UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Atrial fibrillation

Detection of atrial rate and prediction during catheter ablation

Dissertation presented by
Simon MERTENS

for obtaining the Master's degree in
Mathematical Engineering

Supervisor(s)
Pierre-Antoine ABSIL, Sébastien MARCHANDISE

Reader(s)
Emilie RENARD, Christophe SCAVÉE

Academic year 2017-2018

First I would like to thank Pr. Pierre-Antoine Absil *and Emilie* Renard*, for their valuable follow-up and feedback all along this year and a half of work.*

I would also like to thank Dr. Sébastien Marchandise *and all the rhythmology department of the Cliniques Universitaires Saint-Luc, for their kindness, their helpfulness and all their precious advice they give me.*

Then I would like to thank Thierry, for his precious proofreading and his multiple useful suggestions.

I would like to thank my family and my friends next, for their continuous encouragements and for their unfailing faith.

Finally, I would like to thank my parents, for their unstinting support, their patience and their love without which I certainly could not do it.

Contents

Abstract

This thesis treats of the problematic of patients with atrial fibrillation who undergo a catheter ablation. It contains two main objectives. The first one is to implement an algorithm capable of computing the atrial rate of a patient during the procedure from the detection of the A peaks on the patient's atrial rhythm signal. The second one is to find a way to predict the catheter ablation outcome (i.e. the absence of recurrence of the pathology) based on the patient's characteristics.

An empirical method has been implemented to detect the atrial rate, since the traditional frequency-based methods were unsuccessful. This method is based on finding the peaks of a signal and keeping the ones which fulfil certain criteria. Those are determined by three parameters: the minimum height, the minimum peak separation and the minimum peak prominence. The first parameter is fixed, while the two other ones have to be selected. The user can either find these parameters manually (which gives a very good result) or use a suggested set of parameters (which can either provide a good result or lead to a very major error).

The learning has been conducted on 66 patients and 45 features. Among these features, the atrial rate and the maximum AA interval could not have been used, since those data where highly incomplete. Three algorithms have been considered: CART algorithm, Random Forest algorithm or K-Nearest Neighbours algorithm. The feature selection has been produced by two different methods: ReliefF and mutual information. Only the CART algorithm was able to output acceptable results with a BCR $= 0.73$, while the other algorithms were not satisfying at all, giving BCR near 0.5 (corresponding to total uncertainty).

Introduction

In the domain of rhythmology, atrial fibrillation remains one of the major causes of multiple significant health problem [1]. Although good progress in the management of patient with this pathology has been made, the number of patients with atrial fibrillation is predicted to rise steeply in the coming years. To meet the growing demand for effective care of patients with atrial fibrillation, new information is continually generated and published.

One of the main techniques to halt the atrial fibrillation is the catheter ablation. During this procedure, it is possible to measure the atrial rhythm via a dipole placed in the nearest position possible of the left atrium, in the coronary sinus. These signals form a new set of information that needs to be analysed. This thesis is an opportunity to find if there is any useful information that can be identified in these signals. This study has focused on two main characteristics, in particular the A peaks and the AA interval from the atrial rhythm. Not a lot of studies have looked into this direction and it is interesting to examine if something useful can come out from this perspective.

Two main objectives have therefore been determined. First of all, one wants to implement a method to compute the atrial rate on these extracted data. To do so, multiple methods have been considered, from frequency transform-based methods to empirical methods. These empirical methods are based on the detection of A peaks in the signal, peaks that express the electric activity in the left atrium.

The second objective is to observe if there is any way to predict the procedure's outcome. For a catheter ablation to be successful, the sinus rhythm must be recovered at the end of the procedure and there has to be no recurrence of the pathology in the months or years after the procedure. To predict this outcome, learning algorithms are considered. These algorithms have to take into account a full range of features, features that can be patient's physical characteristics, other pathologies, diverse medications intake, but also atrial rate-related informations detected at different key moments of the catheter ablation. These key moments are the disconnection of each pulmonary vein during the procedure. Two interesting values can therefore be computed at those times: the atrial rate and the maximum AA interval.

The aim of this paper is to report the progress made for the last year and a half. After having described the knowledge that is needed in order to fully understand the problematic of atrial fibrillation and catheter ablation, the paper focusses on the two targeted objectives. It describes the different envisaged methods to detect the atrial rate first. This covers also the preprocessing task as well as the detection of eventual parameters, before analysing the results. The paper goes then through the learning part of the thesis, which includes the detailed patient's features and the feature selection. The different considered models, as well as their possible parameters selection, are depicted, to end with the model selection.

Chapter 1

Atrial fibrillation and catheter ablation

1.1 A common abnormal heart rhythm: the atrial fibrillation

Atrial fibrillation (AF) is the most common arrhythmia affecting patients today. It is due to the apparition of disorganised electrical signals in the atria, causing them to fibrillate. This leads the heart to beat at a abnormally high rate. This disease can lead to major health problems, such as heart failure, syncope and stroke.

When a patient's heart is in AF, the patient might feel palpitations, tiredness, dizziness, shortness of breath or chest pain. However, AF may sometimes be asymptomatic. Therefore, an electrocardiogram (ECG) is needed in order to detect the disease (cf. Figure 1.1). For a healthy patient, the ECG should detect a regular rhythm called sinus rhythm. In case of AF, a abnormally rapid and irregular rhythm should be detected.

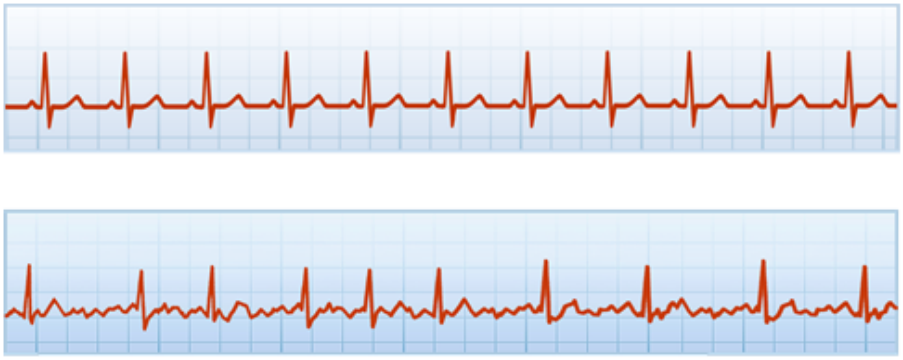

Figure 1.1: ECG of a healthy patient (above) and ECG of a patient with AF (below) [2].

1.1.1 Heart's electrical system

In normal case

The heart's electrical system controls the rate and rhythm of the heartbeat. With each heartbeat, an electrical signal spreads through the heart, from the top to the bottom, causing the contraction of the heart.

Each electrical signal is generated in a group of cells called the sinoatrial (SA) node, located in the right atrium. Normally, for a healthy adult at rest, this signal is sent 60 to 100 times a

minute. From the SA node, the signal is propagated to both atria, causing the contraction of these atria.

The electric signal reaches then a group of cell located between the atria and the ventricles called the atrioventricular (AV) node. Next the signal moves down to the ventricles, causing them to contract. These ventricles then relax and the whole process can start all over again from the SA node [3].

In atrial fibrillation

In the case of an atrial fibrillation, the heart's electric signals are not generated in the SA node. Small potentials appear in other parts of the atria or in the nearby four pulmonary veins (linked to the left atrium). Therefore, the electric signals do not propagate normally but get spread throughout the atria in a rapid, disorganized way, which causes the atria to fibrillate. The atrial rate may rise to 300 beats per minute.

All these faulty signals submerge the AV node with electric impulses, resulting in ventricles beginning to beat very fast too, but still not as fast as the atria since the AV node cannot send the signals to the ventricles as fast as they arrive. This results in atria and ventricles no longer beating in a coordinated way [3].

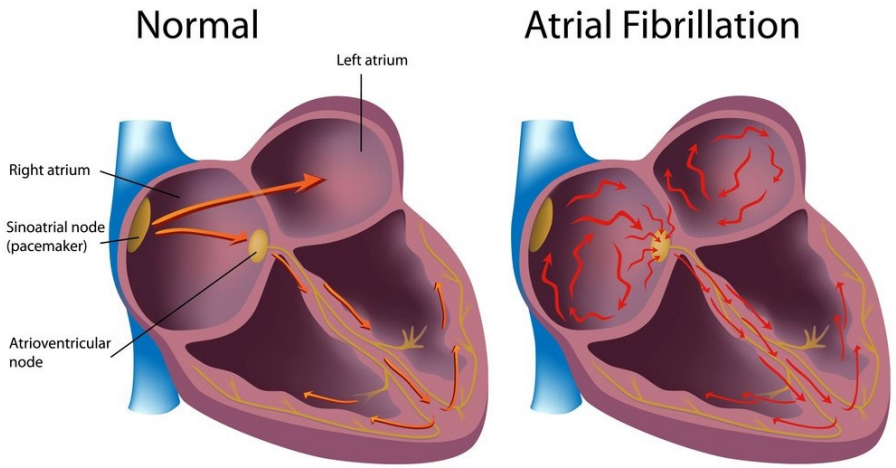

Figure 1.2: Heart's electrical system in normal case and in atrial fibrillation [4].

1.1.2 Types of atrial fibrillation

There are different types of atrial fibrillation depending on how long the detected episode is [5]:

- Paroxysmal AF: recurrent episodes that stop spontaneously within a week;
- Persistent AF: recurrent episodes that last more than a week;
- Permanent AF: long-time episodes.

Within the framework of the thesis, only permanent AF will be considered, since only these can lead to catheter ablation.

1.1.3 Management of an atrial fibrillation

The first way to manage atrial fibrillation is via medication. Anticoagulants to avoid the risk of stroke, antiarrhythmic agents to suppress abnormal rhythms of the heart or beta blockers to reduce the heart rate are the three main ways to control AF with drugs.

An other approach consists of the cardioversion in order to convert an arrhythmia to an normal rhythm by administration of a DC electrical shock.

If those previous techniques have not worked, then catheter ablation can be attempted. This will avoid years of drug therapy.

1.2 An invasive method to treat atrial fibrillation: the catheter ablation

Catheter ablation of an AF is an invasive method in order to eliminate all abnormal electrical pathways that are contributing to an AF. To do so, small scars are made in the involved heart tissue. Those scars are made with an energy-emitting probe situated at the tip of a catheter. This probe, that deliver the ablating energy, can use either radiofrequency (heat generated by medium frequency AC), or cryothermic energy (extreme cold) to destroy tissues.

This catheter reaches its working zone via the femoral vein, the inferior cave vein, the right atria and the left atria.

