in questo intenso e meraviglioso anno.

Contents

1 The Heartbeat Cycle: Arrhythmia interpretation on the ECG signal **1**
1.1 The electrical activity of the heart . . . 1
1.2 Heart conduction system . . . 1
1.3 Waves and measurements . . . 2
1.3.1 Monitoring leads . . . 3
1.3.2 Time and voltage measurements . . . 4
1.3.3 Cardiac cycle from the ECG perspective . . . 4
1.3.4 Artifacts . . . 6
1.3.5 Refractory Periods . . . 6
1.4 Basic Arrhythmias . . . 7
1.4.1 Sinus Rhythms . . . 7
1.4.2 Atrial Rhythms . . . 9
1.4.3 Ventricular Rhythms . . . 12
1.4.4 Junctional Rhythms . . . 14

2 Heartbeat Detection **19**
2.1 State of the art . . . 20
2.1.1 Artificial Neural Network . . . 20
2.1.2 Convolutional Neural Network . . . 21
2.1.3 Hidden Markov Models . . . 22
2.1.4 Support Vector Machine . . . 22
2.1.5 First-Order Difference . . . 23
2.1.6 Pan Tompkins . . . 24
2.2 K Nearest Neighbors approaches . . . 24
2.2.1 Signal Preprocessing: Filtering and Squaring Techniques . . . 25
2.2.2 SSK Algorithm . . . 26
2.2.2.1 Training . . . 27
2.2.2.2 Model Selection . . . 27
2.2.3 QRS KNN Algorithm . . . 29
2.2.3.1 Training . . . 29
2.2.3.2 Model Selection . . . 29
2.3 Evaluation . . . 30
2.3.1 Database . . . 31
2.3.1.1 Annotations . . . 31
2.4 Protocol Fine-Tuning . . . 31

2.4.1 First-Order Difference Algorithm 32
2.4.2 SSK Algorithm 32
2.4.2.1 Model Parameters 32
2.4.3 QRS KNN 32
2.4.3.1 Model Parameters 32
2.4.3.2 Window Size 33
2.4.3.3 Channels 33
2.4.3.4 Training size 35
2.5 Comparative Evaluation 35
2.5.1 Results Comparison 36
2.5.2 Computational Time Comparison 38
2.6 Critical Issues 39
2.6.1 Example 1: First Order Difference Algorithm 39
2.6.2 Example 2: QRS KNN Algorithm 39

3 Arrhythmia Classification 43
3.1 State of the Art 45
3.1.1 Reservoir Computing and Logistic Regression 45
3.1.2 Weighted SVM 45
3.1.3 Mixture of Experts 47
3.1.4 ECG Morphology and Heartbeat Interval Features 47
3.2 Recurrent Neural Network 48
3.3 Preprocessing 49
3.3.1 Input Preparation 49
3.3.2 Dataset Rebalancing 49
3.3.3 Standardization 50
3.4 Network Structure 50
3.4.1 Neural Networks Activation Functions 54
3.5 Training and Validation 56
3.6 Evaluation 56
3.6.1 Performance Measures 57
3.6.2 Database Annotations 58
3.7 Protocol Fine-Tuning 58
3.7.1 Window Size 59
3.7.2 Number of Channels 60
3.7.3 Augmentation and Reduction Factors 61
3.7.4 Timesteps 62
3.7.5 Network Layers 62
3.8 Results 64
3.8.1 Performance Comparison with State of The Art Algorithms . 65

4 Conclusions and Future Work 67

Chapter 1

The Heartbeat Cycle: Arrhythmia interpretation on the ECG signal

The electrical activity of the heart can be monitored by the ECG tracing, while the mechanical activity is evaluated by assessing pulses and blood pressure [5]. An ECG record is designed to give a graphic display of the electrical activity of the heart. The pattern displayed on the ECG is called **rhythm**. Therefore, the word **arrhythmia** refers to an abnormal heart rhythm.

1.1 The electrical activity of the heart

At the start of the cycle, when the heart is in a resting state, positive and negative electrical charges are balanced. This is called the **polarized** state, where no electrical flow is generated. A difference of potential between the inside of the heart and the outside is needed for the organ in order to receive the stimulus to start beating. When the charges inside and outside the earth trade places, the electricity flows in a wave-like motion throughout the heart. This wave is called **depolarization**, i.e. the process of the electrical discharge and flow of electrical activity. The process that follows the depolarization, which brings the heart ot the initial state, is called **repolarization**.

1.2 Heart conduction system

The electrical cells in the heart are all arranged in a system of pathways, called the **conduction system**. The study of this system is an essential part for the arrhythmia interpretation purpose. Normally, the electrical originates in the *Sinoatrial(SA) node* and travels to the ventricles by way of the *AV node*; 1.1 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 distribute the electrical impulses to muscle cells to stimulate contraction. These fibers are called *Purkinje fibers*.

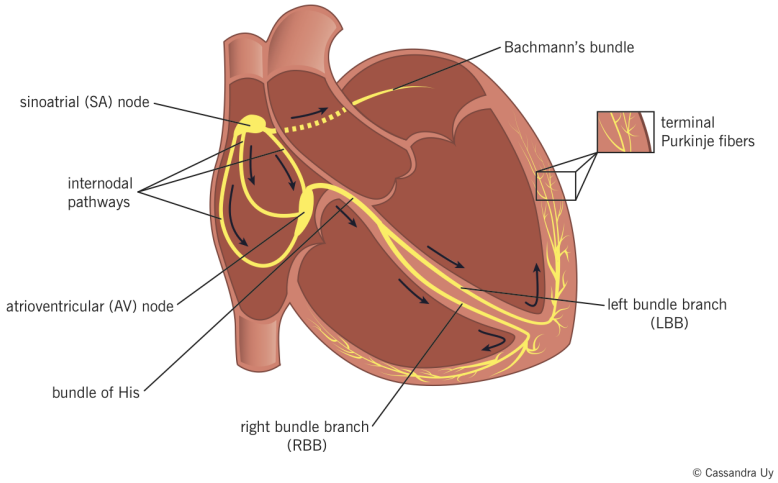


Figure 1.1. Electrical Conduction through the Heart, source:[5]

Each site has its own rate, called *inherent rate* at which it initiates impulses. The SA node, in normal conditions, has the greatest inherent rate; for this it is the normal **pacemaker** of the heart. If, however, a site becomes irritable and begins to discharge impulses at a faster than normal rate, it can override the SA node and take over the pacemaking function for the heart. This mechanism of an irritable site speeding up and taking over as pacemaker is called **irritability**. It is usually an undesirable occurrence, since it overrides the normal pacemaker and causes the heart to beat faster than it otherwise would. Something very different happens if the normal pacemaker slows down for some reason. If the SA node drops below its inherent rate, or if it fails entirely, the site with the next highest inherent rate will usually take over the pacemaking role. The next highest site is within the AV junction , so that site would become the pacemaker if the SA node should fail. This mechanism is called **escape** and is a safety feature that is built into the heart to protect it in case the normal fails.

1.3 Waves and measurements

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 a graph paper. There is a basic rule regarding the flow of electricity through the heart and out of the electrodes: if the electricity flows toward the positive electrode, the patterns produced on the graph paper will be upright. Conversely, if the electricity flows away from the positive electrode or passes through the negative one, the pattern will be a downward deflection. 1.2

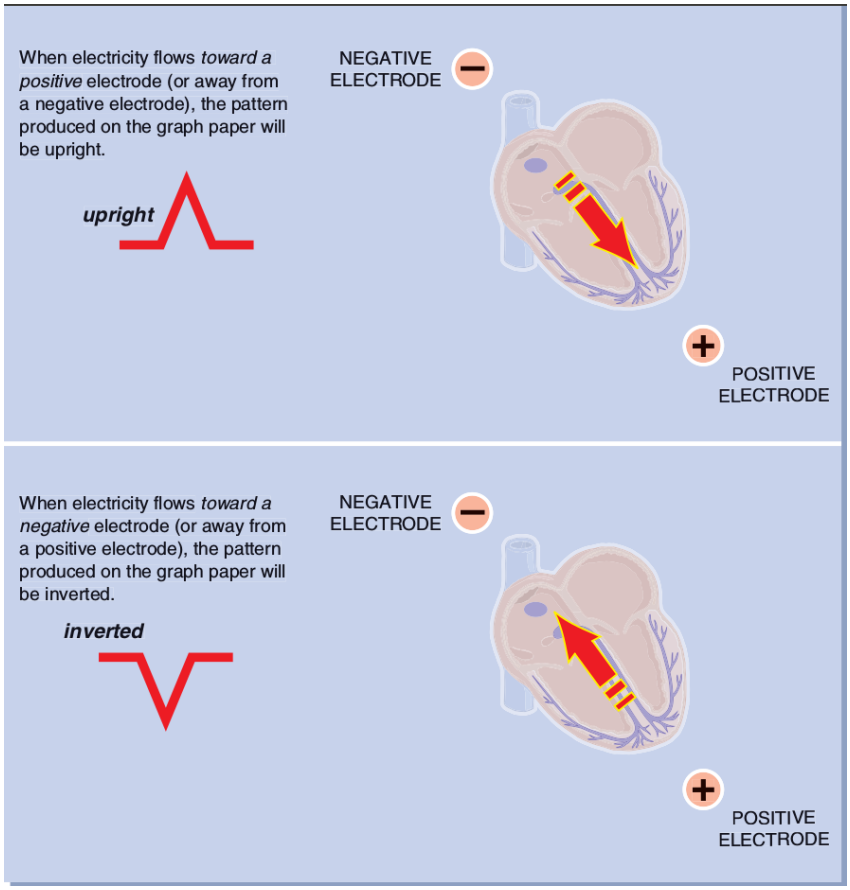


Figure 1.2. Rule of electrical flow, source:[5]

1.3.1 Monitoring leads

The positioning of an electrode for monitoring the ECG allows to see a single view of the heart's electrical pattern. Each view of the heart is called **lead**. For sophisticated ECG interpretation, many leads are inspected to visualize the entire heart, however for basic interpretations of arrhythmia only one single lead can be considered sufficient. Single lead that give good pictures of the basic waves are called *monitoring leads* beacuse they are used to monitor patterns such as arrhythmia. The first widely used monitoring lead was *Lead II*, but now it is common to use other leads as well, especially variations of the chest leads(MCL_1 for instance). Figure 1.3 shows the placement of electrodes to monitor Lead II. Note that the positive electrode is at the apex of the heart, and the negative electrode is below the right clavicle.The third electrode is a ground electrode and does not measure electrical flow in this lead. A widely used configuration of electrodes is one composed of 10 electrodes: one of the electrodes is positioned on the left arm (LA), on on the right arm *(RA)*, one on the left leg *(LL)*, one on the right leg *(LL)* and six on the chest*(V1 to V6)*, allowing a formation of 12 leads[6]. The 10 electrodes (12 lead) configuration can be seen in 1.4

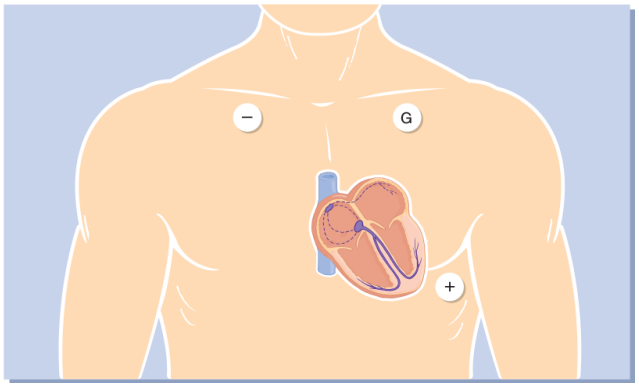


Figure 1.3. Electrode Placement for Monitoring Lead II, , source:[5]

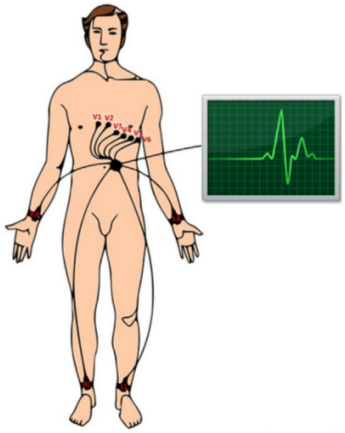


Figure 1.4. Typical 10 electrodes configuration, source:[6]

1.3.2 Time and voltage measurements

It is the strength of the current, or its voltage, that will determine the magnitude of the deflection. Therefore, the height of the deflection will indicate the amplitude of the electrical charge that produced the deflection. The second, and more important thing that the graph paper can provide is the determination of time. The vertical lines in the graph paper can tell how much time it took to the electrical current within the heart to travel from one area to another. This information is one of the most important for identifying an arrhythmia.

1.3.3 Cardiac cycle from the ECG perspective

The heart has four chambers: the upper two are the atria and the lower two are the ventricles. Atria and ventricles both operates as a single unit. In the normal heart, blood enters both atria simultaneously and then is forced into both ventricles simultaneously as the atria contract. All of this is coordinated so that the atria fill while the ventricles contract and viceversa. During each phase of the cardiac cycle, a distinct pattern is produced on the ECG graph paper. By learning to recognize these wave patterns and the cardiac activity they represent, we can study the relationships

between the different areas of the heart and begin to understand what is taking place within the heart at any given time.

In the ECG, each of the heartbeat phases is displayed by a specific wave pattern. A single cardiac cycle is expected to produce one heartbeat. In labeling the activity on the graph paper, the deflection above or below the median (a.k.a. *isoelectric*) line are called **waves**. In a single cardiac cycle there are five prominent waves, and each is labeled with a letter. In Figure 1.5 is it possible to determine the **P, Q, R, S** and **T** waves. An **interval** refers to the area between waves, and a **segment** identifies a

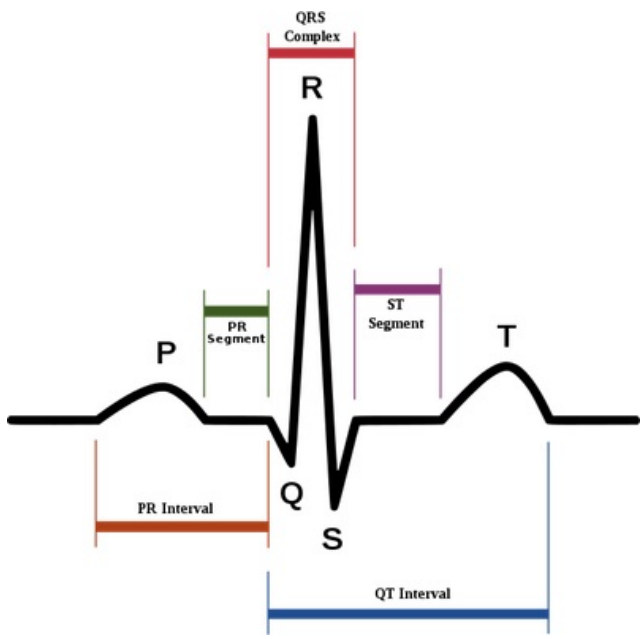


Figure 1.5. The ECG complex

straight line or area of electrical inactivity between waves.

The first identifiable wave of the cardiac cycle is the *P wave*. The *P wave* starts with the first deflection from the isoelectric line. It is indicative of atrial depolarization.

As the impulse leaves the atria and travels to the *AV node*, it encounters a slight delay. The tissues of the node do not conduct impulses as fast as other cardiac electrical tissues. This means that the wave of depolarization will take a longer time to get through the *AV node* than it would be in other parts of the heart. On the ECG, this is translated into a short period of electrical inactivity called the **PR segment**.

The subsequent wave is the **QRS complex**. The ventricular depolarization is shown by the ECG by a large complex of three waves: the Q, the R, and the S. This wave is significantly larger than the P wave beacuse ventricular depolarization involves a greater muscle mass than atrial depolarization. The Q wave is defined as the first negative deflection following the P wave but before the R wave. The Q wave

flows immediately into the R wave, which is the first positive deflection following the P wave.

Next comes the S wave, which is defined as the second negative deflection following the P wave, or the first negative deflection following the R wave.

After the ventricles depolarize, they begin their repolarization phase, which results in another wave in the ECG. The T wave is indicative of ventricular repolarization. The atria also repolarize, but this event is not significant enough to show up in the ECG.

Between the S and the T wave is a section called the **ST segment**. The ST segment is the flat, isoelectric section of the ECG between the end of the S wave and the beginning of the T wave. It represents the interval between ventricular depolarisation and repolarisation. The most important cause of ST segment abnormality (elevation or depression) is myocardial ischaemia / infarction.

1.3.4 Artifacts

The complexes on an ECG tracing are created by electrical activity within the heart. But it is possible for things other than cardiac activity to interfere with the tracing under analysis. Some common causes of interference, or, **artifact** are:

- Muscle tremors
- Patient movement
- Loose electrodes
- The effect of other electrical equipment in the room.

All these artifacts may interfere with the arrhythmia interpretation, because they may cause deflections on the signal.

1.3.5 Refractory Periods

Since depolarization takes place when the electrical charges begin their wave of movement by exchanging places across the cell membrane, it would follow that this process cannot take place unless the charges are in their original position. This means that the cell cannot *depolarize* until the *repolarization* process is complete. When the charges are depolarized and have not yet returned to their polarized state, the cell is said to be electrically **refractory** because it cannot yet accept another impulse.

On the ECG, the refractory period of the ventricles is when they are depolarizing or repolarizing. Thus, the QRS and the T wave on the EKG would be considered the refractory period of the cardiac cycle, since it signifies a period when the heart would be unable to respond to an impulse. Sometimes an electrical impulse will try to discharge the cell before repolarization is fully complete. In most cases nothing will happen because the cells are not back to their original position and therefore

cannot depolarize. But once in a while, if the stimulus is strong enough, an impulse might find several of the charges in the right position and thus discharge them before the rest of the cell is ready. This results in *abnormal depolarization* and hence is an undesirable occurrence. Thus, there is a small part of the refractory period that is not absolutely refractory. This small section is called the *relative refractory period* because some of the charges are polarized and thus can be depolarized if the impulse is strong enough.

1.4 Basic Arrhythmias

Arrhythmia analysis is a quite complex task since not only does every person on earth have his or her own individual ECG, distinct from all others, but one person's ECG can look very different from one moment to the next. It is inadequate to memorize some of the most common ECG patterns and trying to recognize them in the future. This type of ECG analysis is called *pattern recognition* and is a common but accidental way to approach arrhythmias. A much more reliable way to approach an EKG tracing is to take it apart, wave by wave, and interpret exactly what's happening within the heart.

Arrhythmias can be categorized into groups according to which pacemaker site initiates the rhythm. The most common sites, and thus the major categories of arrhythmias, are:

- Sinus
- Atrial
- Junctional
- Ventricular

The most common cardiac rhythm is sinus in origin, because the sinus(SA) node is the usual pacemaker of the heart (Section 1.2). Therefore, a normal, healthy heart would be in **Normal Sinus Rhythm (NSR)** because the rhythm originated in the SA node.

1.4.1 Sinus Rhythms

This category comprehends the rhythms originating in the Sinus node (SA) (Section 1.2). This group includes:

- Normal Sinus Rhythm
- Sinus Bradychardia
- Sinus Tachycardia
- Sinus Arrhythmia

Technically speaking, the **Normal Sinus Rhythm** is not an arrhythmia beacause it is a normal, rhythmic pattern. In a Normal Sinus Rhythm 1.6 the pacemaker impulse originates in the sinus node and travels through the normal conduction pathways within normal time frames. The P waves will be uniform, and since conduction is normal, one P wave will be in front of every QRS complex. Since the SA node

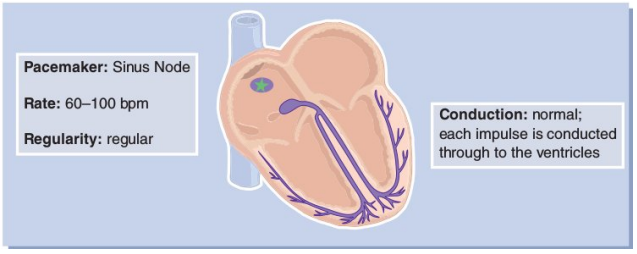


Figure 1.6. Mechanism of the Normal Sinus Rhythm, source:[5]

inherently fires at a rate of 60-100 times per minute, a Normal Sinus Rhythm must, by definition, fall within this rate range.

