Master thesis : Exploring the Fractal Dynamics of the Heart Rate: Modeling and Analysis of Components in Health and Disease

Auteur : de la Brassinne Bonardeaux, Ophélie
Promoteur(s) : Sacré, Pierre
Faculté : Faculté des Sciences appliquées
Diplôme : Master : ingénieur civil en informatique, à finalité spécialisée en "management"
Année académique : 2022-2023
URI/URL : http://hdl.handle.net/2268.2/17701

University of Liège - Faculty of Applied Science

Exploring the Fractal Dynamics of the Heart Rate

Modeling and Analysis of Components in Health and Disease

Master's thesis completed in order to obtain
the degree of Master of Science in Computer Engineering
by de la Brassinne Bonardeaux Ophélie

Author :

de la Brassinne Bonardeaux Ophélie

Academic Supervisor :

Sacré Pierre

Academic year 2022-2023

Abstract

The presence of a fractal pattern, characterized by self-similarity at various scales, has been observed in numerous physiological signals. This thesis specifically investigates the fractal dynamics of the heart rate. The main objective of this project is to analyze the underlying mechanisms that contribute to this complex structure, aiming to gain a deeper understanding of its nature and how it is influenced by different diseases. The study focused on three distinct groups which included healthy subjects, individuals with congestive heart failure, and those with atrial fibrillation. The study aimed to adapt and expand a stochastic model proposed by *Ivanov et al.* [21], which originally focused on modeling the fractal dynamics of the heart rate in healthy individuals. The model assumes that the heart rate is influenced by the parasympathetic and sympathetic systems. In this project, the model was re-implemented and extended to incorporate diseased subjects. The extended stochastic model accurately represented the fractal dynamics of the heart rate, showing strong correlation with observed patterns. Atrial fibrillation resulted in a complete loss of fractal dynamics at low scales, replaced by a random signal. Congestive heart failure had a lesser impact but showed a reduction in the influence of the sympathetic and parasympathetic systems. These findings highlight the model's strength in elucidating complex physiological processes. Looking ahead, further advancements can be made by incorporating additional factors, paving the way for even more comprehensive understanding.

Acknowledgments

I would like to thank my supervisor, Prof. Pierre Sacré, as well as Dr. Grégory Hammad and Dr. Christina Schmidt for offering me the opportunity to work on this project and helping and advising through the year.

I would also like to thank Dr. Philippe Marcelle and Dr. Sebastien Robinet for taking the time to review one of my models and providing medical interpretation.

I am grateful to Arnaud Delaunoy for his assistance with neural networks.

Finally, I would like to thank my family for their unwavering support throughout the entire duration of this project.

Contents

Chapter 1

Getting Started

1.1 Introduction

Fractals are complex structures that exhibit self-similarity at different scales and have been observed in a wide range of natural systems. In particular, the fractal pattern has been observed in many biological signals, including the heart rate, which is an important indicator of cardiovascular health. This pattern is believed to reflect the complex interactions between the different physiological systems that regulate the heart rate. This thesis aims to analyze the fractal dynamics of the heart rate using various techniques and models and investigate the differences in the fractal pattern between healthy and diseased subjects.

1.1.1 Background and Significance

Fractal patterns have been observed in many physiological signals, including gait [15, 20, 18, 29] and heart rate [12, 11, 42]. These signals display complex patterns that cannot be easily characterized by traditional signal analysis methods. In 1994, *Peng et al.* [32] introduced a method that quantifies the long-range correlation observed in signals with a fractal pattern. This method, called detrended fluctuation analysis (DFA), has since been widely used to identify fractals in various biological signals [15, 20]. In particular, DFA has been used to investigate how the fractal pattern of heart rate signals changes with the progression of diseases or aging [33, 12].

The dynamics of the heart rate have been the subject of numerous mathematical and statistical modeling attempts. One of the earliest models was introduced by *Warner and Cox* [35], who proposed various mathematical equations to predict the heart rate based on sympathetic and vagus efferent nerves to the heart. Subsequently, several other models emerged, with many of them adopting statistical approaches such as autoregressive models [6] and point process models [3]. In a more specific context, *Stirling et al.* [39] developed a model for heart rate response to exercise using differential equations, considering the dependence of heart rate on time and oxygen demand. More recent advancements have explored the application of machine learning techniques. Neural networks, ranging from simple artificial neural networks [41] to more complex models incorporating Long Short-Term Memory (LSTM) networks, have been investigated. The utilization of LSTMs enables capturing correlations between interbeat intervals, often augmented with attention mechanisms [26].

Despite the existence of several models with compelling features and outcomes, none of them have incorporated the fractal dynamics of heart rate. The only model identified that considers the fractal pattern is the one proposed by *Ivanov et al.* [21] which models the heart rate of healthy subjects. Given that the primary focus of this study is on fractal dynamics, this model appears to be the most pertinent foundation for modeling the fractal dynamics of heart rate.

The objective of this project is to explore this existing stochastic model developed for modeling the heart rate and adapt it to incorporate diseased patients. By doing so, we aim to gain a deeper understanding of the effects of these diseases on the fractal dynamics of the heart rate and how these changes can be modeled. In addition, this project aims to evaluate the performance of these models in differentiating between healthy and diseased patients based on their heart rate dynamics. This

information could be used to develop new diagnostic tools for cardiovascular diseases and to better understand the underlying mechanisms of these conditions.

1.1.2 Structure of the Thesis

To achieve the objectives of this thesis, the following structure is adopted.

In the remaining sections of this chapter, section 1.2 provides a brief overview of fractals and the heart rate, which are essential concepts for understanding this project. Following that, section 1.3 introduces the two datasets utilized in this work.

Chapter 2 focuses on analyzing the structure of fractal signals. This chapter is divided into two parts: first, we explore the methods used to analyze fractal signals (section 2.1), and then we apply the most suitable method to heart rate signals (section 2.2).

Once a solid understanding of fractals is established, the subsequent focus is on modeling the fractal pattern of heart rate. Chapter 3 explores the Ivanov model, a stochastic model that has proven successful in generating fractal heart rate signals. In section 3.1, we present the model and identify the parameters that influence the fractal pattern of the generated signals. In sections 3.2, 3.3, and 3.4, we explore three different methods to fit the most appropriate parameters on real heart rate datasets, which include data from healthy subjects, as well as individuals with congestive heart failure and atrial fibrillation. Finally, in section 3.5, we analyze the quality of the parameters estimation for each subject and provide an interpretation.

Lastly, in chapter 4, we briefly explore generalized linear models, another potential heart rate model and evaluate its potential for fractal modeling.