The first objective of the procedure is to electrically isolate the four pulmonary veins since a lot of electric potentials comes from the ostium of these veins. To do so, on each vein, scars are made elliptically all around the ostium. At a certain point, disconnection of the vein should be observed by the heart surgeon. An other approach often used today, is to encircle both pulmonary veins ostia on one side with one single wider elliptical line. Still the surgeon must ensure that each pulmonary vein is disconnected.

It is sometimes not enough to ensure the return of a sinus rhythm at the end of the procedure. Therefore some other zones of the atrium can also be isolated as a result of the ablation of complex fractionated atrial electrograms (CFAE), site where an unusual electrical pattern is detected.

Usually, at the end of the procedure, a cardioversion is carried out in order to hopefully restore a sinus rhythm.

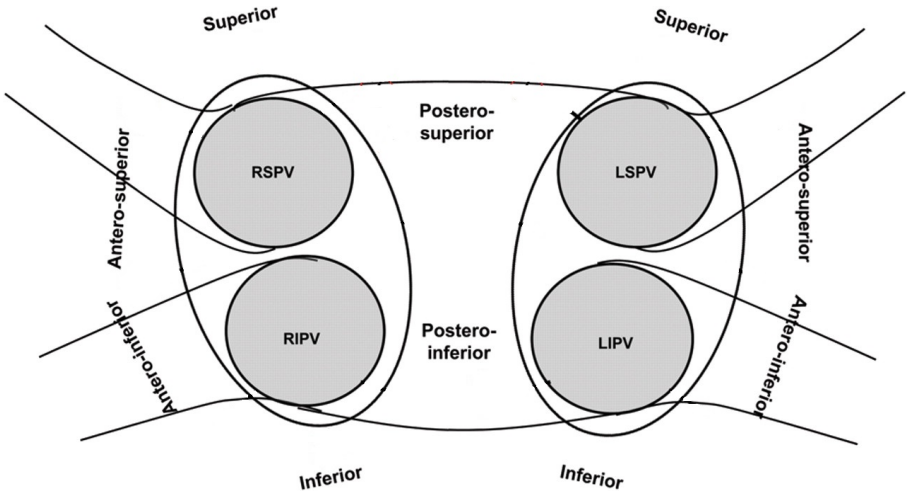

Figure 1.3: Ablation procedure in left atrium [6].

Chapter 2

Detection of atrial rate

The first step of this thesis is to detect the atrial rate of a patient. To do so, signals throughout catheter ablation procedures have been considered. During the procedure, the atrial rhythm is measured with a dipole placed in the coronary sinus. All these signals have a sampling frequency of 1000 Hz. A classical atrial rhythm is displayed on Figure 2.1. As expected, this is very irregular and fast rhythm.

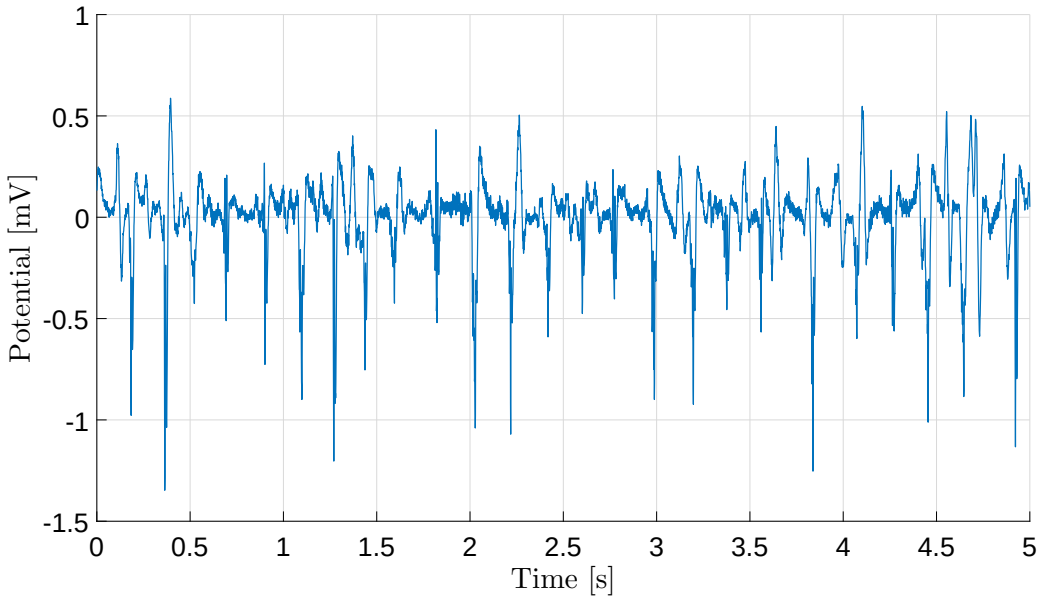

Figure 2.1: Usual atrial rhythm on a 5 seconds interval during AF.

To compute the atrial rate, two different ways have been considered. A first approach is to work with those signals in the frequency domain, through either Fourier transform or wavelet transform, in order to perform an harmonic analysis. An other way to proceed, is to detect AA interval via empirical methods. An AA interval is the interval between two A peaks. An A peak (cf. Figure 2.2) is the repercussion on the signal of an important electrical activity detected by the dipole. Therefore the objective is to detect all the A peaks of a signal, and then to compute all the AA interval, to finally obtain the atrial rate with a moving average.

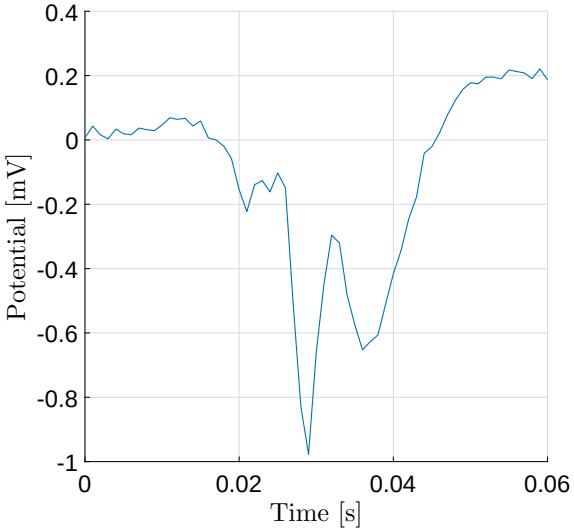

Figure 2.2: An A peak.

2.1 Signals exportation

In the context of this thesis, 72 patients who suffered from permanent AF and who underwent a radiofrequency catheter ablation in 2014 and 2015 have been considered. The data of these patients have been extracted from the database of the Cliniques Universitaires Saint-Luc. These files are `.txt` files which have been imported subsequently in MATLAB. Simultaneously, the disconnection time of each pulmonary vein has been gathered from the medical records related tothe catheter ablation (when clearly referenced) and compiled in a `.xlsx` file in order to import it in MATLAB too.

2.2 Frequency transform-based methods

2.2.1 Preprocessing

Since the signal is very noisy and with a very high sampling frequency, preprocessing is imperative. Different techniques have been tested in order to remove all the small variations on the signal in a smoothing procedure. Among all of them, the Savitzky-Golay filter provided the best results.

The Savitzky-Golay filter derives directly from a particular formulation of the data smoothing problem in the time domain [7]. This works through the convolution of the signal using a moving average. A simple moving average consists in replacing each signal value f_i by a linear combination g_i of itself and some number of its nearest neighbours:

$$g_i = \sum_{j=-n}^{n} C_i f_{i+j}$$

with $C_i = \frac{1}{2n+1}$. This is called moving window averaging., since the considered moving window is a constant. The idea of Savitzky-Golay filtering is to approximate the underlying function within the moving window by a polynomial of higher order (cf. Figure 2.3. Therefore, for each point, this polynomial is fitted to all the $2n + 1$ points in the moving window via the least-squares method. For an equally spaced signal values, an analytical solution to the least-squares equations can be found, forming the set of the convolution coefficients C_i.

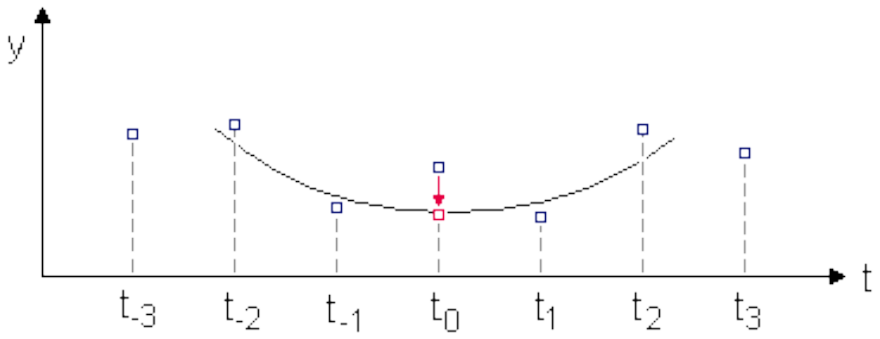

Figure 2.3: Illustration of Savitzky-Golay filtering [8].

All the analysed signals in this thesis have been filtered using the Savitzky-Golay algorithm. An example of this filtering is displayed on Figure 2.4. One can observe that this filter can slightly decrease the amplitude of the strong variations. This has to be taken into account when analysing the signals.

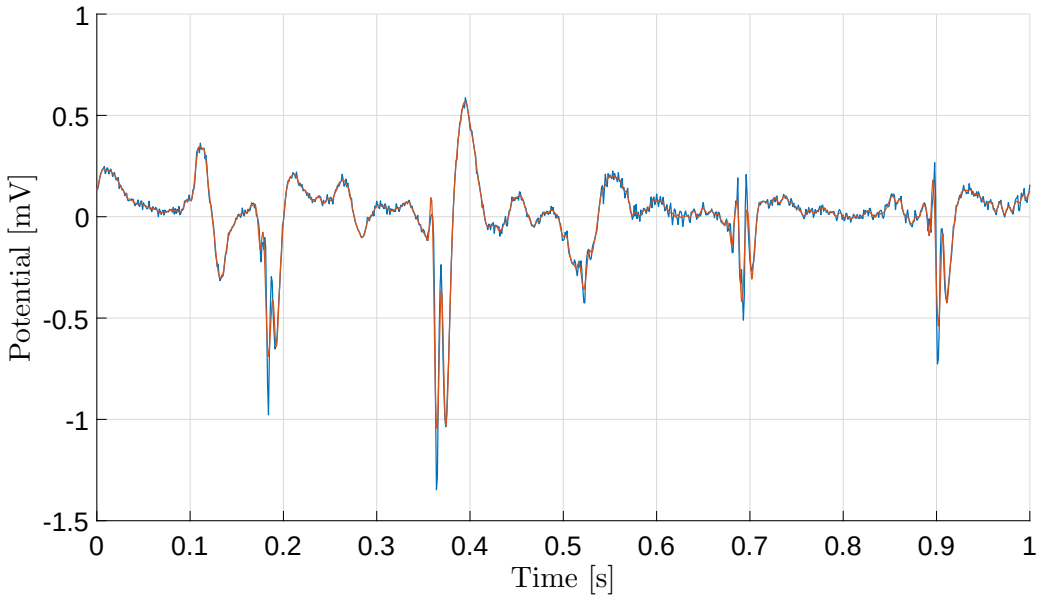

Figure 2.4: Application of the Savitzky-Golay filter on an signal (original signal marked in blue, filtered signal in orange).