If a rhythm has all the characteristics of a NSR except fot the rate, that is lower than 60 bpm, it is called a **bradychardia**(Fig. 1.7), meaning *slow heart*.

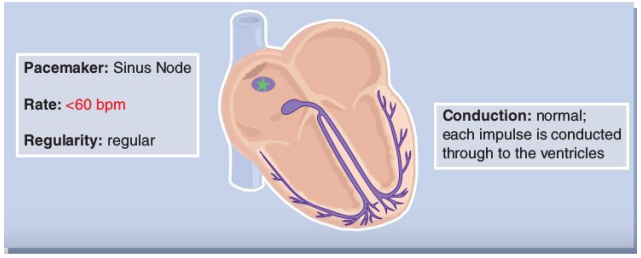


Figure 1.7. Mechanism of Sinus Bradychardia, source:[5]

The same thing is true for a rhythm that fits all of the rules for NSR except that the rate is too fast. When the heart beats too fast, it is called **tachycardia**, meaning *fast heart*. So a rhythm that originates in the sinus node and fits all rules for NSR except that the rate is too fast will be called Sinus Tachycardia(Fig. 1.8).

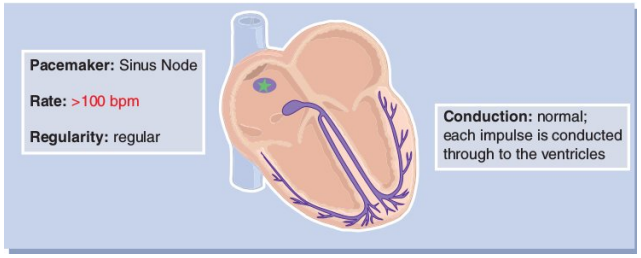


Figure 1.8. Mechanism of Sinus Tachychardia, source:[5]

Finally, The **Sinus Arrhythmia**(Fig.1.9) is characterized by a pattern that would normally be considered NSR, except that the rate changes with the patient's respirations. When the patient breathes in, the rate increases, and when he or she breathes out, the rate slows. This causes the R-R interval to be irregular across the strip. The result is a pattern with an upright P wave in front of every QRS complex, a normal and constant PRI, a normal QRS complex, but an irregular R-R interval. The difference between NSR and Sinus Arrhythmia is that NSR is regular and Sinus Arrhythmia is irregular.

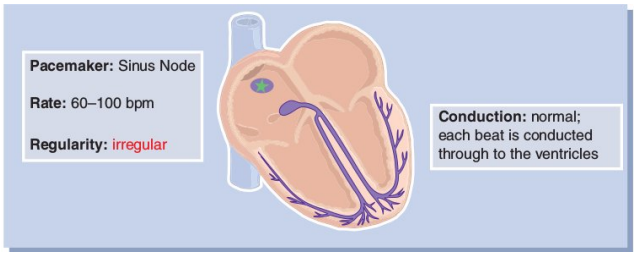


Figure 1.9. Mechanism of Sinus Arrhythmia, source:[5]

1.4.2 Atrial Rhythms

Sometimes the sinus node loses its pacemaking role, and this function is taken over by another site along the conductive system. The site with the fastest inherent rate usually controls the pacemaking function, as explained in Section 1.2. Since the atria have the next highest rate after the SA node, it is common for the atria to take over from the SA node. Rhythms that originate in the atria are called **atrial arrhythmias**. Atrial arrhythmias are caused when the atrial rate becomes faster than the sinus rate, and an impulse from somewhere along the atrial conduction pathways is able to override the SA node and stimulate depolarization. As with a sinus rhythm, an impulse that originates in the atria will travel through the atria to the AV junction anf then through the ventricular conduction pathways to the Purkinje fibers. The only difference is in the atria, where the conduction will be a little slower and rougher than it is with sinus rhythms.

Since atrial depolarization is is seen on the ECG as a P wave, it is expected to be of unusual shape. It can be flattened, notched, peaked, sawtoothed or even *diphasic*(meaning that it goes first above the isoelectric line and then dips below it.)

The first presented atrial arrhythmia is called **Wandering Pacemaker** (Figure 1.10). Wandering Pacemaker is caused when the pacemaker role switches from beat to beat from the SA node to the atria and back again. The result is a rhythm made up of interspersed sinus and atrial beats. The sinus beats are preceded by nice, rounded P waves, but the P wave changes as the pacemaker drops to the atria. The P waves of the atrial beats are not consistent and can be any variety of atrial configuration (e.g., flattened, notched, diphasic). In Wandering Pacemaker, the rhythm is usually slightly irregular.

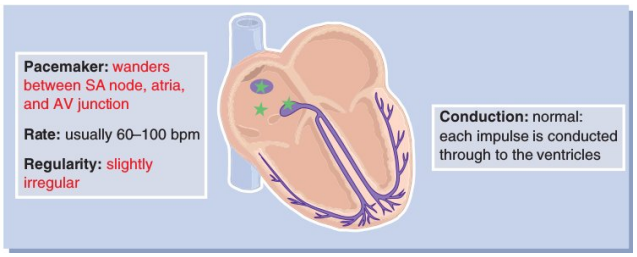


Figure 1.10. Mechanism of Wandering Pacemaker, source:[5]

When a single beat arises from an ectopic focus (a site outside of the SA node) within the conduction system, that beat is called **ectopic** beat. When an ectopic beat originates in the atria, it is called and atrial ectopic. An ectopic beat arises when a site somewhere along the conduction system becomes irritable and overrides the SA node for a single beat.

An atrial ectopic that is caused by irritability is called a **Premature Atrial Complex**(PAC)(Figure 1.11). A PAC is an ectopic beat that comes early in the cardiac cycle and originates in the atria.

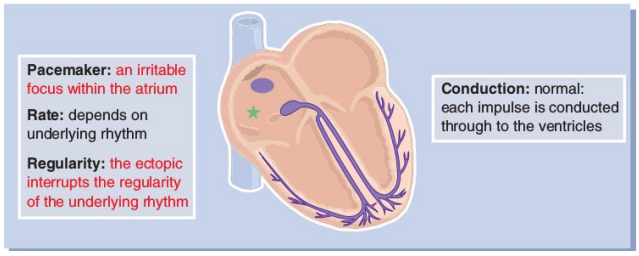


Figure 1.11. Mechanism of Premature Atrial Complex, source:[5]

A rhythm with PACs will be irregular beacuse the ectopics come permaturely and interrupt the underlying rhythm. Because PACs originate in the atria, they will have a characteristic atrial P wave that differs in morphology from the sinus P wave.

It is also possible for a single focus in the atria to become so irritable that it begins to fire very regularly and thus overrides the SA node for the entire rhythm. This arrhythmia is called **Atrial Tachycardia (AT)** (Figure 1.12) Atrial Tachycardia

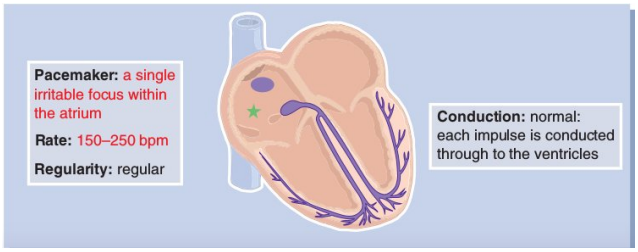


Figure 1.12. Mechanism of Atrial Tachycardia, source:[5]

will have all of the charachteristics of a PAC, except that is an entire rhythm instead of a single beat. All of the P waves in AT will have an atrial configuration. Atrial Tachycardia is a charachteristically very regular arrhythmia. It is usually very rapid, with a rate range between 150 and 250 bpm.

When the atria become so irritable that they fire faster than 250 bpm, they are said to be *fluttering.*This rhythm is called **Atrial Flutter**. In this case the atrial rate is usually in the range of 250-350 bpm.

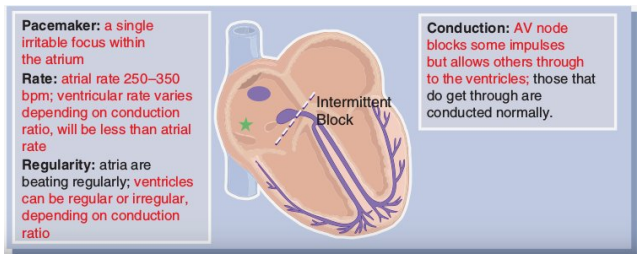


Figure 1.13. Mechanism of Atrial Flutter, source:[5]

The problem with a heart rate this rapid is that the ventricles don't have enough time to fill with blood between each beat. The result is that the ventricles will continue to pump but they won't be ejecting adequate blood volume to meet body needs. The heart has a built-in protective mechanism to prevent this from happening: the AV node. The AV node is responsible for preventing excess impulses from reaching the ventricles. This blocking action allow the ventricles to have time to fill with blood before they have to contract. This will be seen on the ECG as a very rapid series of P waves(Flutter Waves), but not every one followed by a QRS complex.

The last atrial arrhythmia in this study is the **Atrial Fibrillation**(Figure 1.14) This rhythm result when the atria become so irritable that they are no longer beating but are merely quivering ineffectively. This quivering ineffectively is called **fibrillation**. On the ECG tracing it is seen as a series of indescirnible waves along

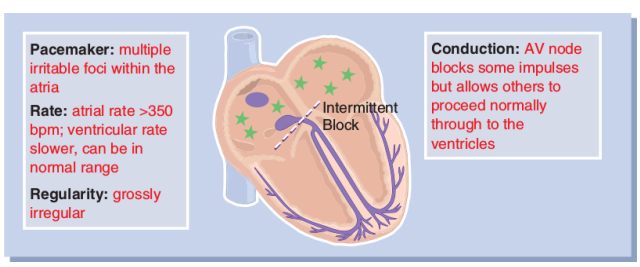


Figure 1.14. Mechanism of Atrial Fibrillation, source:[5]

the isoelectric line. In Atrial Fibrillation there are no discirnible P waves. The fibrillatory waves characteristic of Atrial fibrillation are called *f waves*. The rhythm is grossly irregular because the fibrillatory waves are conducted in a very chaotic way.

1.4.3 Ventricular Rhythms

Ventricular Arrhythmias are very serious for several reasons. First, the heart was intended to depolarize from the top down. The atria were meant to contract before the ventricles in order to pump blood effectively. When an impulse originates in the ventricles, the process is reversed, and the heart's efficiency is greatly reduced. Furthermore, since the ventricles are the lowest site in the conduction system, there are no more fail-safe mechanisms to backup a ventricular arrhythmia. In this section five ventricular arrhythmias are presented:

- Premature Ventricular Contraction
- Ventricular Tachychardia
- Ventricular Fibrillation
- Idiovenrticular Rhythm
- Asystole

The **Premature Ventricular Complex(PVC)**(Figure 1.15)is not a rhythm but instead a single ectopic beat originating from an irritable ventricular focus. For this reason, the complex will come earlier than expected on the cardiac cycle and will interrupt the regularity of the underlying rhythm. Because PVCs originate in the ventricles, the QRS will be wider than normal. But a second feature of a ventricular focus is that there is no P wave preceding the QRS complex. One of the things that

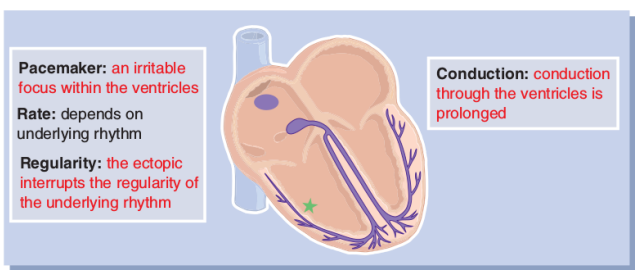


Figure 1.15. Mechanism of Premature Ventricular Contraction, source:[5]

gives a PVC a bizarre appearance, in addition to the width of the QRS complex, is the tendency for PVCs to produce a T wave that extends in the opposite direction of the QRS complex. Another frequent feature that may be helpful in identifying a PVC is the compensatory pause that usually follows a ventricular ectopic.

A **compensatory pause** is created when a PVC comes early, but since it doesn't conduct the impulse retrograde through the AV node, the atria are not depolarized. This leaves the sinus node undisturbed and able to discharge again at its next expected time. The result is that the distance between the complex preceding the PVC and the complex following the PVC is exactly twice the distance of one R-R interval.

Another configuration possibility is that the PVC be followed by no pause whatsoever. This occurs when the PVC squeezes itself in between two regular complexes and does not disturb the regular pattern of the sinus node. This phenomenon is called an **interpolated PVC**, because the PVC inserts itself between two regular beats

If the myocardium is extremely irritable, the ventricular focus could speed up and override higher pacemaker sites. This would create what is essentially a sustained run of PVCs. This rhythm is called **Ventricular Tachychardia(VT)**(Figure 1.16) In ventricular tachychardia you will see a succession of PVCs across the strip at a

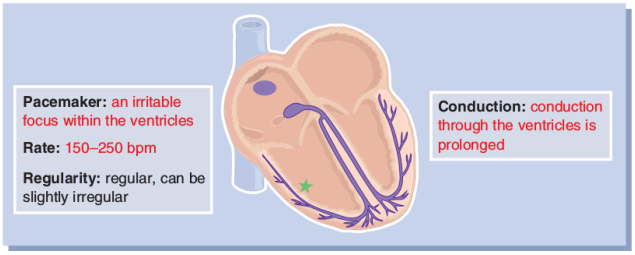


Figure 1.16. Mechanism of Ventricular Tachychardia, source:[5]

rate of about 150-250 bpm. This arrhythmia usually has a very uniform appearance, even though the R-R interval may be *slightly* irregular. It is possible for VT to occur at slower rates, but when it does, it is qualified by calling it a slow VT. All the description related to PVCs apply to VT.

In extremely severe cases of ventricular irritability, the electrical foci in the ventricles can begin fibrillating. This means that many foci are firing in a chaotic, ineffective manner, and that the heart muscle is unable to contract in response. **Ventricular Fibrillation**(Figure 1.17) is a lethal arrhythmia, since the rhythm is very chaotic and ineffective. Ventricular Fibrillation (VF) is probably the easiest of

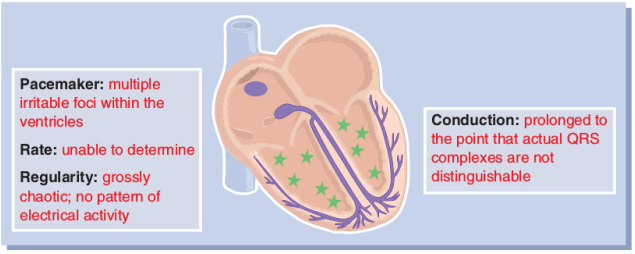


Figure 1.17. Mechanism of Ventricular Fibrillation, source:[5]

all the arrhythmias to recognize. This is because there are no discernible complexes or intervals and the entire rhythm consists of chaotic, irregular activity. VF has no measurable waves or complexes.

There are two ways a ventricular focus can assume control of the heart. One is irritability and the other is **escape**(see Section 1.2). A ventricular escape rhythm is one that takes over pacemaking in the absence of a higher focus and depolarizes

the heart of the inherent rate of the ventricles, which is 20-40 bpm. This rhythm is called **Idioventricular Rhythm**(Figure 1.18). It is not possible to see P waves in an Idioventricular Rhythm, since the escape mechanism would take over only if the atrial pacemaker sites had failed. Idioventricular Rhythm is initiated by the

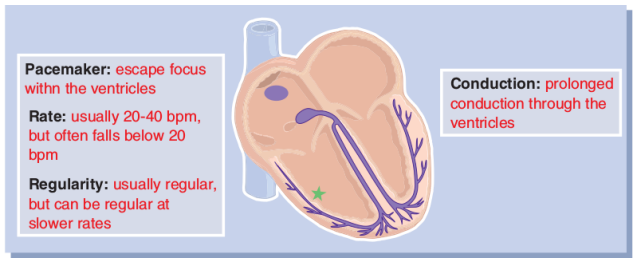


Figure 1.18. Mechanism of Idioventricular Rhythm, source:[5]

very last possible fail-safe mechanism within the heart. When the rhythm is in its terminal stages, that is, as the patient is dying, the complexes can lose some of their form and be quite irregular. In this stage, the arrhythmia is said to be **agonal**, or a dying heart. The word agonal is used to describe a terminal, lethal arrhythmia, especially when it has stopped beating in a reliable pattern. Idioventricular Rhythm is an agonal rhythm, especially when the rate drops below 20 bpm and the pattern loses its uniformity.

The last stage of a dying heart is when all electrical activity ceases. This results in a straight line on the ECG, an arrhythmia called **Asystole** Asystole is a lethal

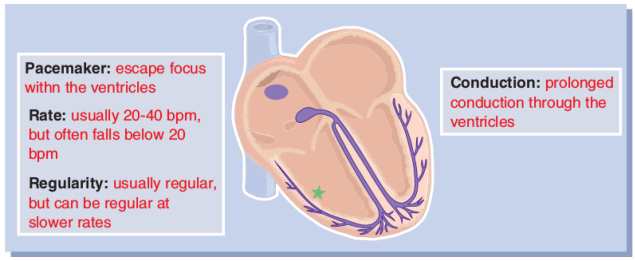


Figure 1.19. Mechanism of Idioventricular Rhythm, source:[5]

arrhythmia that is very resistant to resuscitation effort

1.4.4 Junctional Rhythms

The *AV junction* consists of the *AV node* and the *Bundle of His*(see Section 1.2). This unique part of the conduction system is responsible for conducting impulses from the SA node down the conduction pathways to the ventricles. The *body* of the AV node is responsible for delaying each impulse just long enough to give the ventricles time to fill before contracting. The lower region of the AV junction houses the pacemaking cells that initiate the group of arrhythmias called **junctional rhythms**.

When electrical impulses originate in the AV junction, the heart is depolarized in a somewhat unusual fashion: with the pacemaker located in the middle of the heart, the impulses spread in two directions simultaneously(see Figure 1.20). Recalling from Section 1.3.1, electrode positions for Lead II place the negative electrode above the right atria and the positive electrode below the ventricle. In the normal

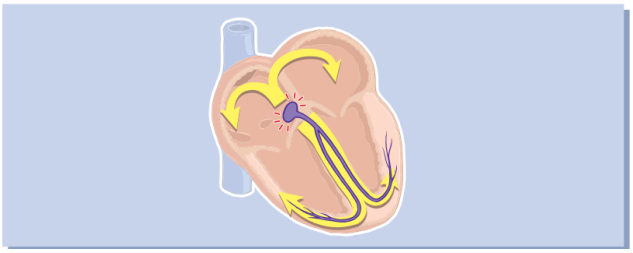


Figure 1.20. Electrical Flow in Junctional Arrhythmias, source:[5]

heart, the major thrust of electrical flow is toward the ventricles (and toward the the positive electrode in Lead II), thus producing an upright QRS complex. In a junctional rhythm, the ventricles are depolarized by an impulse travelling down the conduction system toward the positive electrode; thus the QRS complex is upright. But, at the same time, the impulse can spread upward through the atria toward the negative electrode. When the atria are depolarized in this *backward fashion*, it is called **retrograde** conduction because the electrical impulse travels in the opposite direction it usually takes. Thus, the atrial activity will produce a *negative* deflection on the ECG. In other words, the P wave of an AV junctional arrhythmia should be inverted beacuse it was produced by an impulse travelling toward the negative electrode.

In junctional arrhythmias, the P wave does not always have to precede the QRS complex beacause it is possible for the ventricles to be depolarized before the atria, if the force reach them first. If they both depolarize simultaneously the P wave will be hidden within the QRS complex.

The junctional pacemaker site can produce a variety of arrhythmias, depending on the mechanism employed. Four basic mechanisms common to the AV junction will be discussed:

- Premature Junctional Complex
- Junctional Escape Rhythm
- Accelarated Junctional Rhythm
- Junctional Tachycardia

The **Premature Junctional Complex** or PJC (Figure 1.21), is not an entire rhythm; it is a single ectopic beat. A PJC is similar in many ways to a Premature Atrial Contraction(PAC)(see Section 1.4.2). In the case of PJC, the irritable focus comes from the AV junction to stimulate an early cardiac cycle, which interrupts the underlying rhythm for a single beat. The P wave of a PJC is inverted as for the others junctional arrhythmias.