1.1.3 Code Availability

All the different methods employed and discussed in this study were implemented using the Python programming language within a Jupyter Notebook environment. The documented source codes can be accessed from the following GitHub repository:
`https://github.com/OdlBB/ATFE0015-1-Fractality-as-a-health-indicator`.

1.2 Context

1.2.1 What is a Fractal?

When someone evokes the word fractal, most people think of geometric figures that self contain themselves, i.e. figures that whatever the scale at which we look, we will always see the same figure. Some known examples are the Koch snowflake (Figure 1.1) or the Sierpinski carpet. If these figures are indeed fractals, they are just a small subset of the whole set and represent an extreme example of what a fractal actually is. What characterises a fractal is the self similarity (and not the self-containing) at all scales. Let's take the example of the coastline of Britain. At a scale of 10km, the beaches and cliffs can be seen. Once zoomed at a scale of 1km, new small beaches appear. This keeps happening by zooming again and again[1]. As can be seen on Figure 1.2, even though the pictures were taken at different scales, the coast line remains pretty similar. However, at a certain point, such as the molecular scale, this phenomenon is no longer observable. Self-similarity at all scales is an ideal case, and in reality, the range of scales at which fractals can be observed is limited by upper and lower bounds.

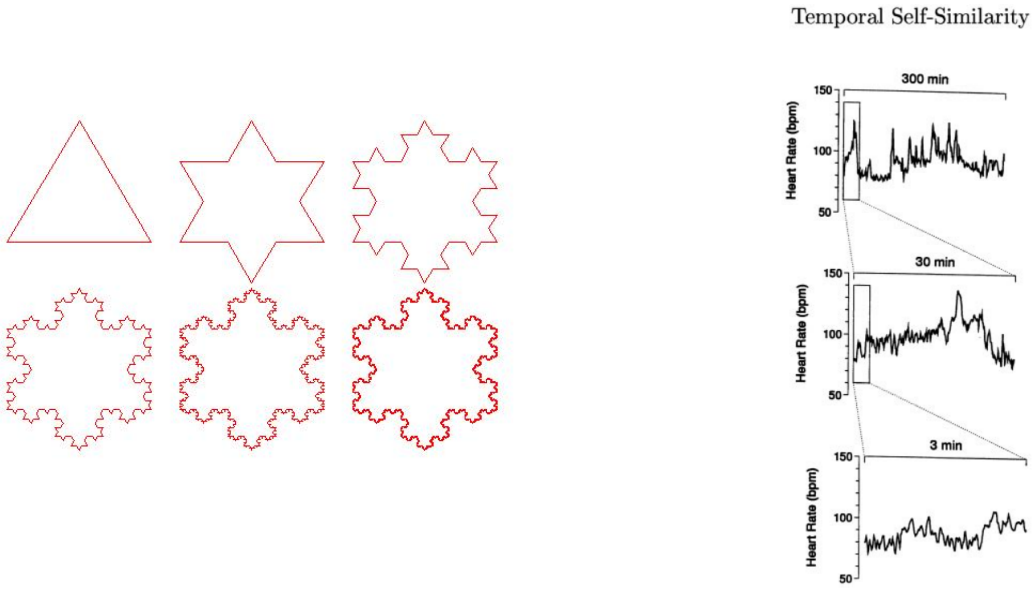


Figure 1.1: Left: Koch snowflake. Right: A fractal temporal process at different time scales. *Figure from ref [12].*

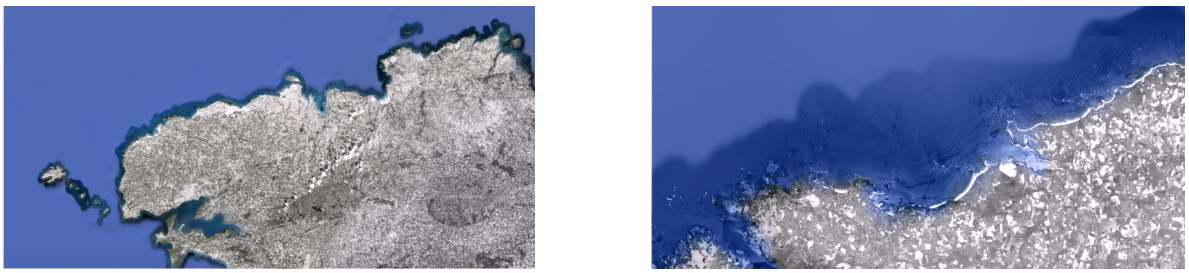


Figure 1.2: Britain coast at scales of 10km (left) and 1km (right). Source: `https://www.youtube.com/watch?v=pAiR7g5xzaQ`.

Another characteristic of fractals is that they have an infinite perimeter (due to the self-similarity at all time scales) but a finite surface. Let's consider the Britain coastline again. Obviously, the surface of Britain is finite. However, the perimeter will grow with the precision. This happens because the more we zoom in and increase the precision, the more details of the coastline such as cliffs and small beaches become visible, causing the perimeter value to increase.

[1]More details in `https://www.youtube.com/watch?v=pAiR7g5xzaQ`

1.2.1.1 Fractals in Time Series

In the context of this project, the concept of fractal patterns is applied to time series analysis and more specifically on biological signals. As depicted in Figure 1.1, some signals also exhibit self-similarity.

Before diving into the fractal signal, we first need to know what are white and Brownian noise.

White noise is a type of noise that has a flat frequency spectrum meaning that it contains all frequencies at equal intensity. It is called "white" because it is analogous to white light, which contains all visible colors at equal intensity. In terms of time series, it consists in a time series with

- a zero mean,
- a finite standard deviation that is constant in time,
- no correlation between the lags.

Brownian noise is the integration of white noise. It has a power spectral density that is inversely proportional to the frequency squared, which means that its power decreases as the frequency increases. It is also referred as random walk noise, meaning that the time series can be built by using a random walk process. It has the following properties apart from the ones in table 1.1:

- **Drift:** non-zero mean drift, which means that over time, the values tend to move in one direction or another;
- **Diffusion:** the values tend to spread out over time due to random fluctuations. This is in contrast to white noise, which has equal variance at all times.

The fractal signal is more commonly referred as "1/f" noise or pink noise[2] [23]. It exhibits a power spectral density that is inversely proportional to the frequency. This means that the power of the noise decreases as the frequency increases. In terms of that definition, it is also referred as a noise that is between white and Brownian noise (see Figure 1.3). Additionally to the characteristics presented in table 1.1, 1/f noise also exhibits persistence which refers to the characteristic of the noise to exhibit long-lasting fluctuations and variations that persist over time without quickly diminishing or transitioning to different states.