2.2.2 Fourier transform-based method

A fast Fourier transform algorithm is applied on the signal displayed on Figure 2.1 and that underwent a preprocessing. By applying this algorithm, the objective is to find if there is any dominant frequency which could correspond to the atrial rhythm. As the results depicted on Figure 2.5 suggest, the signal does not seem to have a dominant frequency. It remains a very chaotic signal: there is still noise despite preprocessing and the fact that the signal is, by default, highly irregular, leads to the absence of usable results with this method. However, one can observe that Figure 2.5 shows peaks around the frequency of 5 Hz, which corresponds to an

atrial rate of 300 beats per minute, the usual AF rate.

This technique could actually work at some point, when the procedure succeeded and a sinus rhythm has been reached.

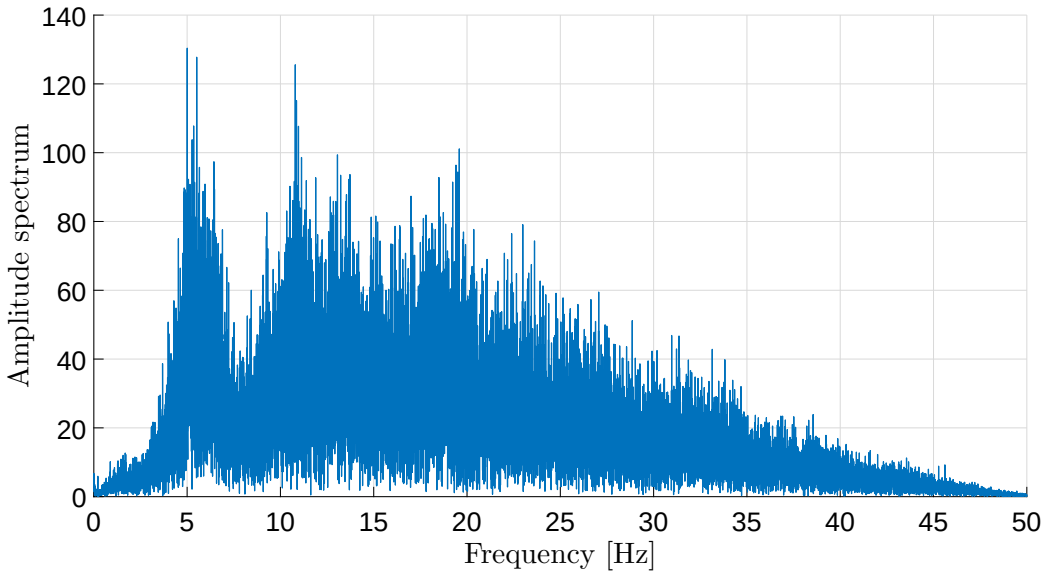

Figure 2.5: Amplitude spectrum of FFT.

2.2.3 Wavelet transform-based method

The difficulty in this method is to find a wavelet that could correspond to the signal. This technique did not bring any useful results and so it was not useful to explore it any further.

2.3 Empirical method

Since the frequency transform-based methods do not give an positive answer to the problem, an empirical method has to be implemented to detect the atrial rate. Again, preprocessing is essential and the same filtering as before is applied on all the data.

2.3.1 Description of the algorithm

Rather than detecting directly the main frequencies of the signal, identifying the A peaks of the signal is now more interesting. In order to do so, the local optima of the signal are detected. For each of them, three features are considered, so that it only remains the A peaks at the end [9].

Minimum peak height Only the optima that have an amplitude greater than the minimum peak height are selected.

Minimum peak separation Only the tallest peaks in the signal are selected and all the other peaks within this minimum peak separation are not considered. The algorithm repeats the procedure for the tallest remaining peak and iterates until it runs out of peaks to consider.

Minimum peak prominence Only the peaks that have a relative importance higher than the minimum peak prominence are selected. The prominence of a peak measures how much the peak stands out due to its intrinsic height and its location relative to other peaks. A low isolated peak can be more prominent than one that is higher but is an otherwise unremarkable member of a tall range. To measure the prominence of a peak [9]:

1. Place a marker on the peak.
2. Extend a horizontal line from the peak to the left and the right until the line does one of the following:
   - Crosses the signal because there is a higher peak
   - Reaches the left or right end of the signal
3. Find the minimum of the signal in each of the two intervals defined in Step 2. This point is either a valley or one of the signal endpoints.
4. The higher of the two interval minima specifies the reference level. The height of the peak above this level is its prominence.

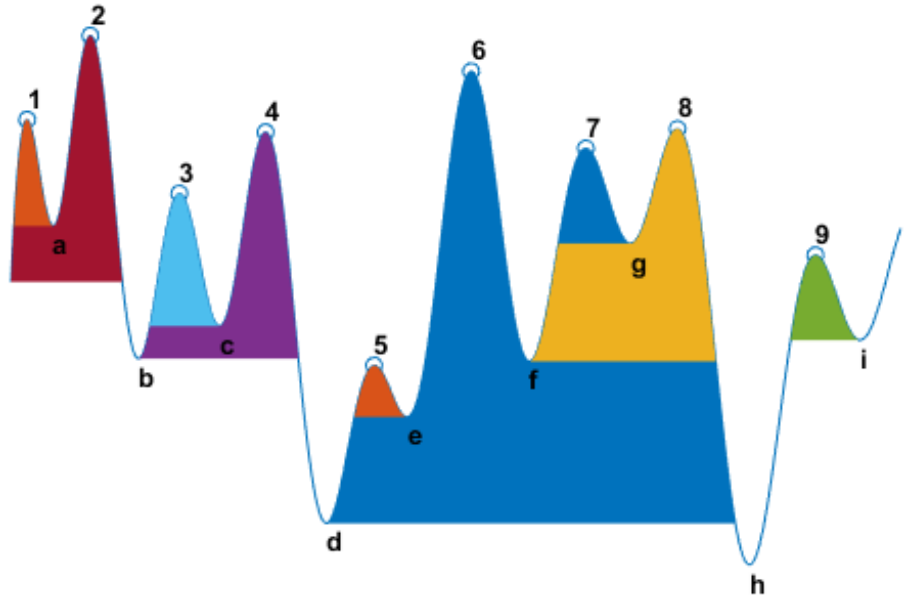

Figure 2.6: Examples of prominence for a given signal [9].

Knowing that, the algorithm reverses all the local optima that match with these three features. The objective is now to find the value of these parameters, so that the returned peaks really correspond to A peaks, and that no A peaks are forgotten.

From the A peaks locations, two values are computed for analysis: the atrial rate (AR) and the maximum AA interval, which is the biggest gap between two consecutive A peaks.

2.3.2 Parameters selection

Among those three parameters, one of them is set: the minimum peak height. This parameter is set on 0.1 mV. The two other parameters have to be determined: the minimum peak separation (PS) and the minimum peak prominence (PP).

In order to test the algorithm and to find suitable parameters, tests are conducted on all the 30 seconds-interval following the disconnection of each pulmonary veins (PV) for the patients whose data are complete. In the frame of this project, the disconnection time of each of the four PV is available for 10 patients (the other data were missing, wrong or not complete for the other patients).

For these 10 patients and for each disconnection of the PV, the best set of parameters is found empirically. Therefore the atrial rate and the maximum AA interval can be computed. All the sets of parameters and the corresponding results are gathered in Table 2.1.

From all the sets of parameters, means can be computed. The mean minimum PS parameters is 125.25 ms and the mean minimum PP is 0.12075 mV. These two mean parameters form a set of parameter that is now used by the algorithm. The objective now is to determine if using this set of parameter leads to a big error. So these parameters are used to run this algorithm with the 40 previously considered signals and again, the AR and the maximum AA interval are computed, as well as the percent error (just for an easy comprehension). All the computations are integrated in Table 2.2.

Those results allows for the computation of the root-mean-square error (RMSE) for the AR and for the maximum AA interval. The root-mean square error is computed as follows for predicted values $\hat{h}_i$ and expected values h_i:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Therefore the RMSE for the AR (RMSE_{AR}) is equal to 23.271 BPM, and the one for the maximum AA interval (RMSE_{AA}) is equal to 164.506 ms. These big errors are due to big outliers that exist (cf. Figure 2.7). It is known that the RMSE is very sensitive to outliers and so the errors are very consequent. However the majority of experiments lead to small errors (and even null errors). Therefore it is interesting to see if the big errors appear only when the two mean computed parameters are too far away from the initial ones.

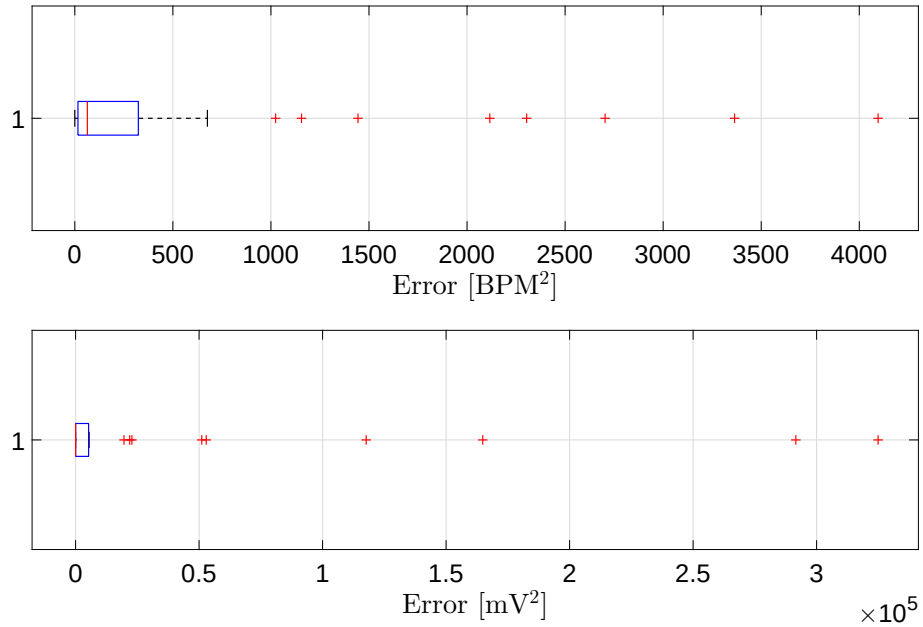

Figure 2.7: Boxplot of all the squared errors (MSE) on the AR (top) and on the maximum AA interval (down).