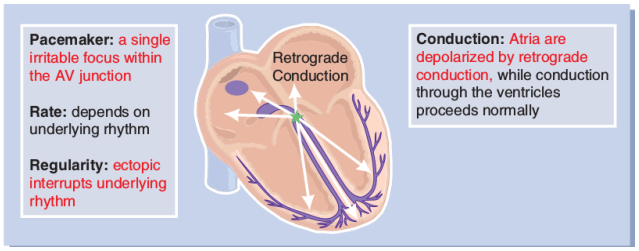


Figure 1.21. Mechanism of Premature Junctional Complex, source:[5]

A **Junctional Escape Rhythm** (Figure 1.22) is expected to have a rate of 40-60 bpm since this is the inherent rate of the AV junction.

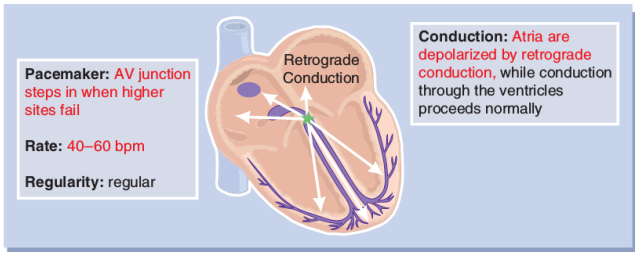


Figure 1.22. Mechanism of Junctional Escape Rhythm, source:[5]

A junctional escape rhythm has all the characteristics previously described of a junctional arrhythmia. Such rhythm us a fail-safe mechanism rather than an irritable arrhythmia(see Section 1.2). However, the AV junction is capable of irritability and is known to produce an irritable arrhythmia called **Junctional Tachycardia**. This rhythm occurs when the junction initiates impulses at a rate faster than its inherent rate of 40-60 bpm, thus overriding the SA node. Junctional Tachycardia is usually divided in two categories, depending on how fast the irritable site is firing. If the junciton is firing between 60 and 100 bpm, the arrhythmia is termed an **Accelarated Junctional Rhythm**(Figure 1.23) When the rate exceeds 100 bpm,

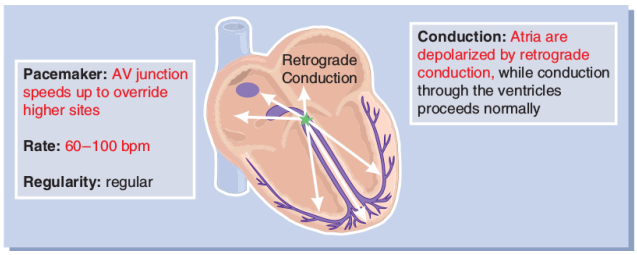


Figure 1.23. Mechanism of Accelerated Junctional Rhythm, source:[5]

the rhythm is considered a Junctional Tachycardia (Figure 1.24)

Junctional Tachycardia can be as fast as 180 bpm, but at this rapid rate, it is extremely difficult to identify positively since P waves are superimposed on preceding T waves.

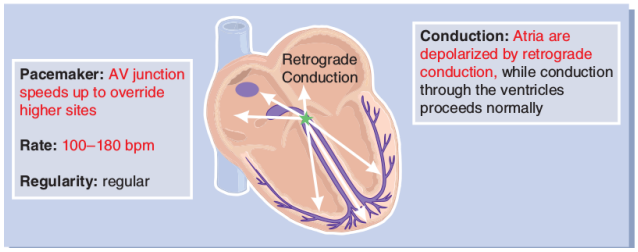


Figure 1.24. Mechanism of Junctional Tachycardia, source:[5]

The only difference appreciable on the ECG signal among Junctional Escape Rhythm, Accelarated Junctional Rhythm and Junctional Tachycardia is the rate. The rates are listed in table 1.1 :

Rhythm	**Rate(bpm)**
Junctional Escape	40-60
Accelerated Junctional	60-100
Junctional Tachycardia	100-180

Table 1.1. Rates of Junctional Rhythms, source:[5]

Chapter 2

Heartbeat Detection

The first stage of an arrhythmia detection system consists in the extraction of heartbeats along the ECG signal. In the following chapter it is described the **Heartbeat Detection** module, aimed to locate a sequence of heartbeats inside a raw ECG signal.

The heartbeat, as observed in the ECG tracing, is composed by three waves (Section 1.3.3): the P, QRS and the T wave. The most prominent in terms of amplitude is the **QRS wave** and therefore it is the easiest to detect. Hence, the heartbeat detection problem is equivalent to the **QRS detection**.

The QRS detection problem, on its standard formulation, takes as input an ECG sample and computes whether it resides in a QRS wave.

QRS detection is difficult, because the beat morphology varies along the time, and different sources of noise can be present[7]. Noise sources include muscle noise, artifacts (Section 1.3.4) due to electrode motion, power-line interference, baseline wander, and T waves with high-frequency characteristics similar to QRS complexes.[9]

Most QRS detection algorithms have two differentiated stages: preprocessing and decision [8]. In preprocessing stage different techniques are applied to the signal, such as linear and non-linear filtering or smoothing to attenuate P and T waves as well as noise. In decision stage the most important task is the determination of thresholds and in some cases the use of techniques to discriminate T waves. Some algorithms include another decision stage to reduce false positives.

Once the QRS waves have been detected, it is possible to solve the more specific **R Peak detection** problem: The R peak is the sample of maximal amplitude inside the QRS wave. From the locations of the R peaks in the ECG tracing it is possible to extract the complete sequence of heartbeats, useful for arrhythmia analysis purposes.

In this work, the R peak detection problem is formalized as a **Machine Learning task**. This is possible when dealing with databases **annotated** by expert cardiologists: a learning algorithm is **trained** on the labeled data in order to extract useful knowledge to be applicable to real world data. This scenario is the

standard **Supervised Learning** framework. In this study, two Supervised learning approaches based on the *KNN algorithm* have been implemented and evaluated in terms of quality of the results and computational time measured on different IoT platforms. Another algorithm has been implemented in order to provide a more complete comparison: the **First-Order difference algorithm** is a naive algorithm designed to find the local maxima in a generic signal. The decision of considering this approach was taken in order to have a comparison between ad-hoc methods for ECG analysis for heartbeat detection and a standard signal processing module.

In Section 2.1 a selection of the most representative state of the art algorithms for R Peak detection, including the aforementioned one, have been described. In Section 2.2 it is possible to find a description of the procedures based on the *KNN algorithm* implemented in this study in order to solve the heartbeat detection problem. In Section 2.3 we define the assumptions considered for evaluating the procedures in terms of correctness of the results and computational time. In Section 2.4 we report all the combinations of parameters studied and the obtained score, in order to select the best ones for each beat detection algorithm. In Section 2.5 a complete comparison between the implemented approaches is presented.

2.1 State of the art

2.1.1 Artificial Neural Network

The approach described in [10] consists in a novel QRS complex detector using **3-layer artificial feedforward back propagation neural network**. The training method is Levenberg-Marquardt back propagation. After **baseline drifts elimination** and **lowpass filtering**, each sample in ECG signal is extractedand then given as input to the neural network. The algorithm is tested with the MIT-BIH Arrhythmia Database records and the accuracy is 99.5%. Baseline drifts elimination is done using two median filters (200-ms and 600-ms). The 200-ms median filter is for removing QRS-complexes and P-waves while the 600-ms median filter is for removing the T-waves. By subtracting the filtered signal from the original signal, the baseline drifts and artifacts are removed. In order to reduce the size of the input of the neural network, features of each sample are firstly extracted. This is accomplished by a function called **Features Extractor** whose inputs are the ECG signal and the sample identifier, n . The output of Features Extractor is a matrix of three rows of values, which are shown below.

1. The average amplitude of the samples from n *-16* to n *+16*
2. The average of derivatives of samples before n, which are $n-1$, $n-2$, $n-5$, $n-10$, $n-20$, $n-50$, $n-100$.
3. The average of derivatives of samples after n , which are $n+1$, $n+2$, $n+5$, $n+10$, $n+20$, $n+50$, $n+100$.

The output of the ANN is just a number which ranges between -1 and 1. By using a Sign function, the positive output values of ANN means the corresponding samples

are determined to be R-peak and the negative output values means the corresponding samples are determined not R-peak.

In the conclusive section of the paper is reported that the accuracy obtained on the recordings of MIT-BIH arrhythmia database used as test is 99.5%, but it is not clear what criterion has been followed for the dataset partition in training and test set. Furthermore, the accuracy measure is not relevant in this context, because of the *Accuracy paradox*.[1] Precision and recall measures are more reliable in classification problems in which one or more classes are dominant with respect to others.

2.1.2 Convolutional Neural Network

In the paper [11] it is presented a QRS detection algorithm based on pattern recognition as well as a new approach to ECG baseline wander removal and signal normalization. Each point of the zero-centred and normalized ECG signal is a QRS candidate, while a **1-D CNN classifier** serves as a decision rule. Positive outputs from the CNN are clustered to form final QRS detections. The data is obtained from the 44 non-pacemaker recordings of the **MIT-BIH arrhythmia database**. Classifier was trained on 22 recordings and the remaining ones are used for performance evaluation. The preprocessing step is conducted solely to ensure the data is zero-centred and normalized for the classifier training and prediction step. The idea is to have a well extracted ECG morphology, with as little information loss as possible. Every point of the signal was described by a sample of 145 neighbouring points, resulting in a total of 14296832 data samples. Each data sample is labelled positive if the candidate point is inside a Âś40ms distance of an original positive beat annotation. The proposed CNN architecture besides a 1-D input layer, consists of two convolutional layers with a max-pooling layer between them , two fully-connected layers and a softmax classification layer. All convolutional and fully-connected layers have a dropout probability of 0.5 to reduce overfitting on training data and make the trained model less sensitive to partial deformations of ECG morphology. Stochastic gradient descent with 0.9 momentum and an initial learning rate of 0.005 was used to train the network. Mini-batch size was 128 samples, and the training was limited to 3 epochs. After the classification step, they performed a **hierarchical group-average agglomerative clustering** upon all CNN decisions of a single recording, with clustering criterion being the temporal euclidean distance. Final QRS detection is the mean of all CNN detections within the same cluster. To determine whether a detection is a true positive (TP), a Âś75ms tolerance window is used. This method achieves a sensitivity of 99.81% and 99.93% positive predictive value, which is comparable with most state-of-the-art solutions.

[1] The accuracy paradox is the paradoxical finding that accuracy is not a good metric for predictive models when classifying in predictive analytics. This is because a simple model may have a high level of accuracy but be too crude to be useful. For example, if the incidence of category A is dominant, being found in 99% of cases, then predicting that every case is category A will have an accuracy of 99%. Precision and recall are better measures in such cases

2.1.3 Hidden Markov Models

An HMM [12] is a statistical model used to characterize signal dynamics as a function of time. A typical HMM is composed of n hidden states and the set of parameters $\lambda = (A, B, \pi)$, where:

1. $A = a_{ij}$ is the matrix of state transition probability distribution from state i to j.
2. $B = b_j(k)$ is the observation symbol probability distribution in state j.
3. $\pi = \pi_i$ is the initial state distribution.

A training stage is performed to obtain the set of parameters λ in order to maximize $P(O_{train}|\lambda)$, the probability that an observation sequence $O_{train} = O_1, O_2..., O_T$, taken from the analyzed system, is generated by the model. In the work explained in [13], an automatic QRS complex detector based on **continuous density hidden Markov models (HMM)** is proposed. HMM were trained using univariate observation sequences taken either from QRS complexes or their derivatives. The detection approach is based on the log-likelihood comparison of the observation sequence with a fixed threshold. A sliding window was used to obtain the observation sequence to be evaluated by the model. The threshold was optimized by receiver operating characteristic curves. Sensitivity (Sen), specificity (Spc) and F 1 score were used to evaluate the detection performance. The approach was validated using ECG recordings from the MIT-BIH Arrhythmia database. A 6-fold cross-validation shows that the best detection performance was achieved with 2 states HMM trained with QRS complexes sequences (Sen = 0.668, Spc = 0.360 and F 1 = 0.309).

2.1.4 Support Vector Machine

The paper [14] presents an algorithm for QRS complex detection based of support vector machine (SVM). The proposed algorithm is evaluated on annotated standard databases such as MIT- BIH Arrhythmia database. The procedure of preliminary processing of a signal is used as the signal contains different noise and artifacts:a low pass filter with a cut-off frequency of 13 Hz and high pass filter with a cut- off frequency of 9 Hz are used, as well as the function of the moving average with a 5 measurements window sizes. Informative feature as **rise speed of a signal** is chosen, because QRS complex possesses the greatest climb rate. This informative feature is implemented as follows: an array of the values of the slope of the tangent to each point of the filtered and squared ECG signal. After that SVM classification function is applied to the received selection. Informative feature as **correlation forms of QRS-complexes** are chosen, because QRS-complex possesses a specific form. This informative feature is implemented as follows: Firstly, test pulse is created based on 1000 QRS-complexes with the R-peak in the middle lasting 51 counting (141.67 ms), which occurs after their averaging, for this procedure the signal N.100 is used. Secondly, an array of correlation coefficients forms of QRS-complexes in a moving window is created. After that SVM classification function is applied to the received selection. For train recording signals No100, No104, No214, No200, No205 are used. The training set consists of 30,000 values for each of the signs.

The function of classification was calculated for each informative feature separately. Signals from MIT-BIH database with 30 min of duration are used for testing. After testing, a train of 1's is obtained at the output of SVM classifier. Then this train of 1's is picked and by using their duration, average pulse duration of 1's is evaluated. Those trains of 1's, whose duration turns out to be more than the average pulse duration are detected as QRS-complex and the others are discarded. The QRS detector obtained a sensitivity Se = 98.32% and specificity Sp = 95.46%.

2.1.5 First-Order Difference

The First-Order Difference algorithm is a naive procedure designed to find the local maxima in a generic signal. The decision of considering this approach was taken in order to have a comparison between ad-hoc methods for ECG analysis for heartbeat detection and a standard signal processing module.

Since this is a standard signal analysis algorithm, the preprocessing phase is not aimed to improve the performance but just to adapt the input data. The implementation used to generate the results, named *PeakUtils*[24], requires a positive normalized signal as input. This means that every amplitude of the signal must be in the positive and in the range [0,1]. In order to achieve this, a *max-normalization* is performed: each sample of the signal is divided by the value of the maximum amplitude. After that transformation, the maximum of the signal will have a resulting amplitude of 1, and the values corresponding to the other samples will represent the amplitude in percentage with respect to the maximum. The whole process is formally expressed by:

$$sample = \frac{abs(sample)}{\max(abs(signal))}$$

The first order difference in the signal, i.e. the first derivative [24] , defined in this context as simple as:

$$y'(n+1) = y(n+1) - y(n)$$

The locations of the local maxima, i.e. the candidate **R Peak indexes**, correspond to the ones that satisfy the following conditions:

- The first order difference is zero, $y'(n) = 0$
- The previous and the subsequent value of the point, are of discordant sign: $y'(n-1) - y'(n+1) < 0 \; or \; y'(n+1) - y'(n-1) < 0$

These conditions lead to the detection of points in which a curvature of the original function occurs, the presumed rpeak locations.

Such computation is refined with the tuning of the following parameters:

- **Threshold :** Only the peaks with amplitude higher than the threshold will be detected.
- **Minimum Distance:** Minimum distance between each detected peak. a peak detection that is at distance lower than the minimum from the previous will be discarded. From a medical point of view, the physiological minimum distance between two heartbeats is 0.2s

2.1.6 Pan Tompkins

The Pan Tompkins algorithm [9], is a real-time *adaptive thresholding* method for R Peak detection. The detection of QRS complexes is based upon digital analysis of slope, amplitude and width of the ECG data.

The algorithm implements a special digital band pass filter. It can reduce false detection caused by the various types of interference present in the ECG signal.

The algorithm automatically adjusts the thresholds and parameters periodically to adapt the changes in QRS morphology and heart rate. There are two sets of thresholds to detect QRS complexes. One set thresholds the filtered ECG while the other thresholds the signal produced by **moving window integration**. The signal emitted after the moving window integration technique is actually the same as the original ECG except for the fact that is delayed by the total processing time of the detection algorithm. The algorithm uses a dual-threshold technique to find missed beats, and thereby reduce false negatives. Hence, there are two separate threshold levels both for the filtered signal and for the one produced after the applying of the moving window integration. One level is half of the other. Note that the higher of the two thresholds in each of the two sets is used for the first analysis of the signal while the lower threshold is used if no QRS is detected in a certain time interval to perform a **search-back technique**: If a QRS complex is not found in the time interval corresponding to 166% of the current average RR interval, the maximal peak detected in that time interval that lies between these two thresholds is considered to be a possible QRS complex. Once a valid QRS complex is recognized, there is a 200ms refractory period before the next one can be detected. This refractory period eliminates the possibility of a false detection such as multiple triggering on the same QRS complex during the time interval. When a QRS detection occurs after the end of the refractory period but within 360ms of the previous complex, it is important to understand if the candidate QRS is a valid QRS complex or a T wave. In this case, the waveform with the largest slope is judged to be a QRS complex.

2.2 K Nearest Neighbors approaches

Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as **supervised learning** problems. [15] Given that the database contains cardiologist **annotations** (Section 2.3.1.1) corresponding to the QRS locations, it is possible to consider a supervised learning approach to solve the R peak detection problem.

The **K Nearest Neighbors**(KNN) algorithm has been chosen in this study because it is considered one of the simplest machine learning approaches. Given a distance function d a training set of labeled data $\{X\_train,\ Y\_train\}$, and a test set X_test: the algorithm retrieves the K-neighbors set N of each sample in X_test according to d, and determines the corresponding label with some voting strategy (Section 2.2.3.2) based on the labels of the samples in N.

Thus to classify a new point, we identify the K nearest points from the training data set and then assign the new point to the class having the largest number of representatives amongst this set. The particular case of K = 1 is called the nearest-neighbor rule because a test point is simply assigned to the same class as the nearest point from the training set[15].

The nearest neighbors of an instance can be retrieved, as an example, in terms of the standard Euclidean distance. More precisely, let an arbitrary instance x be described by the **feature vector**:

$$[a_1, a_2(x), ...a_n(x)]$$

The the distance between two instances x_i and x_j is defined to be $d(x_i, x_j)$ where[16]:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2}.$$

The **QRS detection** problem consists in determining whether a sample of the ECG signal belongs to a QRS wave or not, therefore it is a **binary classification** problem. In this section the Supervised learning procedures implemented in the project in order to solve the QRS detection problem and then R Peak detection task are described. Two different approaches have been considered:

- **SSK algorithm**: it is the algorithm described in [3], based on the *KNN algorithm*
- **QRS KNN algorithm**: it is a **new algorithm**, also based on the *KNN algorithm* according to a novel feature definition

2.2.1 Signal Preprocessing: Filtering and Squaring Techniques

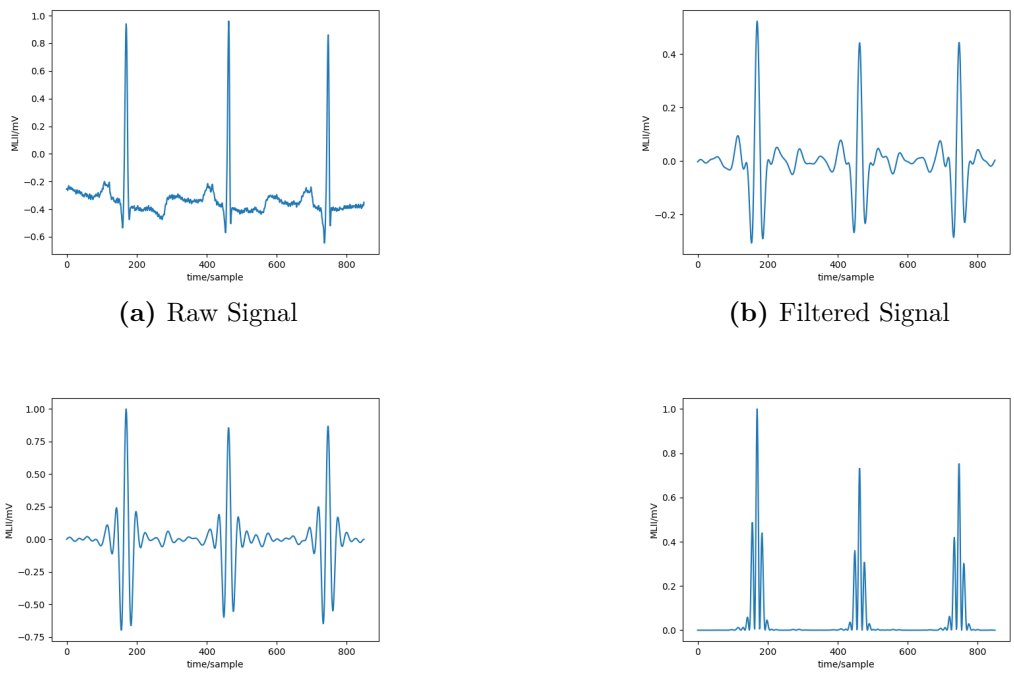
We consider a **record** as a structure composed by a number (n) of **channels**, therefore the vector is n-dimensional. The elements of the channels represent the amplitude of the signal at any given time, called **samples**. A channel is the signal recorded from a specific lead configuration(Section 1.3.1).