Property	1/f noise	White noise	Brownian noise
Randomness	The values of the sequence are not predictable or deterministic.	The values of the sequence are not predictable or deterministic.	The values of the sequence are not predictable or deterministic.
Stationarity	The statistical properties change over time.	The statistical properties don't change over time.	The statistical properties change over time.
Autocorrelation	The correlation between two points decreases slowly as the time that separates them grows.	No correlation between the points.	Due to the random walk structure, each sample depends on the previous one, resulting in a correlation between successive values.
Memory	The value at any given time depends not only on its immediate past but also on its more distant past.	Independence between the points.	The value at any given time is influenced by the values that came before it.

Table 1.1: Statistical characteristics comparison of white noise, 1/f noise and Brownian noise

[2]More details about 1/f noise in `https://www.youtube.com/watch?v=ToZFyIvnQnI&t=492s`

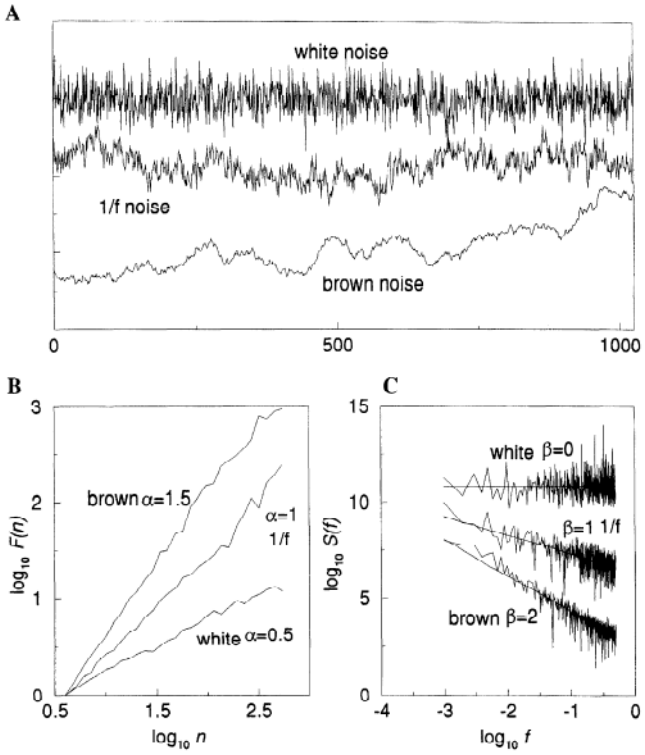


Figure 1.3: (A) Simulation of the noise signals (B) Detrended fluctuation analysis (DFA) (C) spectral analysis. *Figure from ref [29].*

1.2.2 Heart Rate

In the heart rate, what is actually measured is the interbeat time, i.e. the time between two beats of the heart [2]. One could think that the heart rate follows a smooth curve but it turns out to be quite "chaotic". Actually, in the article [12], the authors explain that a constant heart rate is associated with diseased people.

In order to determine the heart rate, we first need to get the ECG signal. This signal is a representation of the electric activity of the heart. The electric pulse is what enacts the contraction of the heart. The form of the electric signal during one heart beat is commonly called the "PQRST" form. In order to explain this "wave", we first need to have a bit of knowledge of the heart anatomy. The heart has four different cavities: the right and left atrium and the right and left ventricle. The atria are the cavities in which the blood enters and the ventricles are the ones from which the blood leaves the heart. These last two cavities are actually the most powerful and the ones responsible for all the blood circulation. Let's now get back to the electric signal. In Figure 1.4, we have two "PQRST" waves. The "P" part corresponds to the electric signal propagating into the atria, while the "R" part is the electric signal propagating into the ventricles. The interbeat interval is measured by taking the time between two "R" peaks. A representation of the whole process is available on Figure 1.4.

The heart rate can be influenced by various factors. In the context of this project, we consider three of them:

1. **The sinoatrial (SA) node** is a small cluster of specialized cells located in the right atrium of the heart, and it serves as the natural pacemaker of the heart. The SA node generates electrical impulses that initiate each heartbeat, and these impulses travel through the atria to the atrioventricular (AV) node, and then to the ventricles, causing them to contract and pump blood throughout the body.

2. **The parasympathetic nervous system**, which is also known as the "rest and digest" system, tends to slow down the heart rate. The parasympathetic fibers that innervate the heart release the neurotransmitter acetylcholine, which binds to receptors on the SA node and slows down

the rate of electrical impulses generated by the node. The parasympathetic nervous system is primarily active during periods of rest and relaxation, such as when you're sleeping or sitting quietly.

3. **The sympathetic nervous system**, which is also known as the "fight or flight" system, tends to speed up the heart rate. When the body perceives a threat or stressor, the sympathetic nervous system is activated, and sympathetic fibers that innervate the heart release the neurotransmitter norepinephrine, which binds to receptors on the SA node and increases the rate of electrical impulses generated by the node. This prepares the body for action by increasing blood flow to the muscles and organs and increasing the heart rate and blood pressure.

Of course, the heart rate can be influenced by other factors such as physical activities, emotions, body temperature, ... However, these are beyond the scope of this project.