Patient	PV	Min PS [ms]	Min PP [mV]	AR [BPM]	Max AA interval [ms]
#1	1	110	0.1	286	523
	2	100	0.11	270	655
	3	100	0.105	256	659
	4	110	0.11	270	706
#2	1	110	0.11	336	347
	2	115	0.115	322	468
	3	125	0.11	315	510
	4	125	0.11	280	581
#3	1	120	0.15	305	350
	2	130	0.14	303	366
	3	130	0.15	290	482
	4	140	0.15	293	410
#4	1	130	0.11	318	294
	2	130	0.11	312	296
	3	140	0.11	300	341
	4	140	0.11	300	303
#5	1	130	0.105	338	357
	2	130	0.1	336	349
	3	120	0.1	260	622
	4	105	0.105	202	1195
#6	1	140	0.12	302	293
	2	140	0.11	324	332
	3	130	0.12	298	371
	4	130	0.1	290	360
#7	1	140	0.12	294	476
	2	130	0.13	290	440
	3	120	0.13	292	459
	4	120	0.13	281	485
#8	1	120	0.13	334	338
	2	120	0.13	334	315
	3	115	0.12	310	284
	4	110	0.12	306	291
#9	1	115	0.14	326	336
	2	115	0.13	334	325
	3	130	0.14	312	348
	4	130	0.15	296	411
#10	1	120	0.13	280	573
	2	120	0.13	314	552
	3	115	0.12	302	474
	4	115	0.12	294	488

Table 2.1: Atrial rate (AR) and maximum AA interval for the 10 considered patients, as well as the corresponding set of parameters.

To see how the error is related to the set of initial parameters, scatter plots are produced (cf. Figure 2.8). Those scatter plots show how big the error is if the set of initial parameters is too far away from the set of mean parameters. The bigger the red points, the bigger the error.

Patient	PV	AR [BPM]	Max AA interval [ms]
#1	1	228 (−20.280%)	866 (+65.583%)
	2	222 (−17.778%)	1195 (+82.442%)
	3	218 (−14.844%)	889 (+34.901%)
	4	236 (−12.593%)	932 (+32.011%)
#2	1	326 (−2.732%)	347 (+0%)
	2	315 (−2.174%)	468 (+0%)
	3	314 (−0.317%)	510 (+0%)
	4	280 (+0%)	581 (+0%)
#3	1	322 (+5.574%)	331 (−5.429%)
	2	316 (+4.290%)	366 (+0%)
	3	308 (+6.207%)	449 (−6.846%)
	4	311 (+6.143%)	371 (−9.512%)
#4	1	326 (+2.516%)	288 (−9.434%)
	2	314 (+0.641%)	296 (+0%)
	3	302 (+0.667%)	341 (+0%)
	4	302 (+0.667%)	303 (+0%)
#5	1	330 (−2.367%)	376 (+5.322%)
	2	318 (−5.357%)	489 (+40.115%)
	3	228 (+12.308%)	1028 (+65.273%)
	4	156 (−22.772%)	1765 (+47.699%)
#6	1	366 (+21.192%)	292 (−0.341%)
	2	376 (+16.049%)	261 (−21.386%)
	3	292 (−2.013%)	371 (+0%)
	4	285 (−1.724%)	511 (+0%)
#7	1	312 (+6.122%)	440 (−7.563%)
	2	301 (+2.381%)	440 (+0%)
	3	298 (+2.055%)	459 (+0%)
	4	287 (+2.135%)	485 (+0%)
#8	1	330 (−1.120%)	332 (−1.119%)
	2	334 (+0%)	315 (+0%)
	3	306 (−1.290%)	286 (+0%)
	4	299 (+0%)	291 (−2.288%)
#9	1	320 (+1.840%)	336 (+0%)
	2	320 (−4.192%)	325 (+0%)
	3	322 (+3.205%)	348 (+0%)
	4	322 (+8.784%)	337 (−30.943%)
#10	1	286 (+2.143%)	573 (+0%)
	2	318 (+1.274%)	404 (−26.812%)
	3	298 (−1.325%)	474 (+0%)
	4	294 (+0%)	488 (+0%)

Table 2.2: Atrial rate (AR) and maximum AA interval for the 10 considered patients, with the set of mean parameters, as well as the error.

These plots can lead to multiple observations. First the minimum PS has a greater impact on the peak detection. Having a minimum PS that is not close enough to the optimal one quickly leads to a significant error for both the AR and the maximum AA interval. Yet minimum PP

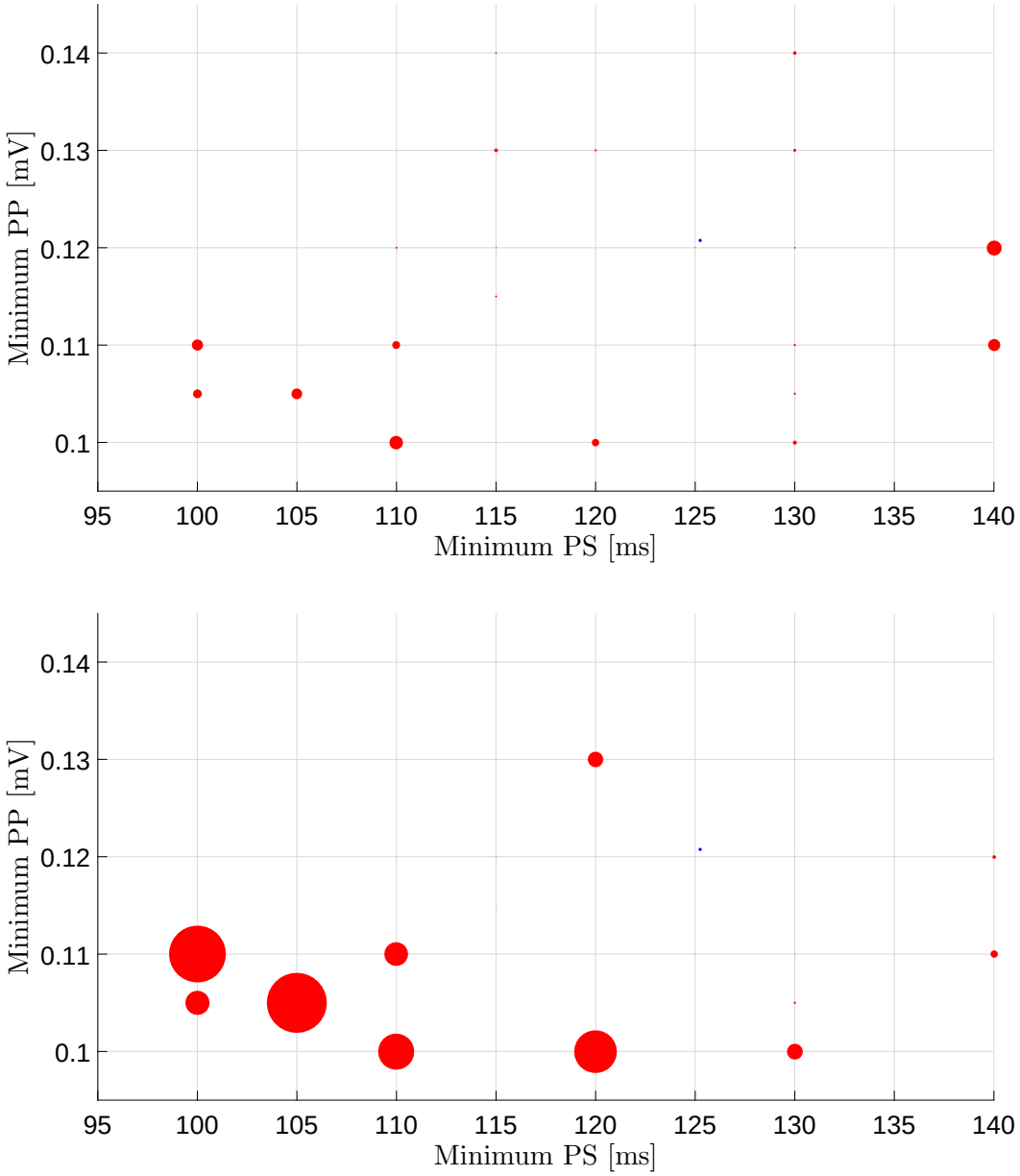

Figure 2.8: Scatter plot of all the squared errors (MSE) on the AR (top) and on the maximum AA interval (down). The blue point represents the set of mean parameters (NB: for clarity, the diameters of the red points on the bottom plot are reduced of a factor 5).

cannot be overlooked since a wrong choice leads to an error, especially for the maximum AA interval, in a lesser degree though. An other observation is that choosing parameters that are greater than the optimal ones leads to bigger errors than choosing ones that are lower to the optimal parameters.

Choosing the set of mean parameters to put in the algorithm may thus work in many cases. However, when it does not, it almost directly leads to major errors. Therefore, this algorithm cannot be blindly trusted and must always be subject to verification. In those cases, a set of optimal parameters can be found empirically.

2.4 Analysis of results

It has been explained that with optimal parameters, the algorithm works very well and allows to detect the A peaks of a signal. The peaks can be detected either positively or negatively, since the signal is not taken under its initial form but in its absolute value, introduced into the algorithm. The reason therefore, is that the algorithm detects local maxima. Since the electric potential detected by the dipole during the procedure may be either positive or negative, the algorithm has to detect the peak that corresponds to only one electric activity at a time.

An example of detection by the algorithm is displayed on Figure 2.9.