The ECG signal is affected by various sources of noise. Noise sources include muscle noise, artifacts(Section 1.3.4) due to electrode motion, baseline wander, and T waves with high-frequency characteristics similar to QRS complexes.[9] In this approach, **digital filters** reduce the influence of these noise sources. The input signal is first processed by a **bandpass filter**. Such filter reduces the influence of muscle noise, baseline wander[2], and T-wave interference. The desirable passband to maximize the QRS energy is approximately 5-15 Hz.[9] It is implemented with a cascade of low-pass and high-pass filters to achieve a 3dB passband from about 5-12 Hz, reasonably close to the desirable passband. The output is then processed by a **derivative filter** in order to provide the QRS complex slope information. After differentiation, the signal is squared point by point. This makes all data

[2]baseline wander (BW) is actually the effect where the isoelectric line (Section 1.3.3) viewed on a screen appears to 'wander' or move up and down rather than be straight. that may be due to respiration or the motion of the patients or the instruments.

points positive and does a nonlinear amplification of the output of the derivative emphasizing the higher frequencies (i.e., predominantly the ECG frequencies and the QRS complex in particular).

The result of the preprocessing stage is shown step-by-step in Figure 2.1:



(a) Raw Signal

(b) Filtered Signal

(c) Derived Signal

(d) Squared Signal

Figure 2.1. Preprocessing stages

2.2.2 SSK Algorithm

This algorithm is the one explained in the paper *QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases*[3] In this approach, a [m × n] feature matrix is formed, consisting of m instances of n features. The number of feature vectors (m) is equal to the number of samples of selected portions of ECGs, i.e. the dimensionality of a channel. Hence, there is a feature vector for each sample. The feature vector is n-dimensional, where n is the number of channels employed in the computation.

The feature labels are assigned according to the database annotations: $\forall$ *sample s in X_train*:

$$label(s) = \begin{cases} 1 & if\ s\ \in QRS\ region \\ -1 & otherwise \end{cases}$$

In the annotations files, the only labeled samples are the R peaks, so it is up to the author to define the amount of points around the annotated peak to consider as a QRS region. Since the QRS width varies patient by patient, it is necessary to find

an average amount of points that seems reasonable for all signals. Unfortunately, the authors did not specify the choice they made. People very rarely agree on what a normal time range is for the QRS measurement. It is usually considered to be between 0.06 and 0.11 seconds([5], Chapter 2.46). For simplicity we decided to consider the QRS of width 0.1 seconds.

The test samples that are predicted from the classifier, are the ones belonging to QRS regions. After testing, a train of 1's is obtained at the output of KNN classifier. Then this train of 1's is picked and by using their duration, average pulse duration of 1's is evaluated. Those trains of 1's, whose duration turns out to be more than the average pulse duration are detected as QRS-complex and the other are discarded.

To the extent of comparison with the other approaches, we added a final module performing the **detection of the R peak** inside the remaining regions. The R peak location is then detected from such regions by simply applying the following: $\forall detected\ QRS\ region\ qrs_i$:

$$rpeak\_i = argmax(|qrs_i|)$$

The location of the R Peak is the one corresponding to the maximum absolute value in the region detected as a QRS wave. Channels corresponding to V1-V5 lead configurations presented the R peak annotations on the negative peaks; that is why we added the absolute value in the computation.

2.2.2.1 Training

The training step is performed with an **inter-patient** perspective: Records (corresponding to patients) are partitioned in training and test sets: a single classifier is employed in the whole process. In this approach, only one record is chosen for training and all the others are used for performance evaluation.

At this point it is performed a **5-fold crossvalidation** in order to avoid to *overfit* on the training set. **Overfitting** is the production of an analysis that corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably.[17] In k-fold cross-validation, the original training set is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the **validation data** for testing the model, and the remaining k - 1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results are then averaged to produce a single estimation.[18]

2.2.2.2 Model Selection

The KNN classifier is then tuned with a configuration of parameters to be chosen inside a space of candidate combinations (Section 2.4.2.1). The classifier is chosen according to the *number of neighbors* and the **distance metric**[19].

The **number of neighbors** attribute determines the cardinality of the neighbors set, i.e. how many neighbors labels to consider for predicting the label of the test sample. The **distance metric** attribute is selected between the *Manhattan distance*, or *City Blocks*, the *Euclidean distance* and the *Correlation distance*.

The *Euclidean distance metric* is the most usual way of computing a distance between two objects. It examines the root of square differences between coordinates of a pair of objects and is defined using the following equation[20]:

$$d_{st} = \sqrt{\sum_{r=1}^{n} (x_{sj} - y_{tj})^2}.$$

The *Manhattan distance* is based on Taxicab geometry, first considered by Hermann Minkowski in the 19th century, is a form of geometry in which the usual metric of Euclidean geometry is replaced by a new metric in which the distance between two points is the sum of the absolute differences of their coordinates defined using the following equation:

$$d_{st} = \sum_{r=1}^{n} |x_{sj} - y_{tj}|.$$

The *Correlation distance* is one minus the sample correlation between points (treated as sequences of values), that is a measure of the strength and direction of the linear relationship between two variables that is defined as the covariance of the variables divided by the product of their standard deviations.[21] In order to pick the best one, the KNN classifier is trained trying all the possible configurations of parameters inside the candidate space.

The result produced by each combination are compared in terms of a selected **score**: In this context is considered the *accuracy score*, defined as[22]:

$$\frac{TP+TN}{TP+TN+FP+FN}$$

In this context, a **True Positive**(TP) is a correctly detected QRS region, a **False Positive** it is a region that is wrongly detected as a QRS, a **True Negative** is a region correctly detected as a part of th signal that is different from a QRS wave and finally a **False Negative** is the case in which a real QRS region has not been detected. In table 2.1 it is possible to find the definition of TP, FP, TN, FN in this context.

predicted/real	**QRS**	**not QRS**
QRS	TP	FP
not QRS	FN	TN

Table 2.1. TP, FP, TN, FN definition for QRS detection

The whole process of trying all the possible combination of parameters and selecting the one with the best score using cross validation is known as **Crossvalidated Grid Search**[19].

2.2.3 QRS KNN Algorithm

This is a **new approach** in which the feature vector corresponds no longer to a sample(Section 2.2.2) but to a time segment, i.e. a **window**. The algorithm determines for each window whether it is a QRS wave or not. The signal is therefore segmented in equally sized (see Section 2.4.3.2) and contiguous windows. Consequently, each segment in the training set is labeled according to the database annotations: $\forall$ *segment s in X_train*:

$$label(s) = \begin{cases} 1 & if\ s\ contains\ a\ QRS\ annotation \\ -1 & otherwise \end{cases}$$

The **QRS detection** output consists on the classifier predictions for each unseen region in the test set. The segments to which corresponds a positive prediction(+1) are the ones which presumably contains a QRS wave.

The **R peak** location is then detected from such regions, as explained in 2.2.2 by applying the following:

$$rpeak\_location = argmax(abs(region))$$

The location of the R Peak is the one corresponding to the maximum absolute value in the region detected as a QRS wave. This is because, given that a time segment has been detected as a QRS wave, it is very likely that, in such a small portion of the signal, the R peak is located at the point of maximum amplitude.

2.2.3.1 Training

The training step is performed with an **intra-patient** perspective: We considered a separated classification instance for each patient; Every single record is partitioned into two sets aimed for training the classifier and for performance evaluation. Therefore, we trained one KNN classifier on the first n seconds (see Section 2.4.3.4) of the signal corresponding to the record from one patient. At this point as in Section 2.2.2.1, we employed a **5-fold crossvalidation** in order to avoid to *overfit* on the training set.

2.2.3.2 Model Selection

The KNN classifier must be tuned with a configuration of parameters to be chosen inside a space of candidate combinations (Section 2.4.3.1). We tuned the KNN classifier for what concerns the *number of neighbors* and *weights* attributes.[19] The **number of neighbors** attribute determines the cardinality of the neighbors set, i.e. how many neighbors labels to consider for predicting the label of the test sample. The **weights** attribute is about the **voting strategy** adopted for the final prediction.

The *uniform strategy* is a simple majority voting: the most frequent label in the neighbors set will be the label associated with the test sample. In this context of binary classification, the number of neighbors value should be set to an odd number to avoid ties.

The *distance strategy* biases the votes counting according to the inverse distance between the test sample and its neighbors. The nearer the neighbor, the greater will be its influence in the final result.

In order to pick the best one, the KNN classifier is trained trying all the possible configurations of parameters inside the candidate space. The result produced by each combination are compared in terms of a selected **score**: In this context we considered the **F1 score**, defined as [22]:

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall}$$

where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$
The definitions of True Positives(TP), False Positives(FP), True Negatives(TN) and False Negatives(FN) in this context are explained in Section 2.2.2.2, Table 2.1.

2.3 Evaluation

Since not all the annotations of the R peak locations in the database are placed precisely on the maximum of the QRS wave but guaranteed to be inside it, we could not use them as we find them in the database for the evaluation purpose. For this, a peak detection is considered **correct** if it resides inside a range of 0.1s around the real peak annotation. Such amount of sample is chosen to be comparable to the average QRS width.([5], Chapter 2, Section 41)

According to this definition of correctness, we evaluate the performance of the proposed algorithms with the following measures:

$$Precision = \frac{|correctly\ detected|}{|detected|}$$

$$Recall = \frac{|correctly\ detected|}{|real\ peaks|}$$

$$F1\ Score = \frac{2Precision \times Recall}{Recall + Precision}\ [22]$$

The result reported in all the plots of this document (Section 2.4) consists in the average of results of precision and recall obtained for all signals (patients).

A critical aspect in the implementation of an ECG analysis algorithm, is the possibility of a *real time* execution. Therefore, we measured the elapsed time of the computation of all the algorithms considered in this study. For this, we picked as unit measure, the **elapsed time for processing a sample**, i.e. the amount of time that occurred to the decision stage for determining whether a sample is an R peak or not. More precisely:

$$elapsed\_time = \frac{t}{n}$$

where t is the total time to process one signal and n the number of samples in it.

2.3.1 Database

The database considered in this study is a collection of annotated ECG recordings obtained by the Arrhythmia Laboratory of Boston's Beth Israel Hospital, named the **MIT-BIH Arrhythmia Database**[23]. From such collection, 48 half-hour excerpts of two-channels, 24 hours, ECG recordings obtained from 47 subjects (record 201 and 202 are from the same subject) have been selected. Of these, 23 (the "100 series") were chosen at random from a collection of over 4000 Holter tapes, and the other 25 (the "200 series") were selected to include examples of uncommon but clinically important arrhythmias that would not be well represented in a small random sample. The subjects included 25 men aged 32 to 89 years and 22 women aged 23 to 89 years; approximately 60% of the subjects were inpatients.

In most records, one channel is a modified limb lead II, **MLII**, obtained by placing the electrodes on the chest as is standard practice for ambulatory ECG recording, and the other channel is usually **V1**(sometimes V2, V4 or V5 depending on the subject). The digitization rate or **frequency of sampling** is of 360 samples per second per channel). Four of the 48 recordings include paced beats, i.e. produced by a pacemaker.

Each recording of the MIT-BIH Arrhythmia Database corresponds to a 2D vector of 650K entries. The vector is composed of two channels, therefore it is two dimensional.

2.3.1.1 Annotations

Once the digital tapes had been prepared, they have been annotated using a simple slope-sensitive QRS detector. The charts for each recording were given to two cardiologists, who worked independently, adding additional beat labels and deleting false detections as necessary, and changing the labels for abnormal beats. The cardiologists also added rhythm and signal quality labels. At this point the two sets of annotations were compared: each discrepancy was examined and resolved by consensus. Notably, six of the 48 records contain a total of 33 beats that remain unclassified, because the cardiologist-annotators were unable to reach agreement on the beat types. The database contains seven episodes of loss of signal or noise so severe in both channels simultaneously that QRS complexes cannot be detected; these episodes are all quite short and have a total duration of about 10 s. In all of the remaining data, every QRS complex was annotated, about 109,000 in all.

2.4 Protocol Fine-Tuning

In this section, we report all the combinations of parameters studied and the obtained score, in order to select the best ones for each beat detection algorithm. The performance comparison allows choosing the combination of parameters that maximizes the *F1 score* (see Section 2.3).

2.4.1 First-Order Difference Algorithm

We tried out different threshold values, precisely in a range from 0.1 to 1 with a step of 0.1. The result, still measured according to precision and recall (see Section 2.3), are depicted in Figure 2.2. The best threshold is the one that maximizes the *F1 score* (Section 2.3).

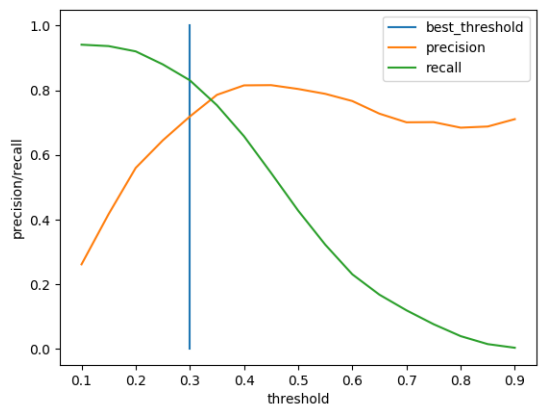


Figure 2.2. Precision and Recall for different threshold values

The best threshold value is 0.3. This means that every sample with amplitude less than 0.3 mV after the preprocessing stage(Section 2.1.5) cannot be detected as a peak.

2.4.2 SSK Algorithm

2.4.2.1 Model Parameters

The model selection phase consists in choosing the best configuration of parameters inside a set of candidates. We tried all the possible combinations inside the space of parameters defined in [3]:

- **number of neighbors** : [1, 3, 5, 7, 9]
- **distance**: [‘Manhattan distance’, ‘Euclidean distance’, ‘Correlation distance’]

2.4.3 QRS KNN

2.4.3.1 Model Parameters

The model selection phase consists in choosing the best configuration of parameters inside a set of candidates. We tried all the possible combinations inside the following space of parameters:

- **number of neighbors** : [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
- **weights**: [‘uniform’, ‘distance’]

2.4.3.2 Window Size

The approach explained in Section 2.2.3, requires to split the signal into contiguous and fixed size regions.For this, we had to choose the window size. We tried different size values and compared them according to the precision and recall (Section 2.3) of the outcomes produced. The results of this experiment are depicted in Figure 2.3:

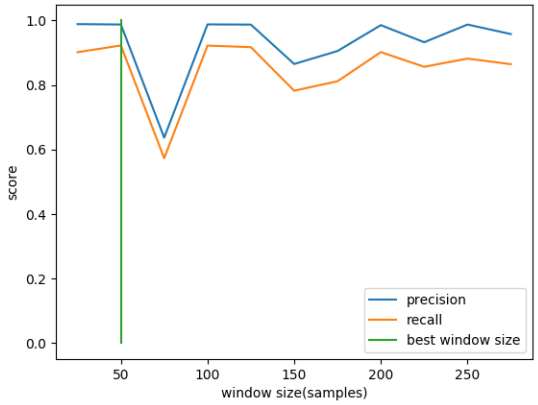


Figure 2.3. Precision and Recall for different feature size values

From this, we deduced that the *F1 Score* is maximized when the window size is set to 50 samples. In the sequel, all experiments will use this window size.

2.4.3.3 Channels

The *MIT-BIH Arrhythmia Database* provides two channels records, i.e. data coming from two different configurations of electrodes. The first channel is, in most of the cases, the output of the *MLII* lead, which is the most used in literature for the R peak detection purpose.

By the way, we wanted to analyze whether the inclusion of the second channel would improve the results. In Figure 2.4 are depicted the results obtained with one channel and two channels separately, and the ones obtained by the combination of both, in terms of *Precision, Recall and F1 Score*(Section 2.3) averaged for all signals.

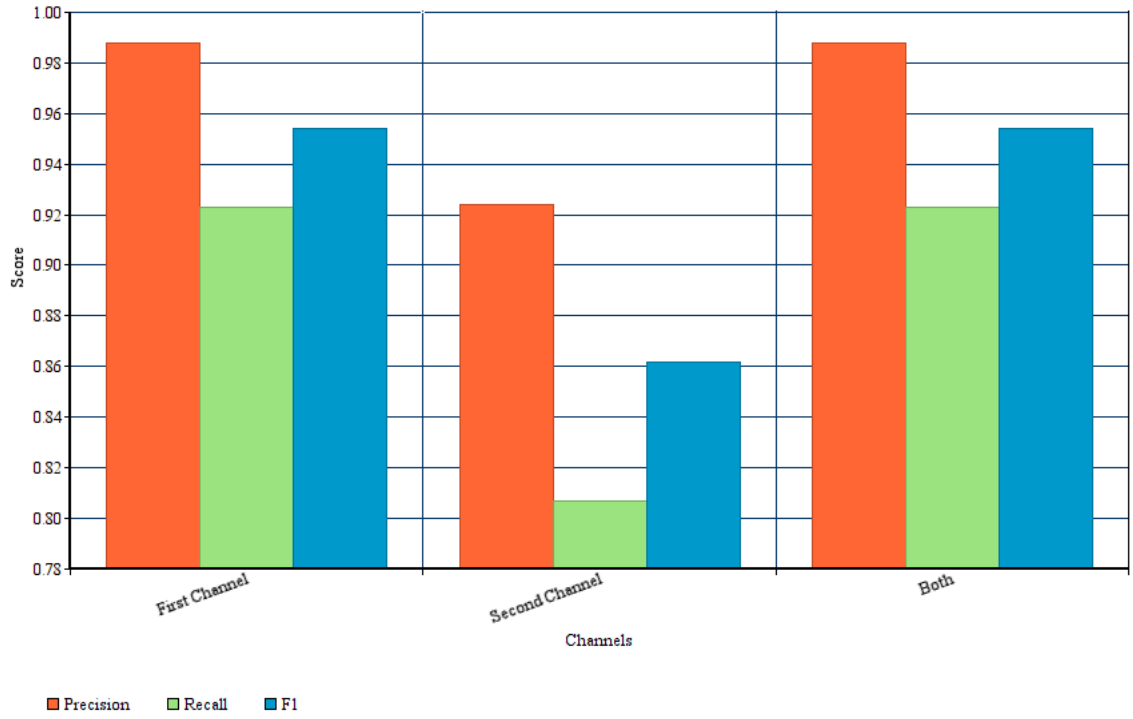


Figure 2.4. Precision, Recall and F1 Score with First, Second and Both Channels

Channels	Precision	Recall	F1 Score
1	0.988	0.923	0.955
Both	0.988	0.923	0.955
2	0.924	0.807	0.867

Table 2.2. Precision, Recall and F1 Score for First, Second and Both Channels

From this, we deduced that the inclusion of the second channel does not provide an improvement of the results.

Due to the way that the electrode positions are orthogonally placed in Holter recording, a high-quality signal on one channel normally implies a low-amplitude ECG with a poor signal-to- noise ratio on the second channel. The only way that two- channel algorithms will yield improved performance for most patients is by adopting a new way of electrode placement that will provide usable signals in both channels.[9]

2.4.3.4 Training size

We studied the minimum amount of signal necessary for the training step in order to achieve appreciable results. A small training set is ideally preferable for a faster computation and a more realistic evaluation of the performance on the test set. Furthermore, with a greater test size, we will have at disposition more outcomes usable in the following steps of the work, that is the Arrhythmia classification. For this reason, we tried different training size values expressed in percentage of the total size of the signal.