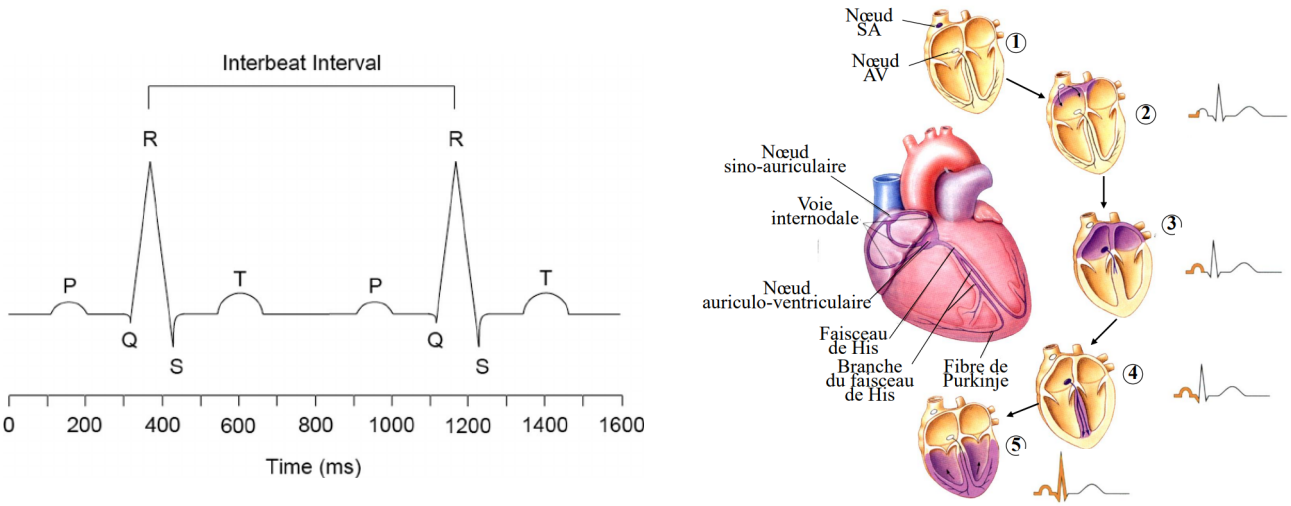


Figure 1.4: On the left: Heart interbeat. On the right: Heart beat process. *Figure from ref [24]*

1.2.2.1 Diseases

In this work, we analyse the heart rate of healthy and diseased subjects and compare them. More specifically, we consider two different diseases:

1. congestive heart failure,
2. atrial fibrillation.

The first one is a condition that occurs when the heart, and more specifically the ventricles loose pumping power, thus not meeting the body's needs. This often occurs when the muscle is weakened or damaged. One of the possible reactions to this lack of pumping can be an acceleration in the heart rate in order to maintain a sufficient blood flow to the body.

The second one also impacts the heart rate but in a different way: it impacts the electrical impulses. As briefly explained earlier, in order to get a beat, an electric signal needs to be propagated into the heart and this is done in a very specific order:

1. The signal starts from a region on the right atrium called the **sinoatrial node** (also called the natural pacemaker).
2. The signal is then propagated through the atria, so that the blood in these cavities flows into the ventricles.
3. The electrical impulses then go through the ventricles, leading to the contraction ejecting the blood out of the heart.

Note that this is a simplified version of what is actually happening but it is enough for the scope of this work.

In atrial fibrillation, the subject has an irregular heartbeat due to the electrical impulses of the heart becoming disorganised, leading to contractions in the atria to be totally desynchronised and thus inefficiently making the blood flow from the atria to the ventricles. Furthermore, the contractions no longer start at the sinoatrial node but now start anarchically anywhere in the atria.

1.3 Datasets

1.3.1 First Dataset: ECG for Healthy Aging

The first dataset is made of various ECGs, grouped by age (see table 1.2) that were recorded from 1.121 healthy subjects. This dataset is available on physionet[3]. The length of the recording varies between 8 and 45 minutes. The fractal pattern isn't something observed on ECG signals but on interbeat signals, thus the signals need to be processed before being used.

However, as will be developed in section 2.2.1, the time series obtained from this dataset are too short to observe the fractal pattern and in the future analysis, only the second dataset will be used.

Group	1	2	3	4	5	6	7
Size	46	422	229	105	43	50	50
Age	18-19	20-24	25-29	30-34	35-39	40-44	45-49
8	9	10	11	12	13	14	15
50	18	24	19	13	7	12	7
50-54	55-59	60-64	65-69	70-74	75-79	80-84	85-92

Table 1.2: Age groups of the first dataset

1.3.2 Second Dataset: Heart Rate of Healthy and Sick Subjects

The second dataset is available on physionet[4]. This dataset offers 15 heart rates signals, each time series being around 24 hours long. The set can be divided into 3 groups:

1. **group n:** this group contains 5 times series of healthy subjects,
2. **group c:** this group contains 5 times series of subjects with congestive heart failures,
3. **Group a:** this group contains 5 times series of subjects with atrial fibrillation.

As a reminder, congestive heart failure occurs when the heart muscle doesn't pump the blood as well as it should and can lead to an acceleration of the heart rate. Atrial fibrillation occurs when the electric activity in the heart goes wrong and thus the contractions in the atrium are totally desynchronised often leading to a fast and irregular heart rate (more detail in section 1.2.2.1).

It happened that some signals contained abnormal heart beats (especially the ones of subjects with congestive heart failures). The intervals that precede and follow these beats should be considered as outliers. For this, a second series was proposed with these outliers being removed. DFA (see section 2.1.1) being very sensitive to outliers, this second series were the ones that were used for further analysis. However, it is important to note that not all of the outliers were removed and a few still remained. Additional information can be found in the dataset documentation.

[3] `https://physionet.org/content/autonomic-aging-cardiovascular/1.0.0/`

[4] `https://physionet.org/content/chaos-heart-rate/1.0.0/`

Chapter 2

Identify the Fractal Pattern on Time Series

2.1 Methods Overview

The first step is to determine the structure of a time series that has fractal patterns and how to characterise it. It has been determined that the fractals can be associated with "1/f" noise (see section 1.2.1.1). There are several ways to characterise these different signals but the two most used methods are the detrended fluctuation analysis (DFA) [15, 20] and the power spectrum analysis [11]. Both methods consist in representing a data with a curve and summarise that data with a value (α for the DFA method and β for the power spectrum analysis). As can be seen in Figure 1.3, the value determines which kind of signal we have. These two methods are actually linked since $\beta = 2\alpha - 1$ [5]. Both are used to evaluate what is called the Hurst exponent, which is a measure of the present correlation in a time series.

2.1.1 Method 1: Detrended Fluctuation Analysis

Method overview The detrended fluctuation analysis (DFA) is a method that quantifies the presence of long-term correlations, or self-similar behavior, in time series data. The method, initially introduced by *Peng et al.* [32] for analyzing DNA sequences, has gained popularity and become a widely utilized technique. It has found applications in the analysis of various signal types, including physiological signals, financial markets, and climate data, and has received over 3.000 citations [1].It can be performed with the following steps[2] [32, 19]:

1. convert the signal to a mean-centered cumulative sum,
2. define the various "scales",
3. for a given scale, split the data into epochs according to the size of that scale,
4. detrend each epochs,
5. compute the root mean square of each epoch

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{n} x_i^2},$$

6. take the mean of all the RMSs as a summary value for that scale,
7. repeat 3 for each scales,

[1] `https://en.wikipedia.org/wiki/Detrended_fluctuation_analysis`
[2] More details in `https://www.youtube.com/watch?v=-RmxLZF8adI`

8. compute the linear fit between log-scales and log-RMS, the slope α of this line will is the coefficient that summarizes the time series.