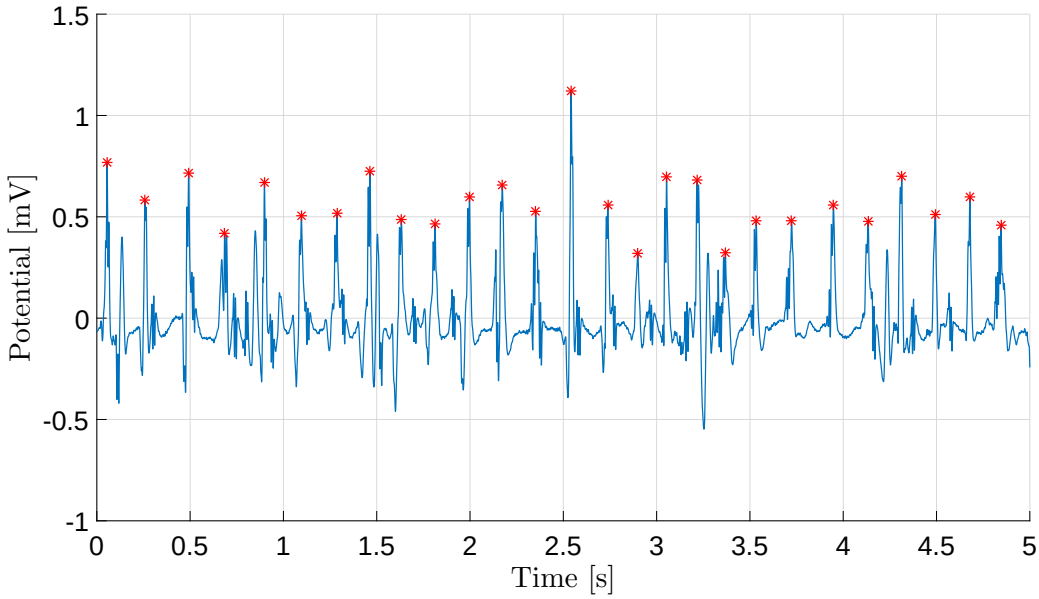

Figure 2.9: Application of the algorithm on an signal with its optimal parameters; A peaks are spotted by the red markers.

Now that the algorithm is applied and the AR and the maximum AA intervals have been found after the disconnection of PV for the considered 10 patients, it is possible to observe evolution during the procedure. To do so, all the results are displayed in the form of boxplots, one after the disconnection of each PV, and for both AR and maximum AA intervals. These boxplots appear on Figure 2.10.

The AR seems to decrease slightly during the procedure and all the values from the different patients tend to close up. Regarding the maximum AA interval, these values seems to slightly growing up, which is actually logical if the AR decreases. However, too high values for the maximum AA interval (i.e. outliers) show that the signal does not reorganise as much as the procedure should permit. A comeback to a more normal rhythm can not be noticed at the stage of the procedure and therefore there may probably be other electric potential to isolate inside the left atria to isolate (CFAE).

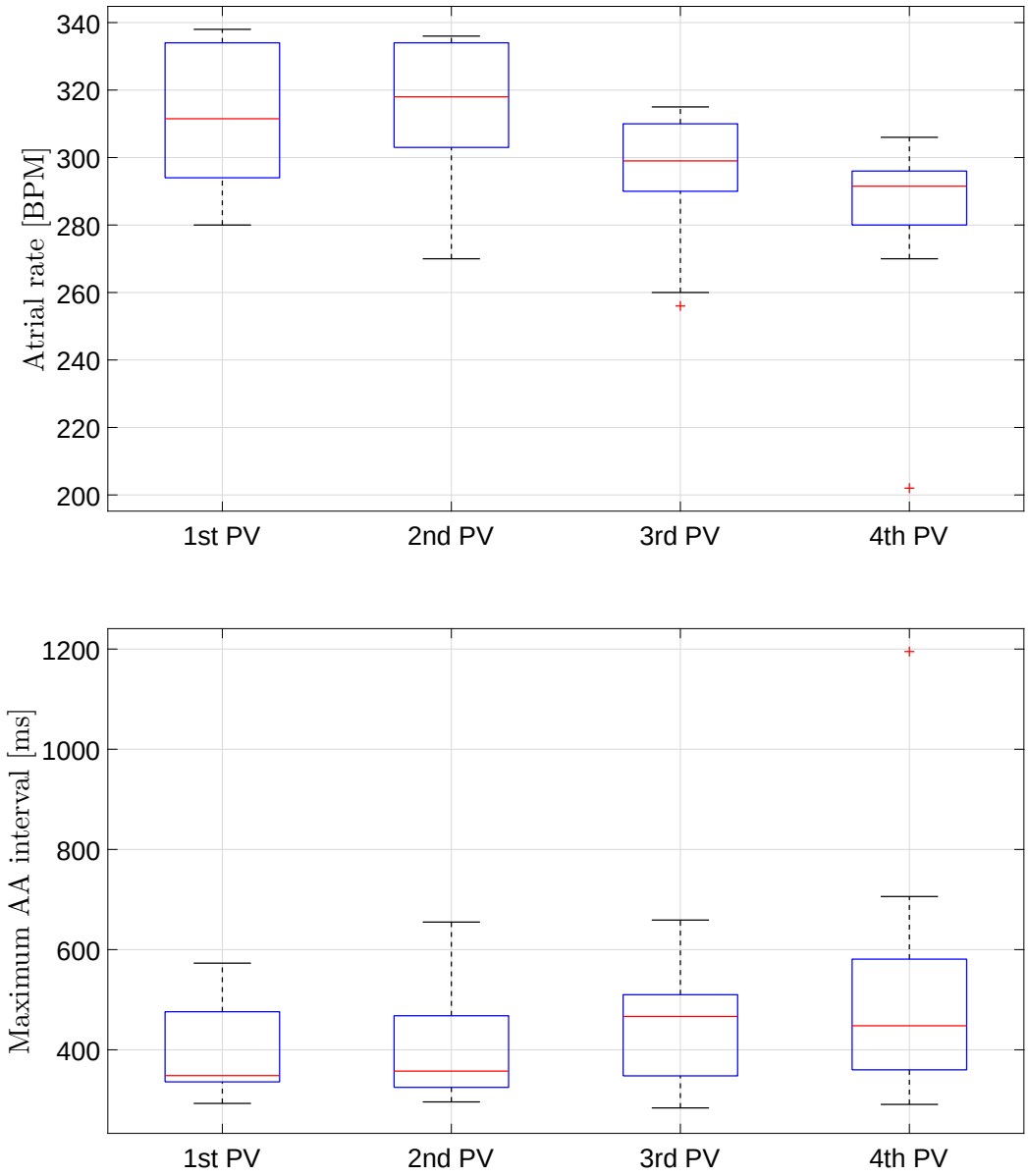

Figure 2.10: Boxplots of the computed AR (top) and maximum AA interval (down) after the disconnection of each pulmonary veins.

Chapter 3

Prediction of procedure success

The objective is to find any relation between a patient's characteristics and the procedure's success. Although catheter ablation is not really a high-risk procedure, it would be interesting to have sufficient information prior to the procedure if this is going to be beneficial to the patient's health. The success of the procedure is related to the absence of recurrence of this pathology during the months and years after the catheter ablation.

Initially the objective was to predict the procedure's outcome using the atrial rates and the maximum AA intervals previously detected, as well as the patient's characteristics. However, since too few patients had complete data, it would have no sense to implement a learning with those kind of data. Dealing with the missing data would have only be an option if many more of those information were available. Therefore only the patient's features are used to learn the procedure's success.

3.1 Patient's features considered

The study is based on the features of 66 patients. For each patient, 45 features have been provided by the Cliniques Universitaires Saint-Luc. Those are very diverse and cover physical characteristics such as gender, age, height, weight, Body Mass Index (BMI), the existence of different diseases, pathologies or addictions (diabetes, hypertension, dyslipidemia, strokes, respiratory disease, HAS-BLED score[1], smoking...), previous or actual intake of various medications (anticoagulants, beta blockers, antiarrhythmic agents, antiaggregant...), or even family medical antecedents, the size of the left atrium or different patient's heart characteristics. The presence of recurrence of AF is known for each patient (at the moment when the data were gathered, i.e. July 2017).

3.2 Models implementation

3.2.1 Feature selection

The feature selection for this problem is difficult, as the data may be continuous, discrete or categorical (most of them are binary). Most feature selection algorithms can deal without difficulty with continuous and discrete variables but not with categorical ones. Therefore, other

[1]HAS-BLED is a scoring system to evaluate bleeding risk in patient with AF. The term HAS-BLED is a mnemonic device for **H**ypertension, **A**bnormal renal or liver function, **S**troke, **B**leeding (bleeding history or predisposition), **L**abile INR (determination of the clotting tendency of blood), **E**lderly (greater than 65 years old), **D**rugs or alcohol [10].

algorithms (or updated ones) are needed. In the context of this thesis, two feature selection algorithms are considered: ReliefF and mutual information.

ReliefF The ReliefF algorithm is an extension of the Relief algorithm. The key idea of the Relief algorithm is to estimate the quality of attributes according to how well their values distinguish between instances that are near to each other [11]. To do so, Relief selects first a instance R_i at random. Then the algorithm searches two other instances which are its two nearest neighbours, one from the same class (the nearest hit H) and one from the different class (the nearest miss M). From these two neighbours, the quality estimation $W[A]$ (initialize to 0) is updated for all attributes A the following way:

$$W[A] := W[A] - \frac{1}{m}\text{diff}(A, R_i, H) + \frac{1}{m}\text{diff}(A, R_i, M),$$

where the function $\text{diff}(A, I_1, I_2)$ computes the difference between the values of the attributes A for two instances I_1 and I_2. What it is interesting with this function is that it can compute the difference between numerical attributes, but also the one between categorical values. Indeed, the function is defined for categorical values:

$$\text{diff}(A, I_1, I_2) = \begin{cases} 0, & \text{if value}(A, I_1) = \text{value}(A, I_2) \\ 1, & \text{otherwise,} \end{cases}$$

as well as for numerical values:

$$\text{diff}(A, I_1, I_2) = \frac{|\text{value}(A, I_1) - \text{value}(A, I_2)|}{\max(A) - \min(A)}.$$

This whole process is repeated m times, this parameter being chosen by the user.

The ReliefF algorithm extends this algorithm in a way that not only the nearest neighbours are considered but k of the nearest neighbours. The quality estimation $W[A]$ is updated, considering the difference with each one of the neighbours (differences that can be weighted according to the user's choice).