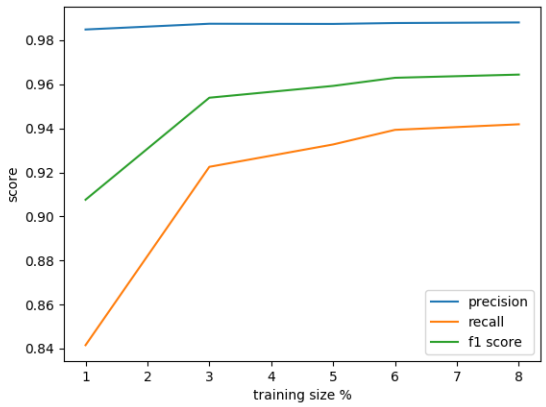


Figure 2.5. Precision, Recall and F1 Score with varying test size

From this, we deduced that a training size corresponding to the 3% of the signal, that is the first 54 seconds of each signal, is the smallest that produces an acceptable result. This is because is the smallest training size that produces an average result greater than 90%. In this way, we discard from the following stages of computation only 56 beats on average. We could obtain better results of devolving more data for training purposes but at the cost of fewer beats for the arrhythmia classification stage.

2.5 Comparative Evaluation

In this section, it is presented the comparative evaluation of the implemented approaches for R Peak detection. At first, we wanted to prove the usefulness of the preprocessing stage(Section 2.2.1); thus, we evaluated the algorithms preformances when applying the same procedures both on the raw and the filtered signals. As another experiment, we studied how much the lead configuration selection, (i.e. which channels to include in the computation) is relevant for the quality of the results. Therefore, we measured the algorithms performances considering all the combinations of channels (First, Second and Both). Finally, we performed the evaluation of the efficiency of the procedures applied to all the aformentioned combinations of the input. The elapsed time to determine whether a sample is a R peak has been considered.

From now on will refer to each combination we tried with the following schema $xx\_yy\_ww\_zz$ that means :

- **xx :** It is the kind of approach. We will refer to KNN using KNN and to First order difference with FOD.

- **yy :** refers to the QRS KNN approach (w) or to the SSK approach (s).

- **ww :** This can be RS if no preprocessing is applied or FS if the preprocessing is applied.

- **zz :** 1 stands for "First channel selected", 2 stands for "Second channel selected and "B" stands for both channels selected.

2.5.1 Results Comparison

In Table 2.3 the results of each combination are listed, while in Table 2.4 the best results for each algorithm are presented.

From the results obtained we denote that the best combination in terms of Precision, Recall and F1 Score is **KNN_w_FS_1**, i.e the **QRS KNN algorithm** applied to the preprocessed first channel. The novel feature definition effectively improved the performance with respect to the implementation of the approach explained in the paper [3], in our proposed evaluation framework.

The preprocessing module in general, resulted to be effective in all the combinations, except when applying the First-Order Difference algorithm.

Note also that, as expected, an ad-hoc procedure designed to process ECG signals, lead to better results with respect to the ones obtained with a generic signal analysis module.

Combination	Precision	Recall	F1 Score
FOD_RS_1	0,817	0,768	0,792
FOD_RS_2	0,574	0,676	0,621
FOD_FS_1	0,301	0,973	0,46
FOD_FS_2	0,319	0,834	0,461
KNN_s_RS_1	0,347	0,457	0,394
KNN_s_RS_2	0,237	0,313	0,27
KNN_s_RS_B	0,304	0,632	0,411
KNN_s_FS_1	0,371	0,825	0,512
KNN_s_FS_2	0,392	0,686	0,499
KNN_s_FS_B	0,322	0,588	0,416
KNN_w_RS_1	0,926	0,843	0,883
KNN_w_RS_2	0,893	0,744	0,812
KNN_w_RS_B	0,93	0,852	0,889
KNN_w_FS_1	0,988	0,923	0,954
KNN_w_FS_2	0,924	0,807	0,862
KNN_w_FS_B	0,988	0,924	0,955

Table 2.3. Algorithm Performances on all combinations of input

LEGEND:

FOD = First Order Difference
KNN = Knn
w = QRS KNN
s = SSK
RS = Raw Signal
FS = Filtered Signal
1 = First Channel
2 = Second Channel
B = Both First and Second channel

\	**QRS KNN**	**SSK**[3]	**First Order Difference**[24]
Precision	0.988	0.371	0.817
Recall	0.923	0.825	0.768
F1 Score	0.954	0.512	0.792

Table 2.4. Best Precision, Recall and F1 Score for each algorithm

2.5.2 Computational Time Comparison

The computational time of each procedure has been measured on the following devices:

1. Computer:
   - **Processor:** Intel(R) Core(TM) i3-3240 CPU @3.40GHz, 4 cores
   - **RAM:** 4 GB
   - **Operative System:** Ubuntu 16.04
   - **Linux Kernel:** 4.4
2. Raspberry:
   - **Processor:** Arm Cortex-A53 CPU @2.1GHz, 4 cores
   - **Ram:** 1 GB
   - **Operative System:** Android 4.4.2
   - **Linux Kernel:** 4.4
3. Android smartphone:
   - **Processor:** Qualcomm SnapDragon 600 CPU @1.9 GHz, 4 cores
   - **Ram:** 2 GB
   - **Operative System:** Android 4.4.2
   - **Linux Kernel:** 3.4
4. Android smartphone:
   - **Processor:** Arm Cortex-A53 CPU @2.1 GHz, 4 cores, @1.7 GHz, 4 cores
   - **Ram:** 3 GB
   - **Operative System:** Android 7.0 Emotion UI v5.0 Nougat
   - **Linux Kernel:** 4.1.18

   The comparison between the different machines in terms of elapsed time is decribed in Table 2.5

Table 2.5. RAM usage and elapsed time measured in millisecond for each combination

	elasped time (ms)				Ratio Cortex/Intel	Ratio	Ratio	Ratio Cortex 8 cores/	RAM usage (MB)
Processor	Intel(R) Core(TM) i3-3240 CPU @3.40GHz, 4 cores	Arm Cortex-A53 CPU @2.1GHz, 4 cores	Qualcomm Snapdragon 600 CPU @1.9 GHz, 4 cores	Arm Cortex-A53 CPU @2.1GHz, 4 cores @1.7GHz, 4 cores		Qualcomm/Intel	Qualcomm/Cortex 4 cores	Cortex 4 cores	
FOD_RS_1	0.00639	0.0611	0.0603	0.0217	9,56	9,43	0,98	0,35	92,372
FOD_RS_2	0.00651	0.0629	0.0605	0.0273	9,66	9,29	0,96	0,43	91,832
FOD_FS_1	0.000180	0.00274	0.0743	0.00332	15,22	412,77	27,11	1,21	92,424
FOD_FS_2	0.000181	0.00204	0.0789	0.00329	11,27	435,91	38,67	1,61	92,025
KNN_s_RS_1	70.350	466.34	672.51	880.00	6,629	9,559	1,442	1,887	127,594
KNN_s_RS_2	64.616	479.79	698.15	892.00	7,425	10,805	1,455	1,859	120,245
KNN_s_RS_B	87.273	1980.1	1705.1	1983.0	22,689	19,538	0,861	1,001	139,637
KNN_s_FS_1	110.45	1154.3	1036.3	1046.0	10,451	9,383	0,898	0,906	138,639
KNN_s_FS_2	105.55	703.54	1026.4	1060.0	6,665	9,724	1,46	1,507	137,461
KNN_s_FS_B	127.52	2556.8	2178.4	2228.0	20,05	17,083	0,852	0,871	181,993
KNN_w_RS_1	0.0029	0.0219	0.0531	0.0504	7,552	18,31	2,425	2,301	109,227
KNN_w_RS_2	0.0032	0.0211	0.0003	0.0499	6,594	0,93	0,014	2,365	109,637
KNN_w_RS_B	0.0044	0.0392	0.0008	0.0919	8,909	0,18	0,02	2,344	128,618
KNN_w_FS_1	0.0031	0.0212	0.0662	0.0499	6,839	21,355	3,122	2,354	127,34
KNN_w_FS_2	0.0032	0.0212	0.0519	0.0501	6,625	16,22	2,448	2,363	110,14
KNN_w_FS_B	0.0049	0.0379	0.0001	0.0911	7,735	0,02	0,003	2,404	129,03

Given that the signals have a *frequency of sampling* of 360 HZ, it means that each sample is processed every 3 ms ($1/360s \times 1000$). If the elapsed time for one algorithm to process one sample is less than this value, the procedure is executable in **real time**. This is true for what concerns both the **QRS KNN** and the **First Order Difference** even in the slowest machine (*Qualcomm Snapdragon 600*). We also measured the amount of RAM needed to process one signal, which resulted feasible for each machine specification and not affecting too much the performances.

2.6 Critical Issues

In this section, we analyze the cases in which the algorithms fail in the peak detection and we try to understand the reason why that happened.

2.6.1 Example 1: First Order Difference Algorithm

Figure 2.6 shows a detected peak from the *First Order Difference* algorithm (Section 2.1.5) in the signal '102' of the 'MIT-BIH Arrhythmia Database'.

As explained in Section 2.3, this detection is not correct because the peak detected does not reside in the evaluation window, which is of 0.1s around the annotation, which corresponds to 36 samples in this context. Recall that the signal in the plot is the absolute value of the signal normalized; We deduce that, after this transformation, the amplitude of the real peak is lower than the algorithm threshold (see Section 2.4.1) and therefore it cannot be detected.

2.6.2 Example 2: QRS KNN Algorithm

Figure 2.7 shows a detected peak from the QRS KNN algorithm (Section 2.2.3) in the signal '101' of the 'MIT-BIH Arrhythmia Database'.

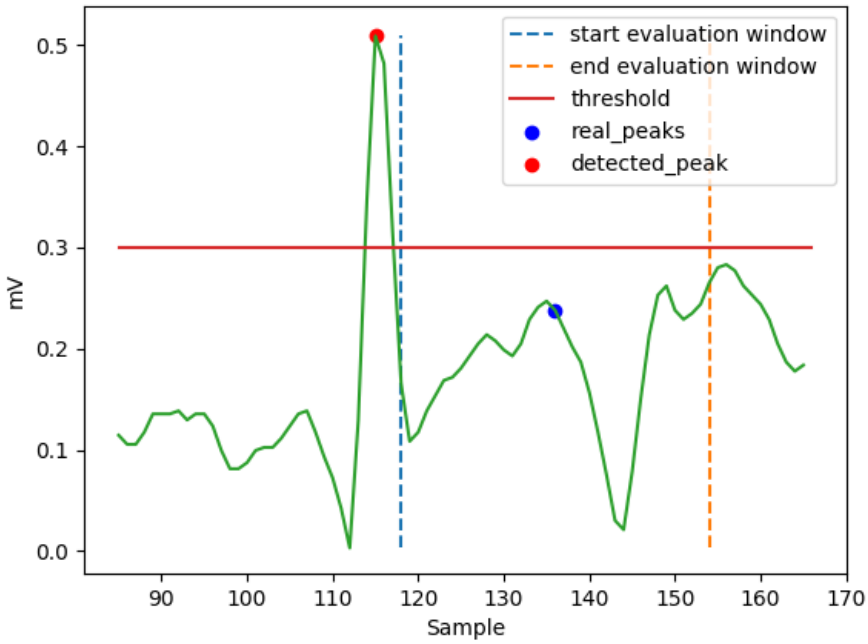


Figure 2.6. First-Order Difference Algorithm Wrong R Peak detection

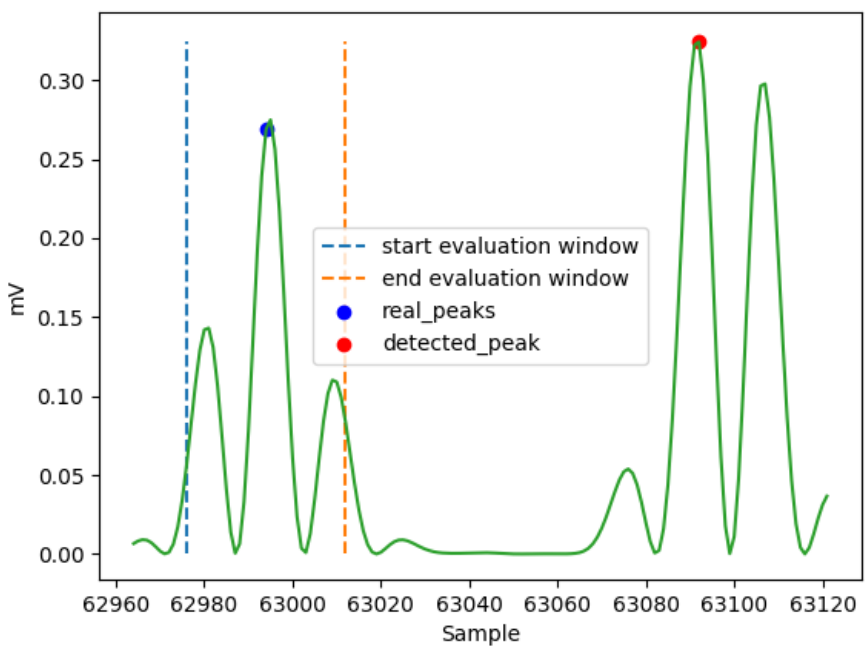


Figure 2.7. KNN Algorithm Wrong R Peak detection

In this plot are depicted the detected R peak and the corresponding real peak, taken from the database annotations. The signal plotted in figure is the result of the filtering and the squaring phases explained in 2.2.1. In order to understand why the algorithm fails in this case, we have to look at the labels present in the neighbors in the training set. Recalling that the signal is first segmented in regions, we have to retrieve the K nearest(according to Euclidean distance in this case) regions to the segment in which the detected peak resides. In this case K=5, and the segments are shown in Figure 2.8:

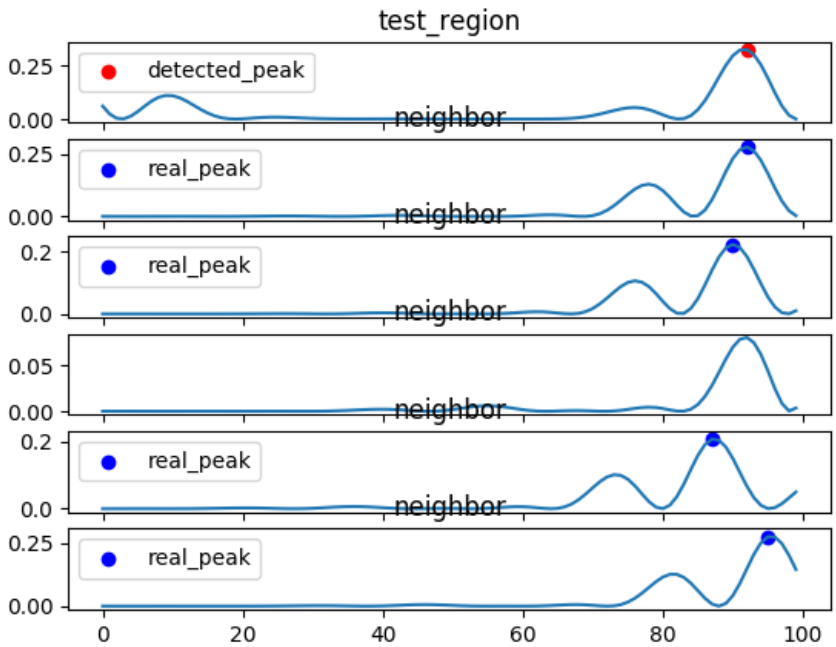


Figure 2.8. KNN of the detected region

Since the strategy adopted for the signal was the 'uniform' (Section 2.2.3.2), this is a simple majority voting. 4 out of 5 neighbors regions contain a peak, so the algorithm decided that the region contains a peak. The detected peak is finally the max of the absolute value of the detected region. In Figures 2.9, 2.10 there is another example of failure in the signal '102'.

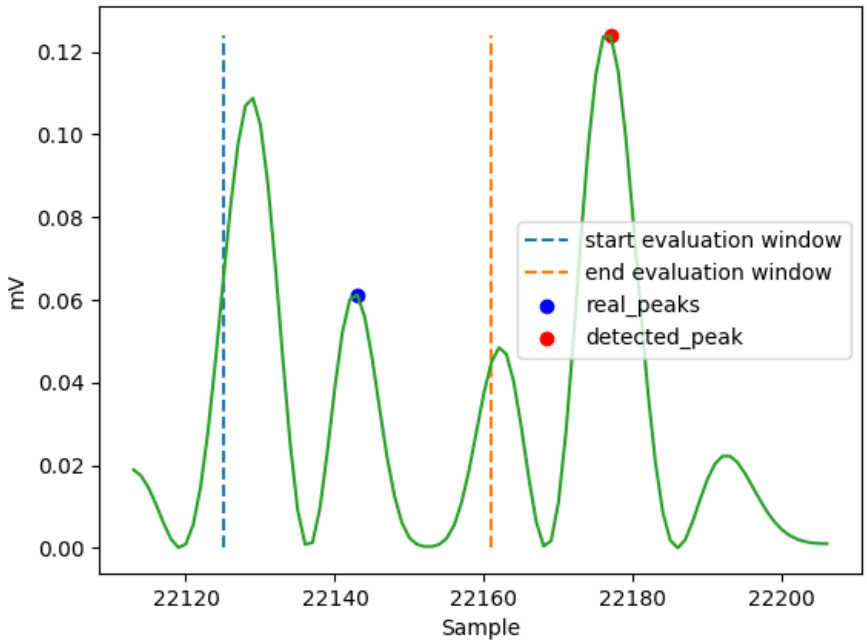


Figure 2.9. KNN Algorithm Wrong R Peak detection

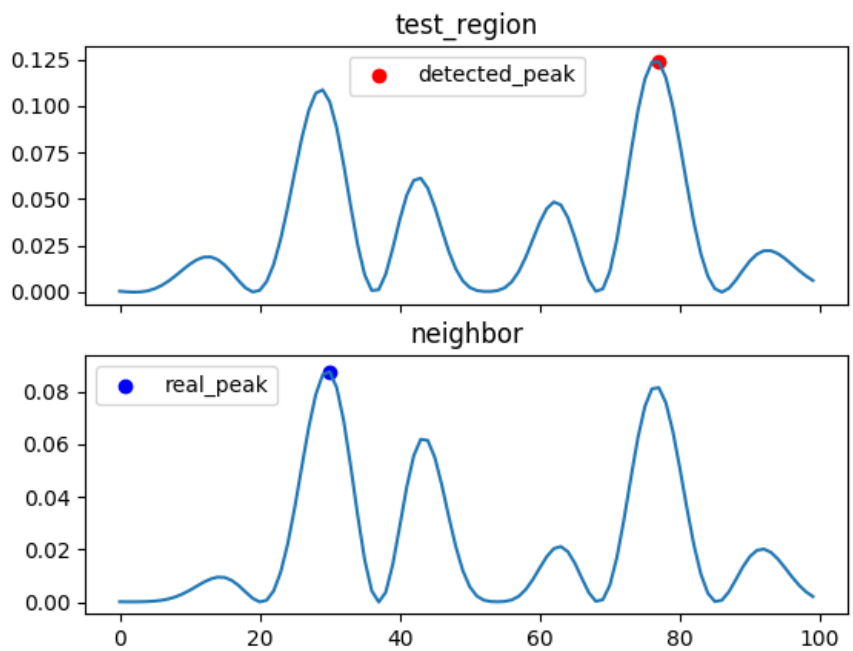


Figure 2.10. KNN of the detected region

Chapter 3

Arrhythmia Classification

Nowadays cardiac arrhythmias are the most common causes of mortality. Early diagnosis of the type of arrhythmia will facilitate proper treatment and thereby a prolonged life [25].

An ECG record is designed to give a graphic display of the electrical activity of the heart. The pattern displayed on the ECG is called **rhythm** [5]. Therefore, the word **arrhythmia** refers to an abnormal heart rhythm.(Section 1.4) Heart arrhythmias result from any disturbance in the rate, regularity, and site of origin or conduction of the cardiac electric impulse [26].