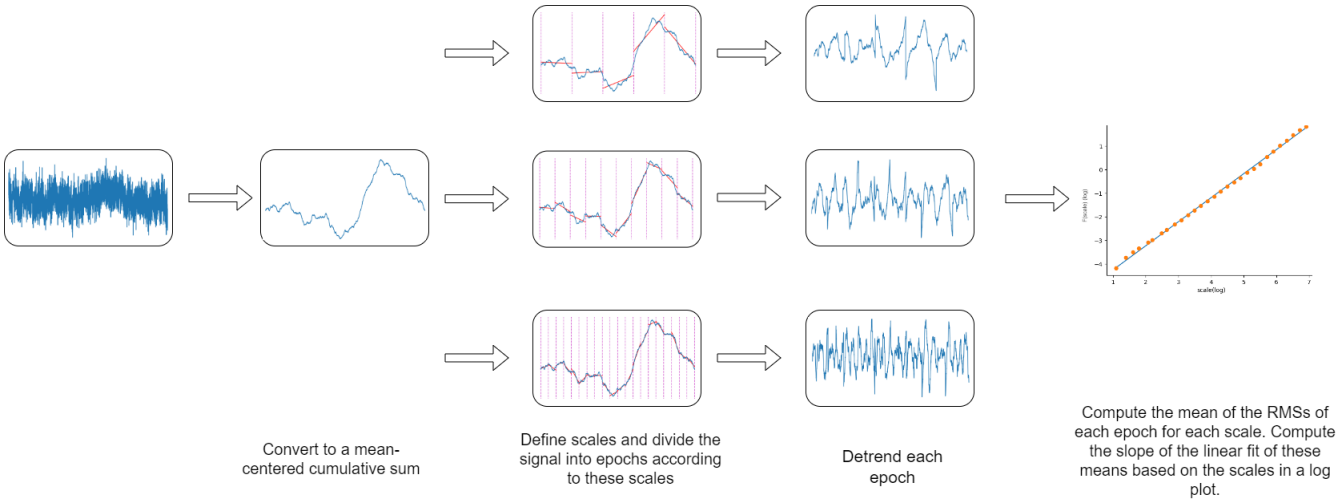


Figure 2.1: This figure illustrates the different stages of the DFA method. The first image shows the original signal, while the next image displays the signal converted to a mean-centered cumulative sum (step 1). The middle images depict examples of the signal divided into different time scales, in which these these sub-signals are detrended (steps 2, 3, 4 and 5), and the image on the right shows the final curve (steps 6, 7 and 8). The slope of this curve corresponds to the value of the coefficient α.

Results interpretation Based on the computed value, the type of signal can be identified (see Figure 1.3):

- anti-correlated time series if $\alpha < 0.5$,
- white noise, uncorrelated time series if $\alpha \approx 0.5$,
- correlated time series if $\alpha > 0.5$,
- 1/f noise if $\alpha \approx 1$,
- Brownian noise if $\alpha \approx 1.5$.

From there, the Hurst exponent H can be identified:

- if $\alpha \in [0, 1]$, the signal is a **fractional Gaussian noise** and $H = \alpha$,
- if $\alpha \in]1, 2]$, the signal is a **fractional Brownian motion** and $H = \alpha - 1$.

Trend of time series The trend of a time series is the general direction in which the values of the time series are changing over time. It represents the long-term behavior of the time series, indicating whether the series is increasing, decreasing, or remaining relatively stable over time.

Detrending is the process of removing the trend component from a time series data, leaving behind only the variation in the series that is not explained by the trend. One of the most common trend is a linear trend, meaning a straight line can be fitted. Detrending this series simply consists of removing the straight line best fit [19].

In the context of detrended fluctuation analysis, the local trend is determined and then removed. The used DFA method is defined by an order N which is the order of the polynomial that will be fitted to the epochs [5, 19]. For example, the method DFA-2 will detrend the epochs using a polynomial of order 2.

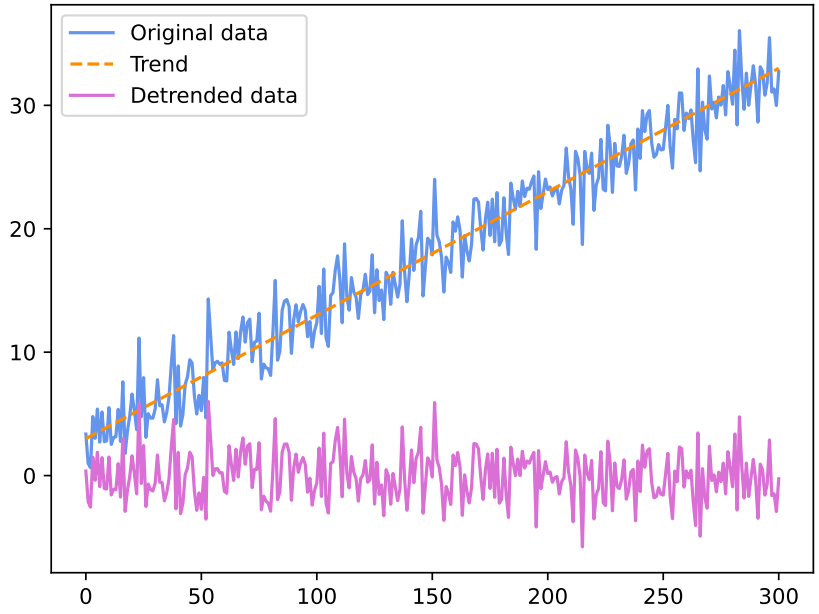


Figure 2.2: Linear detrending of a time series

Intuition of DFA The first step (convert the signal into its mean-centered cumulative sum) is crucial as it helps determine some trends that wouldn't be perceived if the original time series was used. This creates a random walk that is dominated by the large-scale fluctuations in the data, while short-term fluctuations are integrated out. This has the effect of amplifying the long-term correlations in the data and reducing the effects of noise and short-term variability.

If we have a look at the method, it first approximates the time series by making linear regressions into time windows and then measures how much it is impacted by the size of that time window i.e. the variance of the regressions for a given time window size. If the signal can have quite large time windows and remain relatively precise, then logically, the coefficient α will be quite high as at small windows the variance in the detrended signal will be very small (since it was very well approximated) but eventually, it will loose its precision. On the other hand, if the time series needs very small windows to be accurately estimated, it doesn't change much that there is a small or a large time window, there will be a high variance in both, thus the coefficient α will be small, since the variability will (faintly) grow [3].