Knowing how this algorithm works, and after having imported it via the R package named `CORElearn`, it is used to select the 10 features having the highest quality estimation. Those are:

1. Elderly (age > 65)
2. Congestive heart failure
3. Typical atrial flutter
4. Drugs or alcohol use
5. Hypertension
6. Abnormal renal or lever function
7. Left ventricle end-sy stolic diameter
8. Anticoagulant intake
9. ACE inhibitor intake
10. Left ventricular ejection fraction

Mutual information Mutual information between two variables x and y is the measure of how the uncertainty on y is reduced when x is known. To measure the uncertainty of a variable, the concept of entropy is used. The entropy of a variable is defined as $H(Y) = -E[\log(P[Y])]$, which is equal to $-\sum_{y\in\Sigma}\log(P[y])P[y]$ when Y is discrete, and $-\int\log(P[y])\mathrm{d}y$ if Y is continuous. Since the mutual information measures how the uncertainty of y is reduced when x is known, it is formulated as the difference between the entropy of y and the entropy of y when x is known, or $I(y;x) = H(y) - H(y|x) = H(x) - H(x|y)$.

Mutual information permits to identify nonlinear relationships between variables. Once again, the difficulty here is to compute the conditional entropy with a continuous and a categorical variable. However, methods to do so can be found in the R package `mpmi`. These methods enable to select the most relevant features (i.e. features that maximize $I(x_i;j)$ where j is the recurrence of AF) in this case:

1. Anticoagulant intake
2. Height
3. Drugs or alcohol use
4. Typical atrial flutter
5. Left ventricle end-diastolic diameter
6. Bleeding
7. Labile INR
8. Age
9. Left ventricle end-systolic diameter
10. Size of the left atria

All the models that will be implemented, will be tested without feature selection, and with each of these two feature selection algorithm. Note also that, before running all the models, feature scaling is also applied on numerical data, with a rescaling method, to obtain $x' = \frac{x-\min(x)}{\max(x)-\min(x)}$, where x is the original feature vector and x' the normalized one. In this way, all the numerical data are scaled to the range $[0, 1]$, which makes much easier the computation of the distance between numerical data and binary ones.

3.2.2 Description of the models

For this classification problem, with such various data (continuous, discrete and categorical), three kind of algorithms have been considered. Two of them are based on the decision trees training (CART algorithm and Random Forests), and the other one is the K-Nearest Neighbours algorithm. For each algorithm, one parameter is needed, and it has to be determined in order to bring the model to an optimal accuracy.

CART algorithm The first considered learning algorithm is the CART algorithm. This is a binary recursive partitioning procedure that processes continuous and categorical attributes as targets and predictors [12]. From the root node, the data are split into two leaves, themselves being split into leaves. There is no stopping rule: the tree is grown to its maximal size, when no more splits are possible. Then the obtained tree is pruned back to the root, split by split, with

the method of cost-complexity (*cp*, the one parameter of this method) pruning. Splits which are pruned are the ones that contribute least to the overall performance. The cost-complexity parameter *cp* is such, that any split that does not decrease the overall lack of fit by a factor *cp* is not attempted. This method is highly sensitive to overfitting.

Random Forest algorithm The Random Forest algorithm (RF) is based on decision tree algorithms, such as CART. Then not only one but many decision tree models are build [13]. Each model is based on a subset of the training data, that is generated randomly by selecting a certain number of training vectors (about two thirds) with replacement. For each model, the remaining training data are used to estimate error and the variable importance. At the end, the class assignment is made by the number of votes from all the trees. In this case, the key parameter here is number of trees N_{tree} build by the algorithm.

K-Nearest Neighbours algorithm The principle of the K-Nearest Neighbours algorithm (KNN) is very simple. For a test vector v, the algorithm is looking in a training set T for the K nearest neighbours, according to a defined distance metric. To assign a class to v, the algorithm takes the class of the majority of the neighbours. The only parameter for this method is the number of considered neighbour K. A low K will lead to overfitting while a high K will on the contrary lead to underfitting. This method is very sensitive to the curse of dimensionality: KNN will be misled by high-dimensional input space.

3.2.3 Model selection

In order to evaluate the performance of each model, a test performance metric is needed. In the framework of a binary classification problem, the balanced classification rate (BCR) is a good choice. The BCR is computed from the confusion matrix, that can be seen in Table 3.1, and is defined as follows:

$$\mathrm{BCR} = \frac{1}{2}\left(\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} + \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\right)$$

The first term of the BCR is the sensitivity, the second one is the specificity.

		Actual class	
		+	-
Predicted class	+	True positive (TP)	False positive (FP)
	-	False negative (FN)	True negative (TN)

Table 3.1: Representation of a confusion matrix.

In all the models, cross-validation is operated in order to obtain a mean BCR over 50 tests. For each test, a training set is randomly generated and contains 45 samples, that is about 70% of the data, the remaining 30% (21 samples) forms the test set.

The first algorithm to be tested is the CART algorithm. For the cost-complexity parameter *cp*, different values from 5.10^{-10} to 5.10^{-1} have been tested. The BCR obtained after cross-validation and for each of the features set selected is displayed on Figure 3.1.

In this case, feature selection does not bring any improvement. The loss of information coming from this feature selection, feeds through the BCR. The CART algorithm can actually handle all the features at once, the pruning of the tree takes care of the unnecessary or redundant features. It also appears that the BCR is maximum, for two of the three sets of features, for $cp = 0.005$. However, a maximum BCR equal to about 0.72, is not the best possible outcome, surely for a medical study.

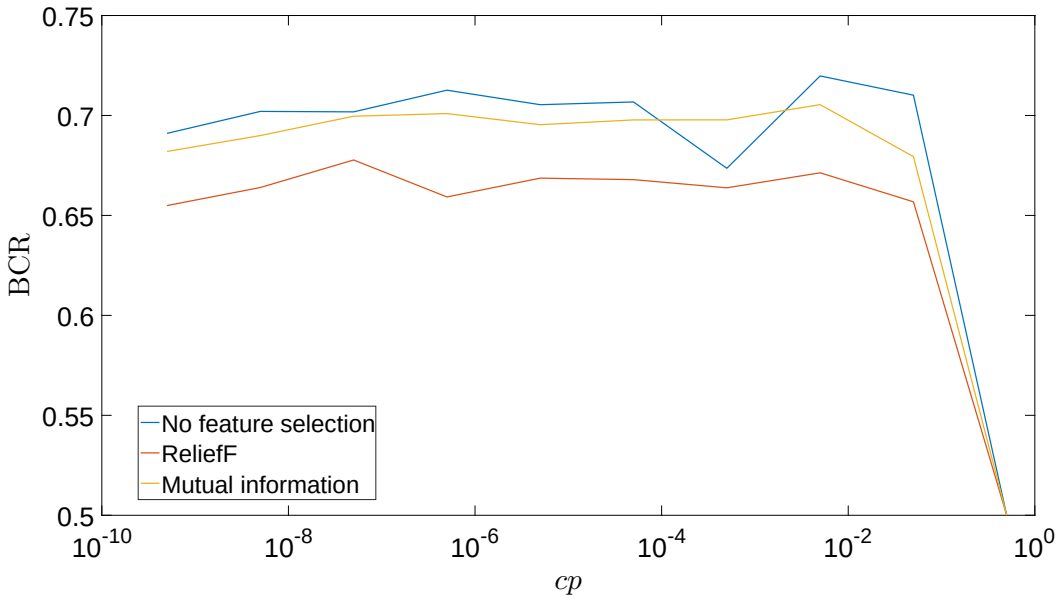

Figure 3.1: BCR for the CART algorithm; the three sets of selected features appear for cp going from 5.10^{-10} to 5.10^{-1} (logarithmic scale for cp).

The second tested algorithm is the RF algorithm. This time, the N_{tree} parameter goes from 500 to 10000, with a step of 500. The tests have been led still with the same cross-validation method. Results are plotted on Figure 3.2.

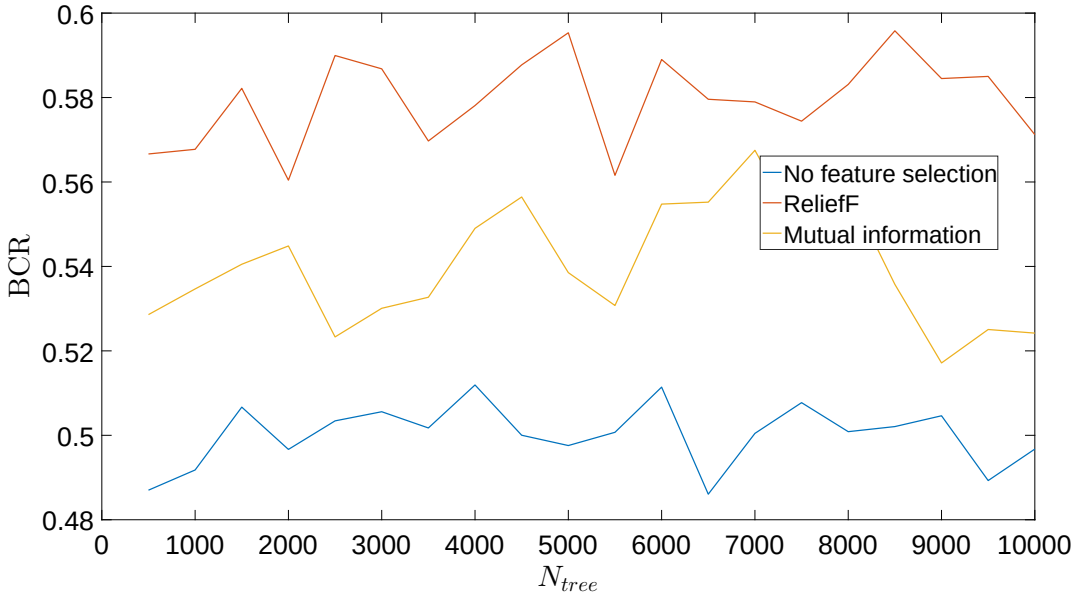

Figure 3.2: BCR for the RF algorithm; the three sets of selected features appear for N_{tree} going from 500 to 10000.

This time, the results obtained seem to be completely chaotic. There is no clear global maximum, whatever the feature set is. Still the feature selection seems to have a (very) slight effect on the BCR, but the three cases cannot make it over a BCR of 0.6.

Finally, the KNN algorithm is tested. The number of neighbours K considered is going from 1 to 30, with a step of 1. All the results are illustrated on Figure 3.3.