Broadly speaking, arrhythmias can be divided into two groups.[27] The first group includes ventricular fibrillation and tachycardia which are life-threatening and require immediate therapy with a defibrillator. Detection of these arrhythmias is well researched and successful detectors have been designed with high sensitivity and specificity [28] [29].

This study investigates the second group which includes arrhythmias that are not imminently life-threatening but may require therapy to prevent further problems. Some arrhythmias appear infrequently and up to a week of ECG activity may need to be recorded using a Holter ECG monitor to successfully capture them. As desccribed in Section 1.4, many arrhythmias manifest as sequences of heartbeats with unusual timing or ECG morphology.

An important step toward identifying an arrhythmia is the classification of heartbeats. The rhythm of the ECG signal can then be determined by knowing the classification of consecutive heartbeats in the signal. Classification of heartbeats can be very time-consuming and hence any automated processing of the ECG that assists this process would be of assistance and is the focus of this study.

A common paradigm for automatic arrhythmia analysis is the classification in in multiple classes: given an heartbeat, the classifier determines whether it is normal and specifies the typology of arrhythmia, if this is the case.

One major problem faced by today's automatic ECG analysis machine is the wild variations in the morphologies of ECG waveforms of different patients [30]. An ECG beat classifier which performs well for a given training database often fails miserably when presented with a different patient's ECG waveform. Such an inconsistency in performance is a major hurdle preventing highly reliable, fully automated ECG processing systems to be widely used clinically. One obvious approach to alleviate this problem is to use as much training data as possible to develop the ECG classifier. This is the approach taken by all the vendors of ECG processing devices: a large in-house ECG database is developed and maintained to test each ECG processing algorithm to be incorporated into the product. However, such an approach suffers two important pitfalls:

1. No matter how large this database may be, it is not possible to cover every ECG waveform of all potential patients, as explained in Section 1.4. Hence, its performance is inherently limited.

2. The complexity of the classifier grows as the size of the training database grows. When a classifier is designed to correctly classify ECG from millions of patients (if it ever becomes possible), it has to take numerous exceptions into account. The result is a complicated classifier which is costly to develop, maintain, and update.

The answer, according to [30], is to allow the classifier to be **patient-adaptable**. That is, to let the classification algorithm adaptable to the special characteristics of each patient's ECG records. For example, we may include the training algorithm to be delivered to the users, so that the classification algorithm can be fine- tuned with each patient data, designing one classifier for each patient, as we did in Section 2.2.3.1. But one problem still occurs. If we consider as training set a single ECG tracing from one single patient, some of the arrhythmic classes may be absent in the training data, resulting in a very poor generalization on unseen recordings. For this reason, an intra patient paradigm similar to the one in Section 2.2.3.1 must be discarded in this stage.

To deal with this problem, we propose a **vertical split**: the training dataset has been defined to include a fraction of heartbeats from each ECG record. A smaller fraction is then considered as validation dataset, in order to tune the hyperparameters of the model, and the remaining beats are used for the performance evaluation, belonging to the test set.

This study demonstrates that **a single classifier** trained on a reasonable amount of heartbeats from one or more patients, it is able to make correct predictions on **unseen heartbeats** from the same source of data.

We implemented two different models of **Recurrent Neural Networks**, using the *MIT-BIH Arrhythmia Database* [23] for training, validation and evaluation purposes. The AAMI classes [4] for evaluation are considered in this context and our results have been compared to those of [30] and [27].

3.1 State of the Art

3.1.1 Reservoir Computing and Logistic Regression

In [34] they employed a machine learning paradigm named reservoir computing (RC)[41] [42]. RC mimics brain neural networks by processing information that generates patterns of transient neural activity as a response to a sensory signal and is composed of layers for processing the information. They used a kind of reservoir consisting of a nonlinear dynamical element subject to delayed feedback. One of the simplest delay system is a single nonlinear node influenced by its own dynamics after a certain delay time τ. A ring topology network of 25 such nodes has been employed, with the output of a node influencing the following ones. The network processes the beats extracted from the *MIT-BIH arrhythmia Database* without any preprocessing. Each beat is processed in a serial manner, sample-by sample; The outcome of each node is then fed to the output layer of the network, which performed a simple *Logistic Regression*, classifying each beat in one of the five classes defined by the *Association for the Advancement of Medical Instrumentation*[4]. They followed the partition of the database proposed by [27], consisting in a *inter-patient* paradigm (Section 2.2.2.1). They obtained good results in terms of average Sensitivity/Recall = 84.3% and Precision=88.75%. We tried to reimplement this approach, but we didn't succede in obtaining good results, since it was not clear how they modeled the dependency between the nodes outcomes in the ring topology. From this we kept the concept of **dependency** between subsequent beats, and the beat extraction techinque from the ECG tracing(see Section 3.3.1).

3.1.2 Weighted SVM

In [39] they proposed a **feature based** approach followed by a classification step performed by a *SVM* classifier. They considered several groups of features extraced from each heartbeat that can be summarized in seven major categories:

- **R-R intervals**: This group consists of three features built from the R-R interval series. The first three features are the R-R interval to the previous beat, the R- R interval to the next beat and the average of R-R intervals in a window of 10 surroundings beats.

- **Segmentation intervals**: A large variety of 24 features are computed from the annotated characteristic points. These features include a boolean flag indicating the presence or absence of QRS, P and T waves. If the waves are present, their duration, maximum and minimum values, area, standard deviation, skewness and kurtosis are computed as features. The Q-T, S-T, Q-R, R-S intervals are also included. When the characteristic points needed to compute a feature failed to be detected in the heart beat annotation step, the feature value is set to the patient's mean feature value.

- **Morphological features**: Ten features are derived by uniformly sampling the ECG amplitude in a window defined by the onset and offset of the QRS complex, and nine other features in a window defined by the QRS offset and the T-wave offset. As the ECG signals were already sampled, linear interpolation

was used to estimate the intermediate values of the ECG amplitude. Here again, when the onset or offset points needed to compute a feature were not detected, the feature value is set to the patient's mean feature value.

- **HBF coefficients**: The parameters for computing the HBF expansion coefficients as defined in [43] are used. The order of the Hermite polynomial is set to 20, and the width parameter ÏČ is estimated so as to minimize the reconstruction error for each beat.Weighted SVMs and Feature Relevance Assessment 219

- **Higher order statistics**: The 2nd, 3rd and 4th order cumulant functions are computed. The parameters as defined in [40] are used: the lag parameters range from -250 msec to 250 msec centered on the R spike and 10 equally spaced sample points of each cumulant are used as features, for a total of 30 features.

- **Normalized R-R intervals**: These features correspond to the ratio between the previous three R-R values and their mean value for this patient. These last features are thus independent from the mean normal behavior of the heart of patients, which can naturally be very different between individuals, possibly misleading the classifier.

- **Normalized segmentation intervals**: This group of features contains the same features as the previous segmentation group, but the values are normalized by their mean value for each patient. The normalization is obviously not applied to boolean segmentation features. Here again, the objective is to make each feature independent from the mean behavior of the heart of a patient, because it can naturally be very different between individuals

They performed an exhaustive search of all the possible combinations of feature groups. Their results indicate that **R-R intervals, normalized R-R intervals and HOS features** are the most important feature sets to include in the model. In order to mitigate the class unbalancing (see Section 3.3.2) of the *MIT-BIH arrhythmia Database*, they propose a **weighted** version of the SVM classifier. The cost function has been weighted in order to take into account the distribution of the training set: a wrong prediction on a least represented class is penalized more than a mistake on the normal class.

They reported just the results in terms of *Accuracy* and *Average Accuracy* between the 4 classes in the *aami* definition[4]. The accuracy measure is not relevant in an unbalanced dataset, beacuse of the **Accuracy paradox**.The accuracy paradox is the paradoxical finding that accuracy is not a good metric for predictive models when classifying in predictive analytics. This is because a simple model may have a high level of accuracy but be too crude to be useful. For example, if the incidence of category A is dominant, being found in 99% of cases, then predicting that every case is category A will have an accuracy of 99%. Precision and recall are better measures in such cases. We reimplemented this approach but we did not succed in obtaining good results according to the non reported measures of *Precision* and *Recall*.

3.1.3 Mixture of Experts

In [30] a **mixture-of-experts (MOE)** approach to develop customized electrocardigram (ECG) beat classifier is presented. A small customized classifier is developed based on brief, patient-specific ECG data. It is then combined with a global classifier, which is tuned to a large ECG database of many patients, to form a MOE classifier structure. This proposed approach is based on three popular artificial neural network (ANN)-related algorithms, namely, the self- organizing maps (SOM), learning vector quantization (LVQ) algorithms, along with the mixture-of-experts (MOE) method. SOM and LVQ together are used to train the patient-specific classifier, and MOE is a paradigm which facilitates the combination of the two classifiers (original and patient-specific) to realize patient-adaptation. In MOE, the two classifiers are modeled as two experts on ECG beat classification. The original classifier, called the **Global expert (GE)** in this work, knows how to classify ECG beats for many other patients whose ECG records are part of the in-house, large ECG database. The patient-specific classifier, called the **local expert (LE)** in this work, is trained specifically with the ECG record of the patient. A **gating function**, based on the feature vector presented, dynamically weights the classification results of the GE's and the LE's to reach a combined decision. The process is analogous to two human experts arriving at a consensus based on their own expertise. In this study they concentrated on the classification of ventricular ectopic beats (VEB's). The 48 records (tapes) from MIT/BIH ECG arrhythmia database [23] are used for the development and evaluation of the classifier. Since such study is to evaluate the performance of a classifier that can identify a **premature ventricular contraction (PVC)** (Section 1.4.3), certain records in the database with no PVC's (11 records) were excluded from the study, leaving 33 records of interest. the fully automatic configuration (GE classifier) approach obtained results are: accuracy 75.3%, recall 69.6% and precision 34.6% ;while the results obtained with their semi-automatic MOE approach: (accuracy 93.6%, recall 78.9% and precision 76.0%).

3.1.4 ECG Morphology and Heartbeat Interval Features

The aim of the study in [27] was to design and test an automatic classification system using a comprehensive ECG database following AAMI recommended practice [4]. Methodology improvements on previous approaches included:

- trialing eight representations of the ECG morphology;
- use of the heartbeat classes recommended in [4];
- comparison of 12 classifier configurations processing feaures obtained from single and multiple ECG leads;
- weighting the training examples to prevent the large classes from dominating the training process.

Features relating to fiducial point intervals were calculated for each heartbeat. Features relating to heartbeat intervals and ECG morphology were calculated separately for the two ECG signals for each heartbeat. Figure 3.1 lists the features used in this study.

Group Label	Features
RR intervals	• Pre-RR interval • Post-RR interval • Average RR-interval • Local avg. RR-interval
Heart-beat intervals A [B]	• QRS duration (QRS offset -QRS onset) of lead A [B] • T-wave duration (T-wave offset - QRS offset) of lead A [B] • P wave flag for lead A [B]
Morphology 1A [1B]	• ECG morphology (10 samples) between QRS onset and QRS offset of lead A [B] • ECG morphology (9 samples) between QRS offset and T-wave offset of lead A [B]
Morphology 2A [2B]	• Normalised ECG morphology (10 samples) between QRS onset and QRS offset of lead A [B] • Normalised ECG morphology (9 samples) between QRS offset and T-wave offset of lead A [B]
Morphology 3A [3B]	• ECG morphology (10 samples) between FP-50ms to FP+100ms of lead A [B] • ECG morphology (8 samples) between FP+150ms to FP+500ms of lead A [B]
Morphology 4A [4B]	• Normalised ECG morphology (10 samples) between FP-50ms to FP+100ms of lead A [B] • Normalised ECG morphology (8 samples) between FP+150ms to FP+500ms of lead A [B]

Figure 3.1. Group of features considered in [27]

Classifier models based on linear discriminants (LDs) were utilized throughout this study. The model parameters were determined using "plug-in" maximum-likelihood estimates calculated from the training data. For the Supraventricular Ectopic Beats(SVEB) class, the recall was 75.9%, the precision was 38.5%. The recall was 77.7% and the precision was 81.9% for the VEB class.

3.2 Recurrent Neural Network

Countless learning tasks require dealing with sequential data. Image captioning, speech synthesis, and music generation all require that a model produce outputs that are sequences. In other domains, such as time series prediction, video analysis, and musical information retrieval, a model must learn from inputs that are **sequences** [31].

Recurrent Neural Networks (RNNs) are connectionist models that capture the dynamics of sequences via cycles in the network of nodes. Unlike standard feedforward neural networks, recurrent networks retain a state that can represent information from an arbitrarily long context window.[31] In a traditional neural network we assume that all inputs (and outputs) are independent of each other. But if the task consists in predicting the next word in a sentence it is useful to know which words came before it. RNNs are called recurrent because they perform the same task for every element of a sequence, with the output being **dependent** on the previous computations.[32]

For this reason it seemed reasonable to apply this approach for the **Arrhythmia classification** purpose: ECG data is **sequential** and determining whether a heartbeat is arrhythmic or not **depends** not only on its morphology but also on those of the previous beats (Section 1.4). In many approaches this dependency is modeled by considering the *RR interval* between heartbeats [33]; In this context this idea is represented by the structure of a particular kind of RNN, such as the **Long Short Term Memory**(LSTM).

3.3 Preprocessing

3.3.1 Input Preparation

This module takes as input the outcome of the previous stage of R peak detection. Arrhythmia detection requires the extraction of the beats from the R peak locations. A beat is defined as the amount of signal necessary to include the P, Q and T waves. We can define one beat as the signal around an R peak location.In this way, we extract one heartbeat from every R peak location. Furthermore, the LSTM network requires a tridimensional shaped input organized as:

$$input\_shape = (n\_beats, timesteps, n\_samples)$$

Where *n_beats* is the total number of detected beats, *timesteps* is the amount of previous beats to consider in order to make a prediction, *n_samples* is the amount of samples which constitute a beat. In other words, the sequence of beats is split in *sliding windows*, each of which is composed by t beats, where t corresponds to the *timesteps* value.

3.3.2 Dataset Rebalancing

In a real world scenario, a dataset composed by individuals ECG recording would likely be characterized by many normal beats and a smaller set of arrhythmic ones. In this case we are dealing with an **unbalanced dataset**, which would lead in a bias in the training stage. A classifier that is trained on a very unbalanced dataset may likely end up in learning too much the features of the instances of the most prominent class, almost completely ignoring the characteristics related to the other classes. Many attempts are known in order to overcome this issue, such as the adoption of a penalized model, and the techinques of data undersampling/oversampling.[35]

In this approach we implemented a simple algorithm to generate synthetic data in order to rebalance the dataset classes distribution, **augmenting** the number of samples in the least populated ones. We augmented the samples in the under-represented class c with the following:
$\forall\ beat\ b \in c:$

$$r_i = random\ number \in (0, 1)$$
$$r = random\ vector\ s.t.\ size(r) = size(b)\ and\ r_j = r_i,\ \forall j \in [0, size(r)]$$
$$synthetic\_sample = b + r$$
$$X\_train = X\_train \cup syntethic\_sample$$

Where X_train is the set of beats for training, r_i is the generic element of the random vector r and the operator $+$ stands for the vector addition.

This procedure has been adopted since the shape is the relevant feature in order to discriminate between arrhythmic beats. Adding the same value r_i to each sample in the beat would in fact preserve the shape, providing more training data to the network, as if it was new data.

A naive oversampling technique consists in simply replicating the instances of the minority classes without any further processing. By the way, this method may lead the network to *overfit* on such data, memorizing the specific amplitudes without capability of generalization on unseen data:for this reason this randomized approach has been preferred.

As another attempt, we tried to **reduce** the number of instances belonging to the over-represented class, i.e. the Normal(N). We implemented a *random sampling technique*, which consists in randomly select a subset of hearbeats in the normal class. The cardinality of the normal class is reduced by a *reduction factor*, more precisely:

$$|N_r| = \frac{|N|}{r}$$

where || denotes the cardinality, N is the set of instances initially belonging to the normal class, N_r is the reduced set and r is the reduction factor.

3.3.3 Standardization

Most Machine learning models requires that the input data has a Gaussian (bell curve) distribution. **Data standardization** is the process of rescaling one or more attributes so that they have a mean value of 0 and a standard deviation of 1. In order to obtain zero mean, the actual mean value of the data is subtracted to each sample. For this, such techinque is also known as *data centering.* A sample x is standardized as follows:

$$standardize(x) = \frac{x-\mu}{\sigma}$$

where μ is the mean and σ the standard deviation.

3.4 Network Structure

Given an input sequence $\mathbf{x} = (x_1, ..., x_T)$ a standard recurrent neural network(RNN) computes the **hidden vector** sequence $\mathbf{h}$ =$(h_1, ..., h_T)$ and the output vector sequence $\mathbf{y} = (y_1, ..., y_T)$ by iterating the following equations from t=1 to T:

$$h_t = H(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

Where the W terms denote weight matrices (e.g. W_{xh} is the input-hidden weight matrix), the b terms denote bias vectors(e.g. b_h is hidden bias vector) and H is the hidden layer function, usually a tanh.[36]

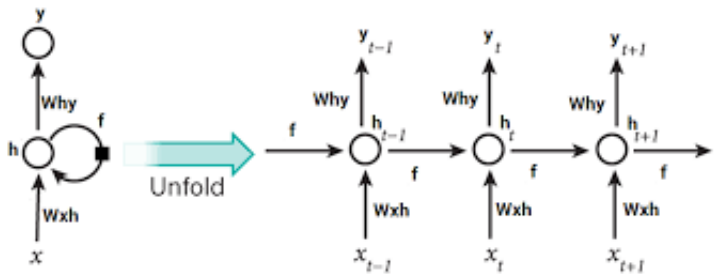


Figure 3.2. Basic RNN cell structure, source:[32]

Recurrent Neural Networks suffer from **short-term memory problem**. If a sequence is long enough, carrying information from earlier time steps to later ones results difficult. During back propagation, recurrent neural networks suffer from the **vanishing gradient problem**. Gradients are values used to update a neural networks weights. The vanishing gradient problem is when the gradient shrinks as it back propagates through time. If a gradient value becomes extremely small, it doesn't contribute too much learning.

These problems have been solved with the design of a variant of the RNN, namely the **LSTM**. The **Long Short Term Memory**(LSTM) architecture uses purpose built **memory cells** to store information, and resulted better at finding and exploiting long range context[37]. An LSTM has a similar control flow as a recurrent neural network. It processes data passing on information as it propagates forward. The differences are the operations within the LSTM's cells.[38]

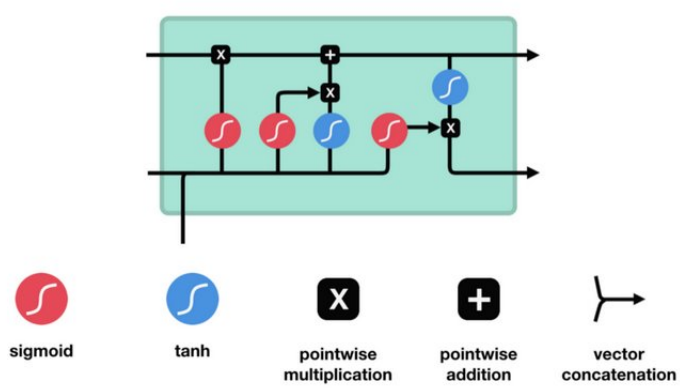


Figure 3.3. LSTM Cell and It's Operations, source:[38]

The core concept of LSTM's are the **cell state**, and the various **gates**. The cell state act as a transport highway that transfers relative information all the way down the sequence chain. It is the **memory** component of the network. So even information from the earlier time steps flows to later time steps, reducing the effects

of short-term memory. Gates contains **sigmoid** activations: A sigmoid activation is similar to the tanh activation. Instead of squishing values between -1 and 1, it squishes values between 0 and 1. That is helpful to update or forget data beacuse zero multiplications cause values to disappear(*forgotten*), while one multiplications allows values to be *kept*. The network can learn which data is not important therefore can be forgotten or which data is important to keep.