If we have a look at Figure 2.3, we can see that for white noise, even with small windows, the variance will already be quite high and will remain high whatever the time window, thus the slope of all the RMSs computed will be low as no big difference will rise if the signal is approximated with small or large time windows. On the other hand, for Brownian noise, the slope will be high. Indeed, for small time windows, the regressions will be very precise and it will gradually loose in precision.

[3]More details in `https://www.youtube.com/watch?v=o0LndP2OlUI&t=322s`

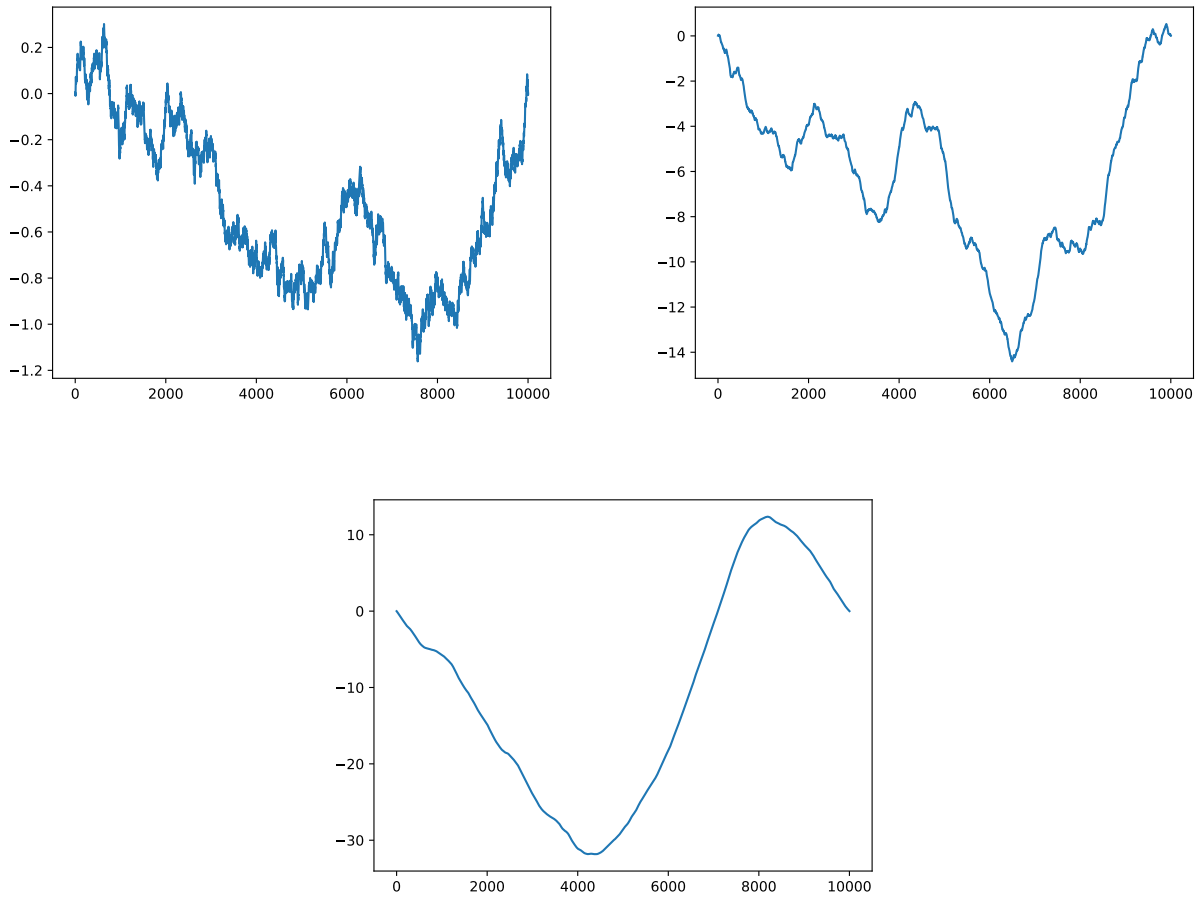


Figure 2.3: Integration of time series. On the top left side: white noise, on the top right side: 1/f noise, on the bottom: Brownian noise.

2.1.1.1 Implementing and Assessing the Method

In order to perform DFA, two different implementations of the method were used:

1. the first was built from scratch following the various steps described in section 2.1.1,
2. the second was already implemented in the `MFDFA` package[4].

Assessing the method Before performing DFA on any dataset, we first need to check that these methods are accurate. For that, several signals were generated with a given value for the coefficient α. DFA (using both implementations) was then performed on these signals to give the estimated α^*. From there, we can compare α with α^* and determine the accuracy of the methods.

The `stochastic` package[5] was used to generate the signals. That package provides a method called `FractionalGaussianNoise` that generates a random signal with a given Hurst exponent. Once that signal was generated, both DFA methods could be called to assess the value. However, it is not possible to generate signals with α above 1 (definition of fractional Gaussian noise, see section 2.1.1). Nonetheless, future analyses will include signals that are fractional Brownian motions. The `stochastic` package doesn't offer other methods that could build a signal based on the Hurst exponent but it has the possibility to generate a signal based on its power spectrum coefficient β (more details in section 2.1.2) with the `ColoredNoise` method. With the relation $\beta = 2\alpha - 1$, it was possible to generate signals with the expected value of $\alpha \in [0.4, 1.6]$. Each signal consisted of 10.000 points.

[4] `https://github.com/LRydin/MFDFA`
[5] `https://github.com/crflynn/stochastic`

NB: another package called `FBM`[6] was also considered to generate random signals based on the Hurst exponent. However, due to technical limitations, the methods proposed by this package were soon discarded as the signals generated with a Hurst exponent close to 0 or 1 weren't accurate.

Scale impact In order to verify whether it is better to use small scales or scales up to the dataset length, the code was launched twice: once with scales between 10 and 100 and once with scales between 3 and 1000.

The results on Figure 2.4 contain various interesting information. First of all, when lower scales are used, the second method (i.e. the one from the `MFDFA` package) is more accurate than the first one (the one that was implemented). On the other hand, if larger scales are used (up to the signal length), the first version tends to become more accurate while the second version becomes less accurate. This leads to no method "standing out", but the difference between the methods with larger scales is less than when smaller scales are considered. This is one element that could be in favor of the second method. Both methods are quite accurate at all scales but some combinations are better than others.