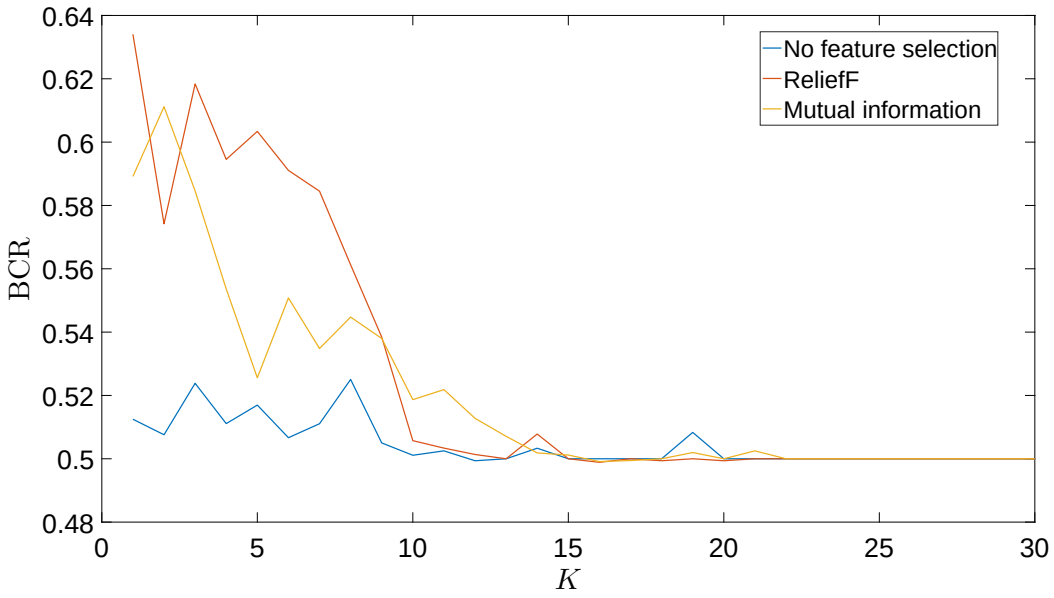

Figure 3.3: BCR for the KNN algorithm; the three sets of selected features appear for K going from 1 to 30.

The figure shows that without feature selection, the KNN algorithm has poor results. This algorithm cannot indeed handle a feature set as big as the full set of the 45 features due to the curse of dimensionality. The two feature selection algorithms enable to improve the BCR with the best choice of parameter, for $K = 1$ (ReliefF) or $K = 2$ (mutual information). Yet the BCR still remains very low against what could have been expected. A maximal value of about 0.635 for the BCR in the best case, cannot lead to the selection of this model.

3.3 Analysis of results

The first thing that comes out of these tests, is that none of them gives full satisfaction. The only one that provides an acceptable result is the CART algorithm without feature selection.

An other observation is that all the algorithms have a good specificity but an awful sensitivity. They indeed could all predict the true negative cases (which are predominant), with a few exceptions, but none of them could find the true positive cases (except for the CART algorithm which could find about half of them). The two other models give poor results that are not acceptable in the framework of a medical study.

Actually, a lot of tests have been conducted, and not all of them have been presented in this report. Other algorithms have been considered: other decision tree algorithms (ID3, C4.5), Weighted Nearest Neighbour classifier as well as Support Vector Machine. Other feature selections have been envisaged, still with the same algorithms but by selecting more or less features. Even other parameters have been modified on the implemented models, parameters that usually don't have any impact on the results but on the creation of the model, to see if it was possible to come up with something that could give better results that the CART algorithm gave. Unfortunately, none of them was successful and therefore there are no better results than the CART algorithm.

So why is it impossible to find something successful, although machine learning is such a powerful tool? One of the possibility is that there are too many binary data. Therefore a lot of decisions are hard and this does not allow a lot of flexibility for the models. An other possibility may simply be that there is no learning possible since the data are not sufficiently correlated to the expected prediction. The different envisaged feature selections did not indeed come out with one or more features that are highly correlated to the risk of recurrence of AF.

Conclusion

A lot of work has been conducted in the context of this thesis, providing though mixed results. Since not a lot of papers have been previously written over A peaks and AA intervals, it was necessary to test a lot of different techniques. Most of them have been ruled out since they did not bring anything useful. This report considers only that could be the subject of discussion.

Regarding the detection of atrial rate, the frequency-based methods have quickly shown their limits since the signals are highly irregular. A new method needed to be created and implemented in order to detect the A peaks. It has been demonstrated that this empirical method could work really well by setting the two needed parameters to their optimal values. However, this set of values is different for all cases. According to these sets, a default set of values has been discovered. This leads to small (even null) errors in many cases, but can lead to considerable errors in some cases. This algorithm is therefore still useful and functional but cannot be used blindly, since the user has to keep an eye on the quality of the result.

For the machine learning part, the feature set could unfortunately not include the AR and maximum AA intervals at the disconnection times of PV since the exact moment of the disconnection was, for most of the patients, unknown. The learning could still be implemented with the other gathered features. This brought unsuccessful results, none of the tested algorithm showing unquestionable BCR. Only the CART algorithm seemed to generate acceptable results (still not enough for a medical study).

This thesis provides opportunities. About the detection part, the creation a Graphic User Interface (GUI) has been attempted. This could have allowed any user to detect the A peaks of a signal by setting the parameter values manually and easily, without browsing through the MATLAB code. And being able to modify them to obtain an instantaneous detection. However it has been impossible to make this GUI work and therefore it has not been included in this work. Regarding the learning part, other leads may certainly be explored. Even though implemented algorithms are not encouraging and seem to lead to a point where no more progress can be made. Other techniques have not been implemented as neural networks, but it would have required a lager population of patients to make its implementation meaningful. Finally some clustering techniques could also be considered.

Bibliography

[1] P. Kirchhof, S. Benussi, D. Kotecha, A. Ahlsson, D. Atar, B. Casadei, M. Castella, H.-C. Diener, H. Heidbuchel, J. Hendriks, *et al.*, "2016 esc guidelines for the management of atrial fibrillation developed in collaboration with eacts," *European heart journal*, vol. 37, no. 38, pp. 2893–2962, 2016.

[2] "Overview of atrial fibrillation." `http://a-fib.com/2-overview-of-atrial-fibrillation`, 2016 (accessed March 10, 2017).

[3] "What is atrial fibrillation?." `https://www.nhlbi.nih.gov/health/health-topics/topics/af`, 2014 (accessed February 28, 2017).

[4] "Atrial fibrillation: Causes, symptoms, types, treatment and more." `https://www.consumerhealthdigest.com/health-conditions/atrial-fibrillation.html`, 2017 (accessed March 10, 2017).

[5] A. . W. Committee *et al.*, "Guidelines for the management of patients with atrial fibrillation," *Circulation*, vol. 114, pp. e257–e354, 2006.

[6] M. A. Miller, A. d'Avila, S. R. Dukkipati, J. S. Koruth, J. Viles-Gonzalez, C. Napolitano, C. Eggert, A. Fischer, J. A. Gomes, and V. Y. Reddy, "Acute electrical isolation is a necessary but insufficient endpoint for achieving durable pv isolation: the importance of closing the visual gap," *EP Europace*, vol. 14, no. 5, p. 653, 2012.

[7] "Savitzky-golay smoothing filters." `http://www.aip.de/groups/soe/local/numres/bookfpdf/f14-8.pdf`, 2016 (accessed April 22, 2017).

[8] "Savitzky-golay filter." `http://www.statistics4u.com/fundstat_eng/cc_filter_savgolay.html`, 2012 (accessed August 15, 2017).

[9] "Find local maxima - matlab findpeaks - mathworks benelux." `https://nl.mathworks.com/help/signal/ref/findpeaks.html`, 2017 (accessed April 23, 2017).

[10] "Has-bled calculator for atrial fibrillation." `http://clincalc.com/cardiology/anticoagulation/hasbled.aspx`, 2017 (accessed August 16, 2017).

[11] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[12] D. Steinberg and P. Colla, "Cart: classification and regression trees," *The top ten algorithms in data mining*, vol. 9, p. 179, 2009.

[13] "Introduction to decision trees and random forests." `http://whrc.org/wp-content/uploads/2016/02/DecisionTrees_RandomForest_v2.pdf`, 2016 (accessed November 30, 2017).