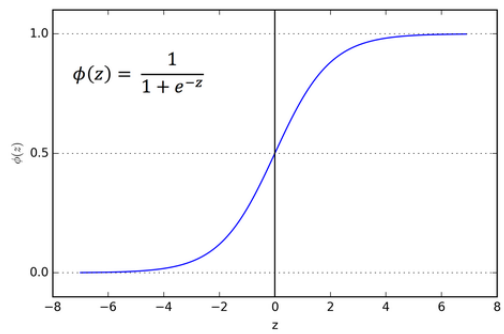


Figure 3.4. Sigmoid Activation function

The **forget gate** decides what information should be thrown away or kept. Information from the previous hidden state and information from the current input is passed through the sigmoid function. Values come out between 0 and 1. The closer to 0 means to forget, and the closer to 1 means to keep.

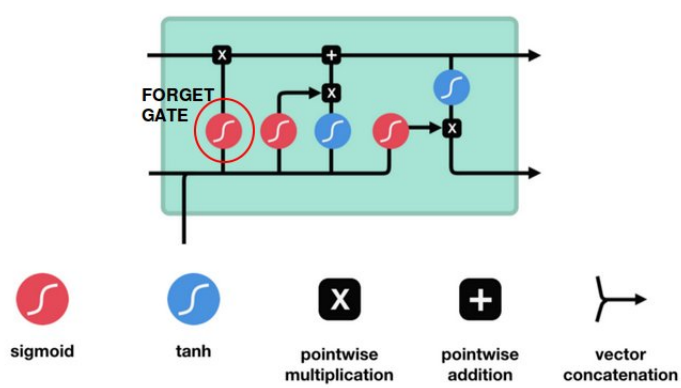


Figure 3.5. Forget gate, source: [38]

The **input gate** manages the update of the cell state. First,the previous hidden state and current input are processed by a sigmoid function. That decides which values will be updated by transforming the values to be between 0 and 1. 0 means not important, and 1 means important. The hidden state and current input are given as input to the tanh function which outputs values between -1 and 1. The tanh

output gets multiplied with the sigmoid output. The sigmoid output will decide which information is important to keep from the tanh output.

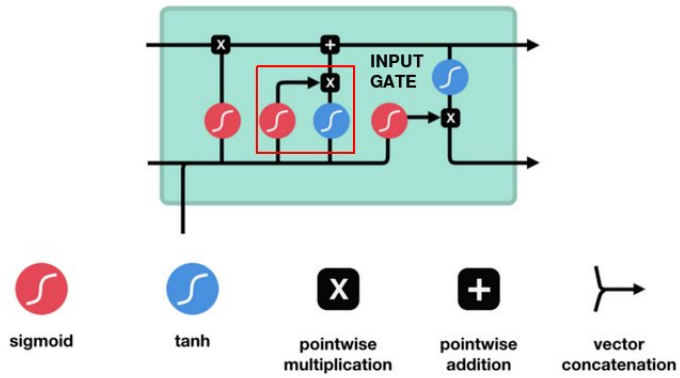


Figure 3.6. Input gate, source: [38]

The **cell state** gets pointwise multiplied by the forget vector. This has a possibility of dropping values in the cell state if it gets multiplied by values near 0. Then the output from the input gate are taken and a pointwise addition is performed. This last operation updates the cell state to new values that the neural network finds relevant.

Finally the **output gate** decides what the next hidden state should be. The hidden state contains information on previous inputs and is also used for predictions. First, the previous hidden state and the current input are passed into a sigmoid function. Then the newly modified cell state is processed by the tanh function. The final multiplication between the tanh output and the sigmoid output determines what information the hidden state should carry. The output is the hidden state. The new cell state and the new hidden is then carried over to the next time step.

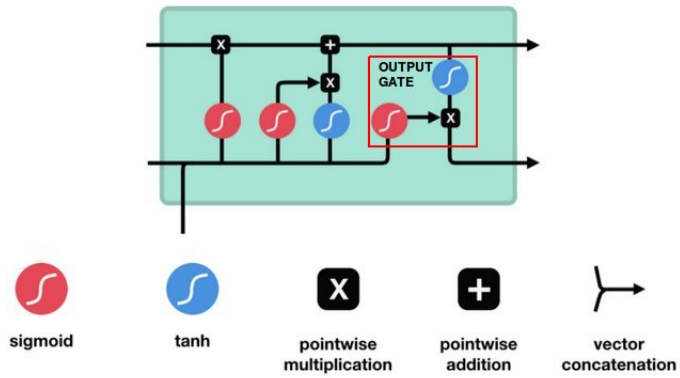


Figure 3.7. Output gate, source: [38]

3.4.1 Neural Networks Activation Functions

The main activation function that was widely used is the Sigmoid function however, when the **Rectifier Linear Unit(ReLU)** was introduced[46], it soon became a better replacement for the Sigmoid function due to its positive impact on the different machine learning tasks. The ReLU activation function has the following form [47]:

$$ReLU(x) = max(0, x)$$

which has derivative:

$$\frac{d}{dx}ReLU(x) = \begin{cases} 0 & if\ x \leq 0 \\ 1 & otherwise \end{cases}$$

It comes with the aim to solve the **vanishing gradient and exploding gradient problems**: while using Sigmoid and working on shallower layers doesn't give any problem, some issues arise when the architecture becomes deeper because the derivative terms that are less than 1 will be multiplied each other many times that the values will become smaller and smaller until the gradient tends towards zero hence *vanishing*. On the other hand if the values are bigger than 1 then the opposite happens, with numbers being multiplied becoming bigger and bigger until they tend to infinity and *explode* the gradient.A good solution would be to keep the values to 1 so even when they are multiplied they don't change. This is exactly what ReLU does: it has gradient 1 for positive inputs and 0 for negative ones.

The second activation function to be examined is the **Exponential Linear Unit(ELU)** [48] which is given as:

$$elu(x) = \begin{cases} \alpha(exp(x) - 1) & if\ x \leq 0 \\ x & otherwise \end{cases}$$

which has the gradient:

$$\frac{d}{dx}elu(x) = \begin{cases} elu(x) + \alpha & if\ x \leq 0 \\ 1 & otherwise \end{cases}$$

where $\alpha = 1$ ELU is a function that tend to converge cost to zero faster and produce more accurate results. Different to other activation functions, ELU has a extra alpha constant which should be positive number. ELU is very similiar to RELU except negative inputs. They are both in identity function form for non-negative inputs. On the other hand, ELU becomes smooth slowly until its output equal to $-\alpha$ whereas RELU sharply smoothes.

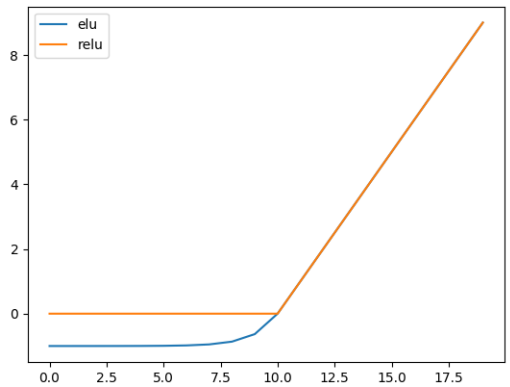


Figure 3.8. ReLU and ELU activations

ReLU and ELU activations functions are plotted in Figure 3.8

3.5 Training and Validation

The dataset has been partitioned in the training, validation and test sets according to a **vertical split**: the training dataset has been defined to include a fraction of heartbeats from each ECG record. A smaller fraction is then considered as validation dataset, in order to tune the hyperparameters of the model, and the remaining beats are used for the performance evaluation, belonging to the test set. The first 50% of each ECG signal is used for training the network, the subsequent 10% for validation and the last 40% for testing. Therefore, a single LSTM model has been trained and validated on a fraction of heartbeats from all the records in the database.Records corresponding to patients with pacemaker (102,104,107,217) have been excluded the database, as in [27]. Figure 3.9 illustrates, for simplicity, the dataset partition on three ECG tracings of 8s duration.

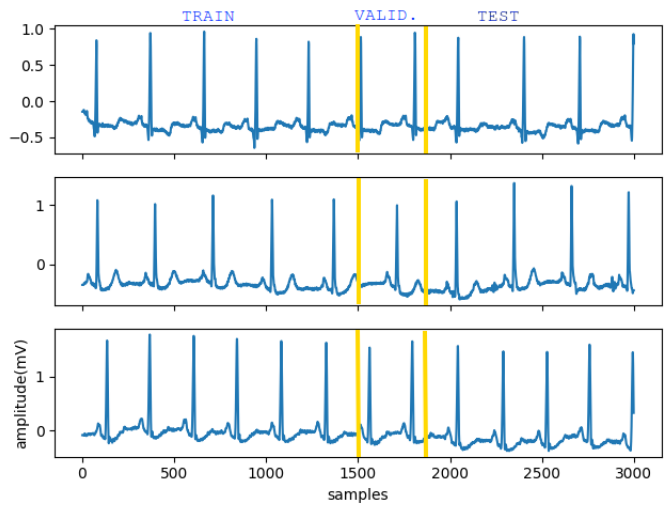


Figure 3.9. Vertical split partition example on 8s duration ECG tracings

3.6 Evaluation

According to the Association for the Advancement of Medical Instrumentation (AAMI) standard [4] for testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms, the evaluation of an Arrhythmia detector algorithm is on the classification performance on 4 major classes of heartbeats:

- **N-class** includes beats originating in the sinus node (normal and bundle branch block beat types);
- **S-class** includes supraventricular ectopic beats;
- **V-class** includes ventricular ectopic beats (VEBs);
- **F-class** includes beats that result from fusing normal and VEBs.

Furthermore, each class contains more fine-grained classes of arrhythmic beats. In the table below are listed the complete classes definitions:

Table 3.1. Beat classification Categories

CLASSES	N	S	V	F
DEFINITIONS	normal beat	atrial premature beat	premature ventricular contraction	fusion of ventricular and normal beat
	left bundle branch block beat	aberrated atrial premature beat		
	right bundle branch block beat	nodal (junctional) premature beat	ventricular escape beat	
	atrial escape beats	supraventricular premature beat		
	nodal (junctional) escape beat			

3.6.1 Performance Measures

The measures considered in this study are the **per-class Precision, per-class Recall, per-class F1 Score, Average Precision, Average Recall and Average F1 Score**, with the following definitions:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}$$

$$Recall_c = \frac{TP_c}{TP_c + FN_c}$$

The definitions of TP, TN, FP and FN for each class are presented in Table 3.2

$$F1\ Score_c = 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c}$$

where c is the class index, and TP_c, FP_c are the true positives and false positives fro what concerns the class c

$$Average\ Precision = \sum_{c=0}^{3} \frac{TP_c}{TP_c + FP_c}$$

$$Average\ Recall = \sum_{c=0}^{3} \frac{TP_c}{TP_c + FN_c}$$

$$Average\ F1\ Score = \sum_{c=0}^{3} 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c}$$

predicted/real	**N**	**S**	**V**	**F**
N	TP_N	$FN_S \& FP_N$	$FN_V \& FP_N$	$FN_F \& FP_N$
S	$FN_N \& FP_S$	TP_S	$FN_V \& FP_S$	$FN_F \& FP_S$
V	$FN_N \& FP_V$	$FN_S \& FP_V$	TP_V	$FN_F \& FP_V$
F	$FN_N \& FP_F$	$FN_S \& FP_F$	$FN_V \& FP_F$	TP_F

Table 3.2. TP, FP, TN, FN definition for Arrhythmia Detection

3.6.2 Database Annotations

The database considered in this study is again the the **MIT-BIH Arrhythmia Database**[23]. In Figure 3.10 it is presented a listing of the beat annotations made by cardiologists on the MIT-BIH Arrhythmia Database.

Table 3.3 shows the distribution of heartbeats in the 4 AAMIs evaluation classes[4]

Symbol	Meaning
· *or* N	Normal beat
L	Left bundle branch block beat
R	Right bundle branch block beat
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature beat
V	Premature ventricular contraction
F	Fusion of ventricular and normal beat
[	Start of ventricular flutter/fibrillation
!	Ventricular flutter wave
]	End of ventricular flutter/fibrillation
e	Atrial escape beat
j	Nodal (junctional) escape beat
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
x	Non-conducted P-wave (blocked APB)
Q	Unclassifiable beat
|	Isolated QRS-like artifact

Figure 3.10. MIT-BIH Arrhythmia Database Beat Annotations

Class	**N**	**S**	**V**	**F**
Instances	90585	2781	7235	802

Table 3.3. Distributions of heartbeats according to the AAMI recommendations

3.7 Protocol Fine-Tuning

The **validation** stage is devoted to search the best combination of hyperparameters, that is the one which results in the best performance on the validation set.

We defined the best combination as the one with the best **Average F1 Score**, Section 3.6.1, since it is it is a measure that takes into account both the precision and the recall of the result for each class.

The following hyperparameters have been fine-tuned:

- Window size
- Number of Channels

- augmentation factors
- reduction factors
- Timesteps
- Number of LSTM layers
- Number of fully connected layers
- Number of neurons for each layer
- Batch Size
- Dropout Probability
- Hidden Units Activation Function

The **Adam** has been chosen as optimization algorithm. The loss function selected is the **categorical cross entropy** [44], defined as:

$$-\sum_{c=1}^{M} y_{o,c} log(p_{o,c})$$

where M is the number of classes, log the natural log, y is a binary indicator(0 or 1) if class label c is the correct classification for observation o, p is the predicted probability observation o is of class c. The **Number of epochs** parameter varied for each combination: an *Early Stopping Criterion* has been employed; the model stops training when the value of the loss function does not decrease for 10 epoch in the validation step. Once this happens, the weights of the network corresponding to the lowest loss value are restored.

Since an exhaustive Grid Search was not feasible to perform due to the elevated number of candidate combinations, each parameter has been tuned in isolation mantaining the others fixed. Once a parameter value has been chosen, such decision is kept for all the subsequent runs.

3.7.1 Window Size

In Section 3.3.1, we defined an heartbeat a fixed amount of signal around an R Peak location. Such amount is not clearly defined in literature so we decided to try different values. Another idea consists in defining the feature vector as a bigger portion of the signal, including more than one heartbeat. The result of this experiment for the varying window sizes of 200, 300, 400, 500 are illustrated in Figure 3.11

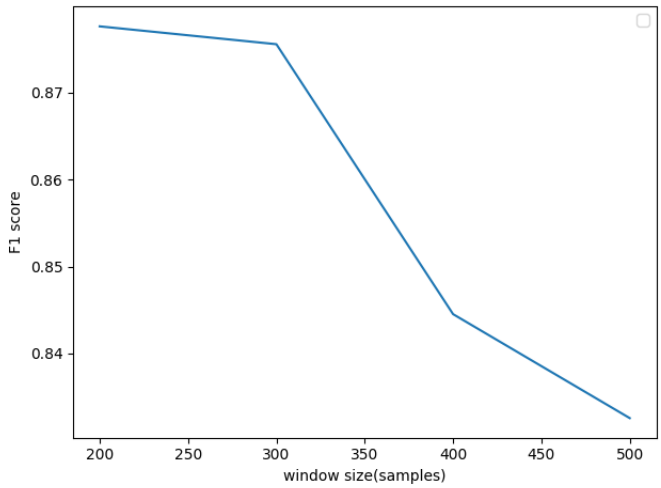


Figure 3.11. Average F1 Score with different Window Sizes

The best value in terms of Average F1 Score on the validation dataset is 200 samples. This means that the best result is obtained with the feature vector composed by one heartbeat.

3.7.2 Number of Channels

As in Section 2.4.3.3 we want to analyze whether the inclusion of the second channel improves the results. In the configuration that uses both channels, the data from the two signals is concatenated in one feature vector, of doubled dimension.

In Figure 3.12 are depicted the results obtained with one channel and two channels separately, and the ones obtained by the combination of both in terms of Average F1 Score

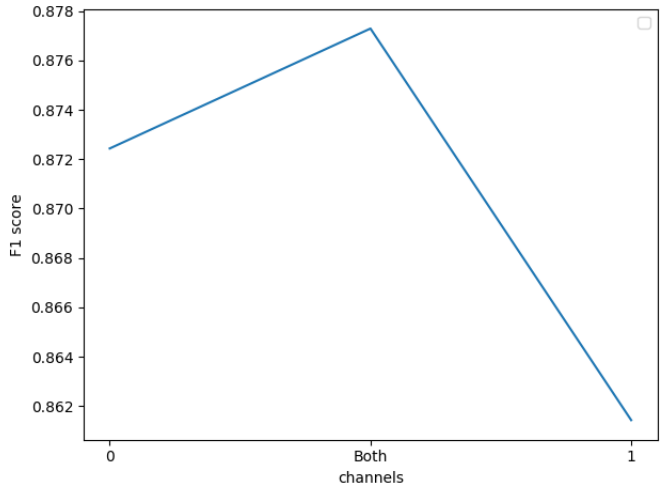


Figure 3.12. Average F1 Score with First Channel, Second Channel and Both

From this we deduce that the inclusion of Both channels improves the classification results.

3.7.3 Augmentation and Reduction Factors

The least represented classes in the MIT-BIH Arrhythimia Database are the **S-class** and the **F-class**, while the over-represented one is the normal class, as depicted in Table 3.3. In order to mitigate the class unbalancing problem we performed a dataset rebalancing techniques (Section 3.3.2). At this point we must define the **augmentation and reduction factors**, i.e. how many times to replicate each heartbeat in the training dataset, belonging to the least represented classes.

In Figure 3.13 are depicted the results obtained with the augmentation factors of None(no augmentation at all); (S:4, F:9), (S:3, F:8), (S:4, F:8) and (S:3, F:9) and the reduction factors of None(no reduction at all); (N:2), (N:3) and (N:4)

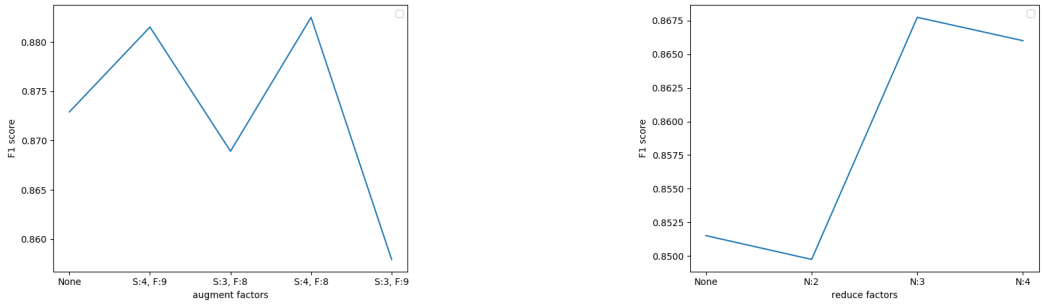


Figure 3.13. Average F1 Score with varying reduction and augmentation factors

From this we deduce that the best augment/reduction factors are (S:4, F:8) and (N:3)

3.7.4 Timesteps

As explained in Section 3.3.1, the **timesteps** value is the amount of previous beats to consider in order to make the prediction for the current beat.

In Figure 3.14 are depicted the results obtained with the timesteps values of 2, 3, 4 and 5

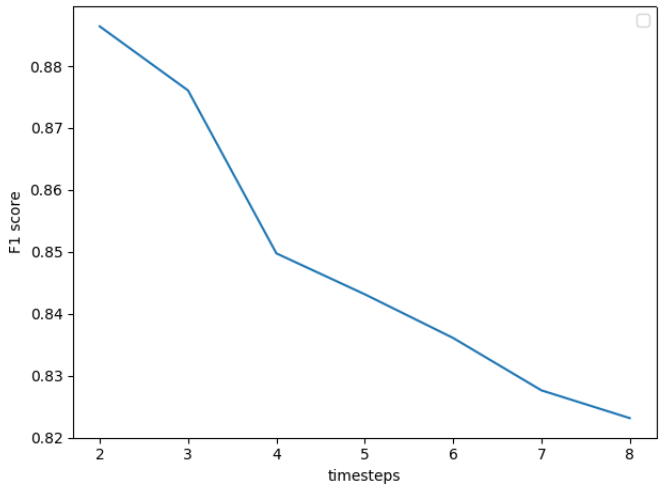


Figure 3.14. Average F1 Score with varying timesteps

From this we deduce that the best timesteps value is 2

3.7.5 Network Layers

For what concerns the Network layers, we must determine how many LSTM layers to concatenate in a sequential manner in our model. Then, we determined whether the inclusion of standard Fully Connected Layers after the LSTMs improves the results. Furthermore, the activation between the *elu* and the *relu*(See Appendix 3.4.1 for further reading) and the number of neurons for each layer has been selected. Finally, we tried to apply *Dropout* regularization layers between the Fully Connected layers. The Dropout Probabilities considered are None(no Dropout layers), 0.1, 0.3, 0.5, 0.8. The Batch size parameter also influences the correctness of the results; it is the number of instances after which the update of parameters is performed. We tuned such value in the space of 32, 64, 128, 256.