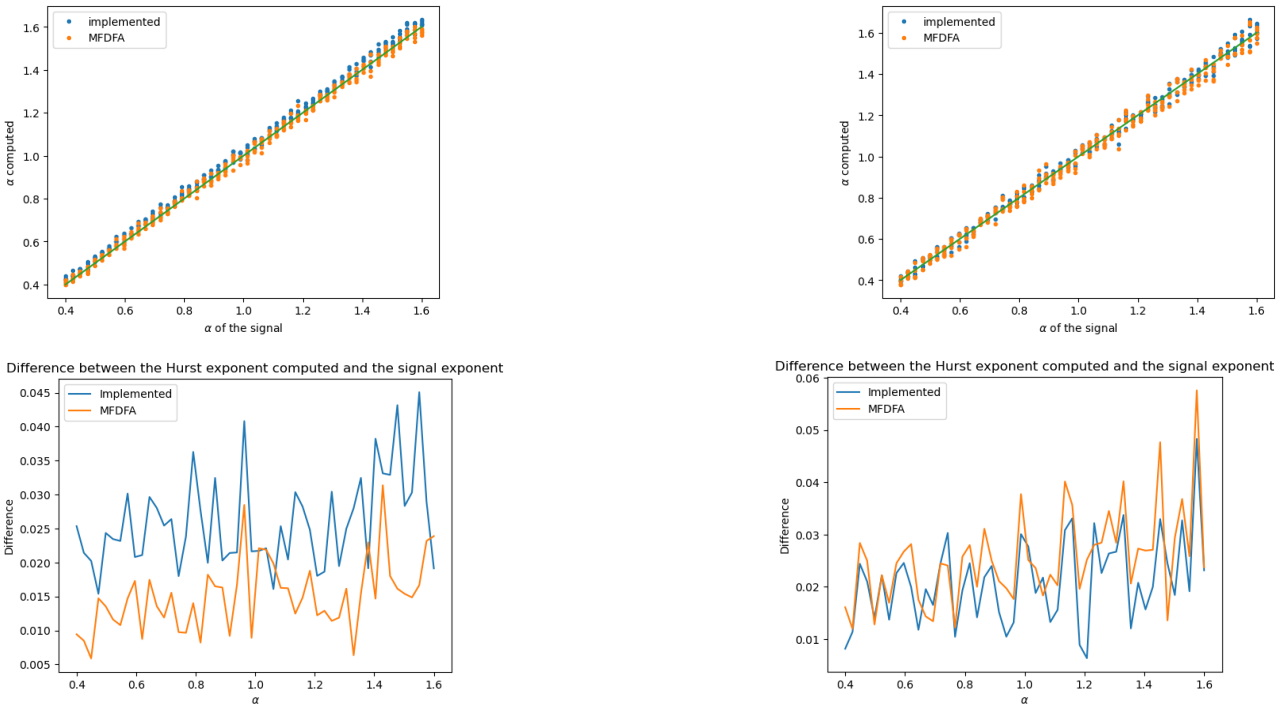


Figure 2.4: This figure presents the results obtained from performing DFA on signals generated with the `stochastic` package of size 10.000. The graphs on the left were obtained from the experiment using scales in the range [10, 100], while the graphs on the right were generated from scales in the range [10, 1000]. The top graphs display the exact values of α along with all the computed values α^*. For each α coefficient, five different signals were generated and the computed values for each signal were subtracted from the actual value. The resulting absolute values were averaged. The bottom graphs show the evolution of these mean absolute errors.

Time series length impact The tests described earlier only utilized time series of a single size. However, in future analyses, longer time series with more than 70.000 values will be considered. To ensure that the DFA method is effective for such larger sizes, the same analysis was repeated with time series of random sizes selected from the interval [10.000, 200.000]. The chosen scales ranged from 10 to $N/10$ with N being the size of the time series. The results are presented in Figure 2.5, indicating that the DFA method was actually more accurate with these longer signals than with those shown in Figure 2.4.

[6] `https://github.com/732jhy/Fractional-Brownian-Motion`

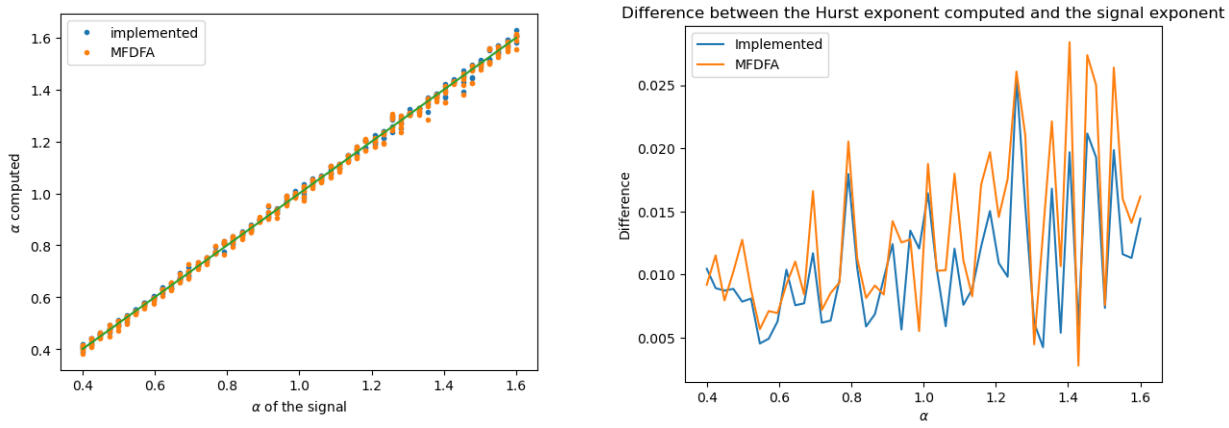


Figure 2.5: This figure presents the results obtained from performing DFA on signals generated with the `stochastic` package of size randomly selected between [10.000, 200.000]. The left graph displays all the computed α^* values along with their exact values. For each α coefficient value, five signals were generated, and the computed α^* values were subtracted from the actual value. Then, the absolute values were averaged together. The right graph shows the evolution of these mean values.

Implementation and scale choice For the following analyses, the second method with smaller scales (around $1/10^{th}$ of the signal length for the maximum scale) will be used as it was the most accurate. The other reason why this method will be favored is that it is much faster.

2.1.2 Method 2: Power Spectrum Analysis

The power spectrum of a time series describes the distribution of power into the frequency components of that signal. In the power spectrum analysis, this power spectrum is then approximated by the function $P(f) \sim f^{-\beta}$. The value β is the one that will help identifying the type of the signal (see Figure 1.3):

- white noise if $\beta \approx 0$,
- 1/f noise if $\beta \approx 1$,
- Brownian noise if $\beta \approx 2$.