Appendix A

Matlab codes

```
function data = importFile(fileName)
%IMPORTFILE Imports the signals from one patient datafile.
%
%   Input: fileName = name of the patient datafile.
%   Output: data = structure with data.startTime the exact time of the
%                  beginning of the signal, data.fs he sampling frequency
%                  and data.S the signal.

d = importdata(fileName,',');
idx_st_time = strncmp(d.textdata, 'Start time', 10);
data.startTime = d.textdata{idx_st_time}(end-8:end);
idx_label = find(strncmp(d.textdata, 'Label: REF 1-2', 14));
index = str2num(d.textdata{idx_label-1}(end-1:end));
data.fs = str2num(d.textdata{idx_label+4}(end-6:end-2));
data.S = d.data(:,index);

end
```

```
function [t, s] = timeInterval(data, hstart, hend)
%TIMEINTERVAL Extract the signal on an expected time interval.
%
%   Input: data = data sturcture of the patient.
%          hstart = starting time of the interval.
%          hend = ending time of the interval.
%   Output: t = vector with the considered time interval.
%           s = vector with the signal on the considered time interval.

s_hstart = 3600*hour(hstart) + 60*minute(hstart) + second(hstart);
s_hend = 3600*hour(hend) + 60*minute(hend) + second(hend);
s_st_time = 3600*hour(data.startTime) + 60*minute(data.startTime) + second(data.↩
    startTime);
tstart = s_hstart - s_st_time;
tend = s_hend - s_st_time;
t = linspace(tstart,tend,(tend-tstart)*data.fs+1)';
s = data.S(tstart*data.fs+1:tend*data.fs+1);

end
```

```
function [a, a_loc] = detection(s, minDist, minProm)
%DETECTION Detects the A peaks (without plot, to be used in a loop).
%
%   Input: s = signal of the patient.
%          minDist = minimum peak separation.
%          minProm = minimum peak prominence.
%   Output: a = vector with the height of the detected peaks.
```

```
%           a_loc = vector with the indices at which the peaks occur.

[a, a_loc] = findpeaks(abs(s), 'MinPeakHeight', 1000, 'MinPeakDistance',...
    minDist, 'MinPeakProminence', minProm);

end
```

```
function a_loc = plotDetection(t, s, minDist, minProm)
%PLOTDETECTION Detects the A peaks (with plot of the signal and the peaks).
%
%   Input: t = time vector related to signal.
%          s = signal of the patient.
%          minDist = minimum peak separation.
%          minProm = minimum peak prominence.
%   Output: a_loc = vector with the indices at which the peaks occur.

[~, a_loc] = findpeaks(abs(s), 'MinPeakHeight', 1000, 'MinPeakDistance',...
    minDist, 'MinPeakProminence', minProm);

figure;
plot(t,s,'b'); hold on;
plot(t(a_loc),s(a_loc),'*r'); hold on;

end
```

```
%% Disconnection times

% import of the file with the disconnection time of each pulmonary vein
d = readtable('Disconnection times.xlsx');

dt = sort(table2array(d(:,2:end))*duration(24,0,0),2);
names = table2array(d(:,1));

%% Patient name

patient = '___________';

%% Import data

fileName = strcat('E:\ECG data\', patient,'.txt');
data = importFile(fileName);

%% Select time interval

[~, ind] = max(strcmp(patient, names));

% considered pulmonary vein (from 1 to 4)
PV = 2;

hstart = dt(ind, PV);
hend = hstart + duration(0,0,30);

[t, s] = timeInterval(data, char(hstart), char(hend));

plot(t,s)

%% Savitzky-Golay filtering

s2 = sgolayfilt(s,7,21);
t2 = t;

plot(t2,s2)

%% Parameters estimation

minDist = 100:5:300;
```

```
minProm = 1000:50:3000;

param = meshgrid(minDist, minProm);
A = zeros(length(minDist), length(minProm));

for i = 1:length(minDist)
    for j = 1:length(minProm)
        [a, ~] = detection(s2, minDist(i), minProm(j));
        A(i,j) = length(a)/(t2(end)-t2(1))*60;
    end
end

contourf(minDist, minProm, A', 20); colorbar
[Gx, Gy] = gradient(A');
N = sqrt(Gx.^2+Gy.^2);
[M,I] = max(N(:));
[I_row, I_col] = ind2sub(size(N),I);
myMinDist = minDist(I_col);
myMinProm = minProm(I_row);

%% Detection

a_loc = plotDetection(t2, s2, 125.25, 1207.5);

%% Atrial rate variability

AA = t2(a_loc(2:end)) - t2(a_loc(1:end-1));
succAA = AA(2:end) - AA(1:end-1);
SDNN = std(AA);
SDSD = std(succAA);
RMSSD = sqrt(mean(succAA.^2));
AA_max = max(AA);
AR = length(a_loc)/(t2(end)-t2(1))*60;
xlswrite('ARV.xlsx', [SDNN, SDSD, RMSSD, AA_max, AR], PV,...
    strcat('B',num2str(ind+1)));

%% Boxplots

D1 = xlsread('ARV.xlsx',1,'B2:F73');
D2 = xlsread('ARV.xlsx',2,'B2:F73');
D3 = xlsread('ARV.xlsx',3,'B2:F73');
D4 = xlsread('ARV.xlsx',4,'B2:F73');

SDNN_tab = [D1(:,1) D2(:,1) D3(:,1) D4(:,1)];
SDSD_tab = [D1(:,2) D2(:,2) D3(:,2) D4(:,2)];
RMSSD_tab = [D1(:,3) D2(:,3) D3(:,3) D4(:,3)];
AAmax_tab = [D1(:,4) D2(:,4) D3(:,4) D4(:,4)];
AR_tab = [D1(:,5) D2(:,5) D3(:,5) D4(:,5)];

lab = {'1st PV','2nd PV','3rd PV','4th PV'};

figure;
boxplot(SDNN_tab,'labels',lab);
title('SDNN')

figure;
boxplot(SDSD_tab,'labels',lab);
title('SDSD')

figure;
boxplot(RMSSD_tab,'labels',lab);
title('RMSSD')

figure;
boxplot(AAmax_tab,'labels',lab);
title('AAmax')

figure;
boxplot(AR_tab,'labels',lab);
title('AR')
```

Appendix B

R codes

```
# Import libraries ------------------------------------------------------------

library(caret)
library(knncat)
library(rpart)
library(CORElearn)
library(randomForest)
library(mpmi)
library(MASS)
library(BBmisc)
library(e1071)

# Loading data ----------------------------------------------------------------

AFdata <- read.csv("AFmod2.csv", row.names = 1);

# Scaling data ----------------------------------------------------------------

AFdata <- normalize(AFdata, method = "range", range = c(0, 1))

# Feature selection -----------------------------------------------------------

fs.relief <- names(sort(abs(attrEval(RcidiveFA ~ ., AFdata,
  estimator="ReliefFbestK", costMatrix = NULL, outputNumericSplits=FALSE)),
  decreasing = TRUE))[c(1:5,7,8,10:12)]

numdata <- AFdata[sapply(AFdata,is.numeric)]
catdata <- AFdata[sapply(AFdata,is.factor)]
mi.num <- mmi(cts = numdata, disc = AFdata['RcidiveFA'])
mi.cat <- dmi(dmat = catdata)
mi <- c(mi.cat$mi[1,37], mi.num$mi[1:4,1], mi.cat$mi[2:22,37],
  mi.num$mi[5:8,1], mi.cat$mi[23:36,37])
mi.data <- data.frame(t(mi))
colnames(mi.data) <- names(AFdata[,1:44])
fs.mi <- names(sort(abs(mi.data), decreasing = TRUE))[c(1:9,11)]

data.relief <- subset(AFdata, select=c(fs.relief, "RcidiveFA"))
data.mi <- subset(AFdata, select=c(fs.mi, "RcidiveFA"))

# Decision tree ---------------------------------------------------------------

BCR.CART <- function(data, c) {
  ind <- sample(dimnames(data)[[1]],45)
  train <- data[ind,]
  test <- data[setdiff(row.names(data),ind),]
  model <- rpart(RcidiveFA ~ ., data = train, method = "class", control = rpart.↩
      control(cp = c))
  pred <- predict(model, train, type = "class")
  t <- table(pred, train$RcidiveFA)
```

```
  (t[1,1]/(t[1,1]+t[2,1]) + t[2,2]/(t[1,2]+t[2,2]))/2
}

N <- 50
cp.range <- 5*10^seq(-10, -1, by = 1)

BCR.CART.all <- matrix(0, nrow = length(cp.range), ncol = N)
BCR.CART.relief <- matrix(0, nrow = length(cp.range), ncol = N)
BCR.CART.mi <- matrix(0, nrow = length(cp.range), ncol = N)

for(i in 1:length(cp.range)) {
  for(j in 1:N) {
    BCR.CART.all[i,j] <- BCR.CART(AFdata, cp.range[i])
    BCR.CART.relief[i,j] <- BCR.CART(data.relief, cp.range[i])
    BCR.CART.mi[i,j] <- BCR.CART(data.mi, cp.range[i])
  }
}

BCRcp.all <- sapply(1:length(cp.range), function(x) mean(BCR.CART.all[x,]))
BCRcp.relief <- sapply(1:length(cp.range), function(x) mean(BCR.CART.relief[x,]))
BCRcp.mi <- sapply(1:length(cp.range), function(x) mean(BCR.CART.mi[x,]))

plot(cp.range, BCRcp.all, type = "l", col = "blue", xlab = "cp", ylab = "BCR", log = "↩
    x")
lines(cp.range, BCRcp.relief, col = "red")
lines(cp.range, BCRcp.mi, col = "green")


# Random forests ---------------------------------------------------------

mdl.rf <- randomForest(training[,1:44], y = training[,45], xtest = testing[,1:44], ↩
    ytest = testing[,45], ntree=5000)

BCR.RF <- function(data, n) {
  ind <- sample(dimnames(data)[[1]],45)
  train <- data[ind,]
  test <- data[setdiff(row.names(data),ind),]
  model <- randomForest(RcidiveFA ~ ., train, ntree = 5000, mtry = 6)
  pred <- predict(model, test, type = "class")
  t <- table(pred, test$RcidiveFA)
  (t[1,1]/(t[1,1]+t[2,1]) + t[2,2]/(t[1,2]+t[2,2]))/2
}

N <- 50
n.range <- seq(500, 10000, by = 500)

BCR.RF.all <- matrix(0, nrow = length(n.range), ncol = N)
BCR.RF.relief <- matrix(0, nrow = length(n.range), ncol = N)
BCR.RF.mi <- matrix(0, nrow = length(n.range), ncol = N)

for(i in 1:length(n.range)) {
  for(j in 1:N) {
    BCR.RF.all[i,j] <- BCR.RF(AFdata, n.range[i])
    BCR.RF.relief[i,j] <- BCR.RF(data.relief, n.range[i])
    BCR.RF.mi[i,j] <- BCR.RF(data.mi, n.range[i])
  }
}

BCRn.all <- sapply(1:length(n.range), function(x) mean(BCR.RF.all[x,]))
BCRn.relief <- sapply(1:length(n.range), function(x) mean(BCR.RF.relief[x,]))
BCRn.mi <- sapply(1:length(n.range), function(x) mean(BCR.RF.mi[x,]))

plot(n.range, BCRn.all, type = "l", col = "blue", xlab = "ntree", ylab = "BCR")
lines(n.range, BCRn.relief, col = "red")
lines(n.range, BCRn.mi, col = "green")


# KNN --------------------------------------------------------------------

BCR.KNN <- function(data, K) {
```

```
  ind <- sample(dimnames(data)[[1]],45)
  train <- data[ind,]
  test <- data[setdiff(row.names(data),ind),]
  model <- knn3(RcidiveFA ~ ., train, k = K)
  pred <- predict(model, test, type = "class")
  t <- table(pred, test$RcidiveFA)
  (t[1,1]/(t[1,1]+t[2,1]) + t[2,2]/(t[1,2]+t[2,2]))/2
}

N <- 50
K.range <- seq(1, 30, by = 1)

BCR.KNN.all <- matrix(0, nrow = length(K.range), ncol = N)
BCR.KNN.relief <- matrix(0, nrow = length(K.range), ncol = N)
BCR.KNN.mi <- matrix(0, nrow = length(K.range), ncol = N)

for(i in 1:length(K.range)) {
  for(j in 1:N) {
    BCR.KNN.all[i,j] <- BCR.KNN(AFdata, K.range[i])
    BCR.KNN.relief[i,j] <- BCR.KNN(data.relief, K.range[i])
    BCR.KNN.mi[i,j] <- BCR.KNN(data.mi, K.range[i])
  }
}

BCRK.all <- sapply(1:length(K.range), function(x) mean(BCR.KNN.all[x,]))
BCRK.relief <- sapply(1:length(K.range), function(x) mean(BCR.KNN.relief[x,]))
BCRK.mi <- sapply(1:length(K.range), function(x) mean(BCR.KNN.mi[x,]))

plot(K.range, BCRK.all, type = "l", col = "blue", xlab = "K", ylab = "BCR")
lines(K.range, BCRK.relief, col = "red")
lines(K.range, BCRK.mi, col = "green")
```

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**