All the results of such experiments are depicted in Table 3.4

From this we deduce that the optimal number of LSTMs and Fully Connected layers are respectively 4 and 3, the inclusion of Dropout layers with Probability of 0.1 improved the F1 Score, the best Activation Function is the *elu* and the best Batch Size is of 128.

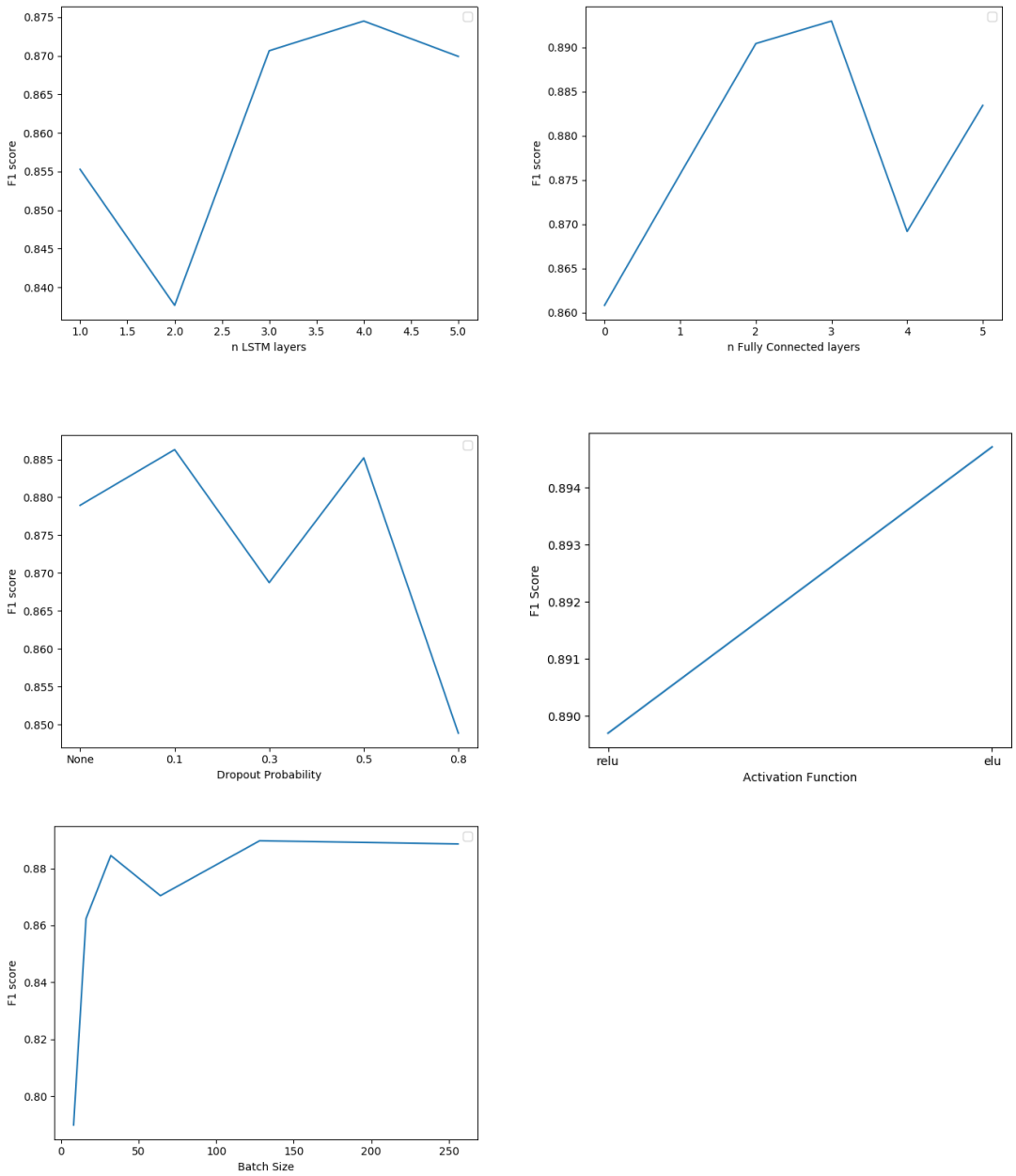


Table 3.4. Average F1 Score with varying Number of LSTMs, Fully Connected layers, Dropout Probablity, Activation Function and Batch size

3.8 Results

The validation step determines which is the set of hyperparameters that obtained the best performance results on the validation dataset. The parameters related to the Network Structure allow us to build our Recurrent Neural Network, depicted in Figure 3.15, as displayed in *TensorBoard*[45], a graphical tool for visualizing and debugging Neural Networks

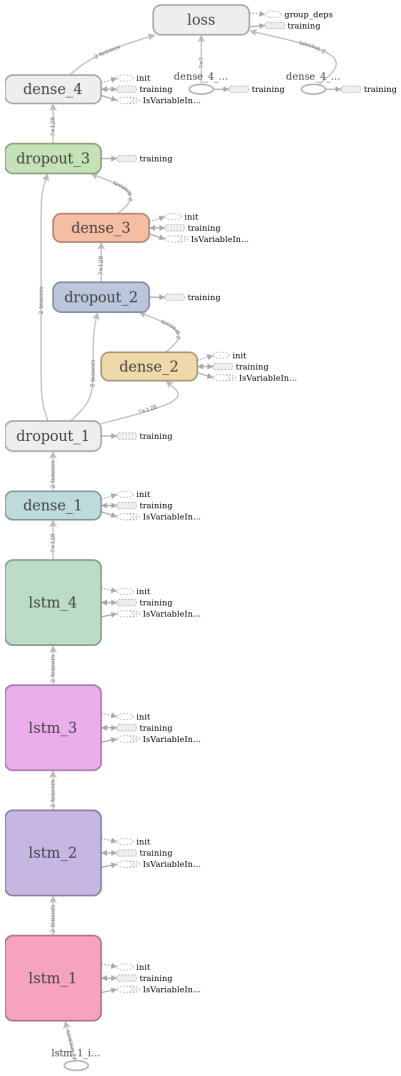


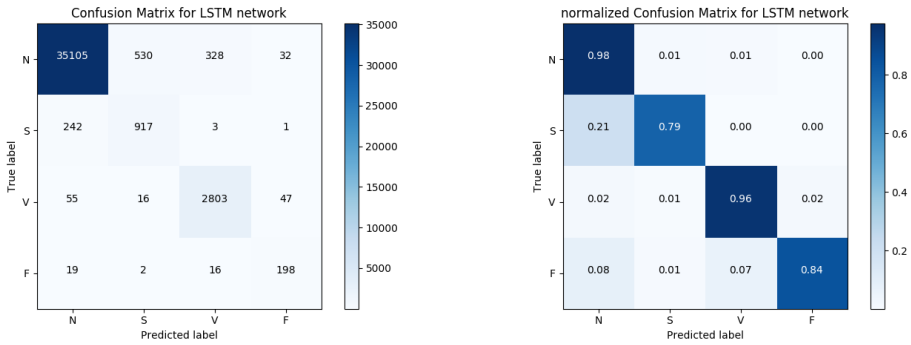
Figure 3.15. Recurrent Neural Network Structure

The best hyperparameters related to the Preprocessing stage:

- Window size = 200
- Number of Channels = Both
- augmentation factors = (S:4, F:8)

- reduction factors = (N:3)
- Timesteps = 2

Once all the parameters have been chosen we measure the performance on the test set. The final results are exposed in the confusion Matrix and the normalized version, shown in Figures 3.16a and 3.16b



(a) Confusion Matrix **(b)** Normalized Confusion Matrix

The values in Figure can be interpreted as explained in Table 3.2. The Complete Performance Measure are shown in Table 3.5

	Accuracy	**Recall**	**Precision**	**F1 Score**
Our RNN	0.968	0.891	0.805	0.844

Table 3.5. Performance Measures

3.8.1 Performance Comparison with State of The Art Algorithms

In Table 3.6 it is shown the comparison with the results of [30]. Note that their results are obtained on 11 signals out of the 48 of the MIT-BIH Database.

	Accuracy	**Recall**	**Precision**
Our RNN	0.968	0.891	0.805
MOE [30]	0.936	0.789	0.76

Table 3.6. Performance Comparison with [30]

In Table 3.7 it is shown the comparison with the results of [27]. Note that the results are not referred to the same test dataset, since they designed a different partition of the database.

	$Recall_S$	$Precision_S$	$Recall_V$	$Precision_V$
Our RNN	0.788	0.626	0.961	0.899
Morphology and Interval Features[27]	0.759	0.385	0.777	0.819

Table 3.7. Performance Comparison with [27]

Chapter 4

Conclusions and Future Work

This study concerns the development of an **end-to-end** system for Arrhythmia analysis along the ECG signal. The first stage, described in Chapter 2 detects the locations of heartbeats in a single ECG tracing. The Arrhythmia classification stage, takes as input the locations of the first stage, extracts the whole sequence of heartbeats and, according to their morphologies, performs a diagnosis of the occurring arrhythmic events. The different stages of computation are depicted in Figure 4.1

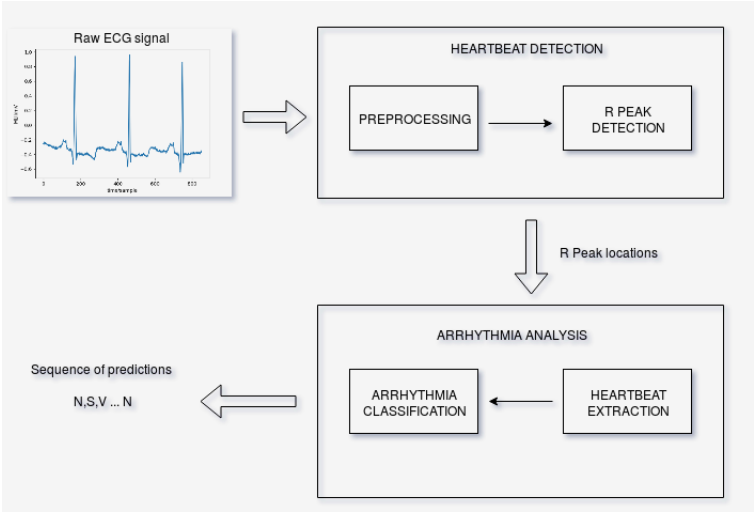


Figure 4.1. End-to-end system for Arrhythmia analysis

The Heartbeat detection stage achieved good performance measures *Precision=0.988 and Recall=0.923*, comparable to the State of the Art approaches.

For what concerns the Arrhythmia analysis, this study demonstrates that **a single classifier** trained on a reasonable amount of heartbeats from one or more patients, is able to make correct predictions on **unseen heartbeats** from the same source of data. The achieved perfomance measures are of *Recall=0.891 and Precision=0.805*.

A use case in which a single classifier is tuned on one patient's data would be more interesting for production purposes. However this solution requires an amount of data greater than the 30 minutes of ECG tracing publicly available. Anyhow, this study proves that this approach would be possible if a larger database for each patient was available.

Bibliography

[1] Orestis Akrivopoulos, Dimitrios Amaxilatis, Athanasios Antoniou and Ioannis Chatzigiannakis *Design and Evaluation of a Person-Centric Heart Monitoring System over Fog Computing Infrastructure* Proceedings of the First International Workshop on Human-centered Sensing, Networking, and Systems Pages 25-30, Delft, Netherlands - November 05 - 05, 2017

[2] Orestis Akrivopoulos, Ioannis Chatzigiannakis, Christos Tselios and Athanasios Antoniou *On the Deployment of Healthcare Applications over Fog Computing Infrastructure* 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), 4-8 July 2017

[3] Indu Saini, Dilbag Singh, Arun Khosla *QRS detection using K-Nearest Neighbor algorithm (KNN) and evaluation on standard ECG databases* Cairo University Journal of Advanced Research, Volume 4, Issue 4, July 2013, Pages 331-344

[4] Association for the Advancement of Medical Instrumentation *Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms* American National Standard 2013 ,ANSI/AAMI EC57:2012

[5] Gail Walraven *Basic Arrhythmias* Brady, 7th Edition

[6] Eduardo Jose da S. Luz, William Robson Schwartz, Guillermo Camara-Chavez, David Menotti *ECG-based heartbeat classification for arrhythmia detection: A survey* Computer Methods and Programs in Biomedicine, April 2016, pp 144-164, Elsevier

[7] Raul Alonso Alvarez, Arturo J. Mendez Penin, X. Anton Vila Sobrino *A comparison of three QRS detection algorithms over a public database* Procedia Technology Volume 9, 2013, Pages 1159-1165, CENTERIS 2013 - Conference on Enterprise Information Systems / ProjMAN 2013 - International Conference on Project Management / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

[8] O. Pahlm and L. Sornmo *Software QRS detection in ambulatory monitoring - a review* Medical and Biological Engineering and Computing, vol. 22, no. 4, pp. 289-297, 1984.

[9] Jiapu Pan and Willis J.Tompkins, *A Real-Time QRS Detection Algorithm* IEE Transactions on Biomedical Engineering, vol. BME-32, no. 3, March 1985 pp 230 - 236

[10] Ieong Chio In, Wan Feng, Vai Mang I, Mak Peng Un Faculty of Science and Technology, *QRS complex detector using Artificial Neural Network* University of Macau Macau, China

[11] Marko Sarlija, Fran Jurisic, Sinisa Popovic, *A Convolutional Neural Network Based Approach to QRS Detection* Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis, 18-20 Sept. 2017

[12] Rabiner L and Juang B, *An Introduction to Hidden Markov Models* Published in: IEEE ASSP Magazine, Volume: 3 , Issue: 1 , Jan 1986, pp 4 - 16

[13] S.Sotelo, W.Arenas and M.Altuve *QRS complex detection based on continuous density hidden Markov models using univariate observations* Journal of Physics: Conference Series, Volume 1002, conference 1

[14] Van G.V., Podmasteryev K.V., *Algorithm for detection the QRS complexes based on support vector machine* Journal of Physics: Conference Series, Volume 929, conference 1

[15] Christopher M. Bishop *Pattern Recognition and Machine Learning*, 2006 Springer Science Business Media, LLC

[16] Tom M. Mitchell *Machine Learning* McGraw-Hill Science/Engineering/Math; (March 1, 1997)

[17] Overfitting, `https://en.wikipedia.org/wiki/Overfitting`, consulted on 30/03/2018

[18] Cross-Validation `https://en.wikipedia.org/wiki/Cross-validation`, consulted on 13/02/2018

[19] Scikit-learn Documentation, `http://scikit-learn.org/stable/modules/grid_search.html`, consulted on 13/02/2018

[20] Karimifard S, Ahmadian A, Khoshnevisan M, Nambakhsh MS. *Morphological heart arrhythmia detection using Hermitian basis functions and KNN classifier* Proceedings of the 28th IEEE EMBS annual international conference, New York City, USA; August 30-September 3, 2006 p. 1367-70.

[21] Correlation coefficient `https://en.wikipedia.org/wiki/Correlation_coefficient`, consulted on 07/06/2018

[22] Christopher D. Manning, P. Raghavan, H. Schutze *An Introduction to Information Retrieval* Section 8.3 Online edition(c) 2009 Cambridge UP

[23] George G. Moody and Roger G. Mark *The Impact of the MIT-BIH arrhythmia Database*. IEEE Engineering in Medicine and Biology Magazine Volume: 20 , Issue: 3 , May-June 2001, pp 45 - 50

[24] Peak Utils Documentation,`https://media.readthedocs.org/pdf/peakutils/latest/peakutils.pdf`, consulted on 17/09/2018

[25] Chandrakar Kamath *A Novel Approach to Arrhythmia Classification Using RR Interval and Teager Energy* Journal of Engineering Science and Technology Vol. 7, No. 6 (2012) 744 - 755 School of Engineering, Taylor's University

[26] M. S. Thaler *The Only EKG Book You'll Ever Need* 3rd ed. Philadelphia, PA: Lippincott Williams and Wilkins, 1999.

[27] P. De Chazal, M. O'Dwyer, and R. B. Reilly *Automatic classification of heartbeats using ECG morphology and heartbeat interval features* IEEE Trans. Biomed. Eng., vol. 51, no. 7, pp. 1196-1206, Jul. 2004.

[28] K. Minami, H. Nakajima, and T. Toyoshima *Real-time discrimination of ventricular tachyarrhythmia with Fourier-transform neural network* IEEE Trans. Biomed. Eng., vol. 46, pp. 179-185, Feb. 1999.

[29] S. Barro, R. Ruiz, D. Cabello, and J. Mira *Algorithmic sequential decision-making in the frequency domain for life threatening ventricular arrhythmias and imitative artefacts: A diagnostic system* J. Biomed. Eng., vol. 11, pp. 320-328, 1989.

[30] Y. H. Hu, S. Palreddy, and W. J. Tompkins *A patient-adaptable ECG beat classifier using a mixture of experts approach* IEEE Trans. Biomed. Eng., vol. 44, pp. 891-900, Sept. 1997.

[31] Zachary C. Lipton, John Berkowitz, Charles Elkan *A Critical Review of Recurrent Neural Networks for Sequence Learning* June 5th, 2015, arXiv:1506.00019 [cs.LG]

[32] RNN tutorial, `http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/`, consulted on 07/10/2018

[33] M.G. Tsipouras, D.I. Fotiadis, D. Sideris, Arrhythmia *classification using the RR-interval duration signal* Comput. Cardiol., 2002, pp. 485-488.

[34] Miguel Angel Escalona-Moran, Miguel C. Soriano, Member, IEEE, Ingo Fischer, and Claudio R. Mirasso *Electrocardiogram Classification Using Reservoir Computing With Logistic Regression* IEEE journal of biomedical and health informatics, vol. 19, no. 3, may 2015

[35] Unbalanced Datasets, `https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/`, consulted on 22/06/2018

[36] Alex Graves, Abdel-rahman Mohamed and Geoffrey Hinton, *Speech Recognition with Deep Recurrent Neural Networks* 22 March 2015, arXiv:1303.5778

[37] S.Hochreiter and J.Schmidhuber *Long Short-Term Memory* Neural Computation, vol.9, no.8, pp.1735-1780, 1997

[38] LSTM tutorial, `https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21`, consulted on 27/09/2018

[39] Gael de Lannoy, Damien Francois, Jean Delbeke and Michel Verleysen *Weighted SVMs and Feature Relevance Assessment in Supervised Heart Beat Classification* BIOSTEC 2010, CCIS 127, pp. 212-223, 2011. Springer-Verlag Berlin Heidelberg 2011

[40] Osowski, Hoai, Markiewicz *Support vector machine-based expert system for reliable heartbeat recognition.* IEEE Transactions on Biomedical Engineering 51(4), 582-589 (2004)

[41] H. Jaeger, *The "echo state" approach to analysing and training recurrent neural networks-with an erratum note* German National Research Center for Information Technology, Bonn, Germany, GMD Tech. Rep., 148, 2001.

[42] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer,*Information processing using a single dynamical node as complex system* Nature Commun., vol. 2, p. 468, 2011

[43] Park, K., Cho, B., Lee, D., Song, S., Lee, J., Chee, Y., Kim, I., Kim, S. *Hierarchical support vector machine based heartbeat classification using higher order statistics and hermite basis function.* In Computers in Cardiology, pp. 229-232 (2008)

[44] Categorical Cross Entropy, `https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html`, consulted on 21/10/2018

[45] TensorBoard, `https://www.tensorflow.org/guide/summaries_and_tensorboard`, consulted on 21/10/2018

[46] Nair, Vinod and Hinton, Geoffrey E. *Rectified linear units improve restricted boltzmann machines* pp. 807-814 In Proc. ICML, volume 30, 2010.

[47] Dabal Pedamonti *Comparison of non-linear activation functions for deep neural networks on MNIST classification task* 8 April 2018,arXiv:1804.02763 [cs.LG]

[48] Clevert, Djork-Arne, Unterthiner, Thomas, and Hochreiter, Sepp. *Fast and accurate deep network learning by exponential linear units (ELUs)* arXiv preprint arXiv:1511.07289, 2015.