2.1.2.1 Implementing and Assessing the method

Implementing the method The value was determined by creating a log-log plot of the power spectrum of the signal and performing a linear regression. The slope of the resulting line corresponds to the β coefficient.

Assessing the method The power spectrum analysis implementation was assessed in the same way as for the DFA method (see section 2.1.1.1 for more details), i.e. a signal with a given value β was generated and the estimation β^* was given by the power spectrum analysis. The accuracy can be deduced from these results.

As can be seen on Figure 2.6, the power spectrum analysis remains quite accurate but is not as good as the DFA method, especially when the β coefficient of the signal is close to 1.

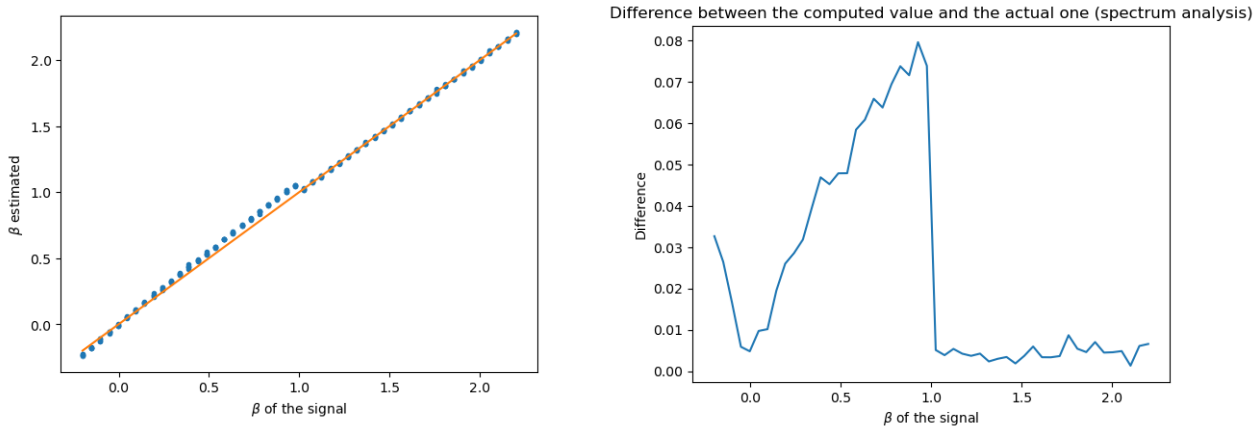


Figure 2.6: This figure presents the results obtained from performing the power spectrum analysis on signals generated with the `stochastic` package of size randomly selected between [10.000, 200.000]. The graph on the left represents all the β^* values computed along with the exact values of β. For each β value, the 5 computed values for each signal were subtracted from the actual value (use of the absolute difference) and were then averaged together. The graph on the right shows the evolution of these means.

2.1.3 Method Selection

After comparing both methods, it was determined that the DFA method was given preference over the power spectrum analysis in this work. DFA demonstrated higher precision, especially for signals with $\alpha = \beta = 1$, and is considered more suitable for fractal pattern analysis in general [19].

2.2 Identify the Fractal Pattern on Heart Rate Signals

2.2.1 DFA on Different Age Groups

DFA was first conducted on the dataset described in section 2.2.1 with the aim of detecting any discernible differences in the fractal pattern based on the age of the subject. However, it turns out that this dataset provides too small signals (with a length ranging from 374 to 5240 beats) leading to an inaccurate analysis that does not yield any significant findings. As depicted in Figure 2.7, the α coefficient is often much smaller than the expected value of 1. Additionally, no significant aging-related pattern can be detected.

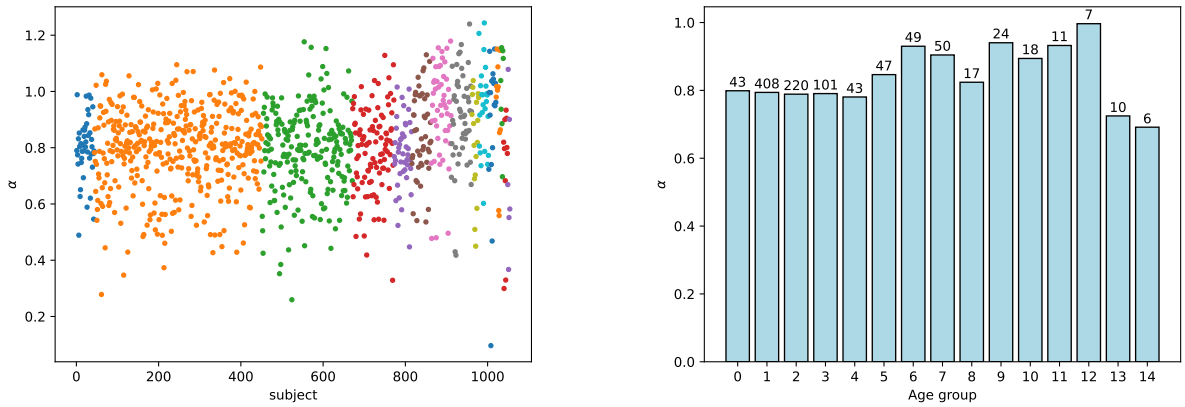


Figure 2.7: The plot on the left displays the α coefficient obtained with DFA of each subject of the first dataset (see section 2.2.1), with dots of the same color indicating membership in the same age group, ordered from youngest to oldest. The youngest are on the left and the oldest are on the right. The scales taken into account ranged from 5 to $l/8$, where l is the length of the time series. The plot on the right shows the average of the α coefficients for each age group. The numbers above the bars represent the count of subjects in each group.

2.2.2 DFA on the Heart Rate of Healthy and Sick Subjects

Let's now use the dataset described in section 1.3.2. Several analyses were conducted on the time series by considering the use of the entire time series or only a portion of it.

Analysis on the whole time series The first analyses are DFA on the whole time series. These analyses are inspired from the ones made by *Peng et al. (1995)* [33] which performed DFA on healthy patients and on patients with congestive heart failure.

When DFA was performed on the whole time series, different scales were considered:

1. between 10 and 10.000,
2. between 10 and 100,
3. between 100 and 10.000.

The findings are presented in Figure 2.8 and Figure 2.9. These figures reveal several important observations. First, for healthy subjects, the results remain consistently close to the significant value of 1 across all time scales, as depicted by the straight line in Figure 2.8. On the other hand, for patients with atrial fibrillation, the time series appears to be completely random at low scales, indicating a lack of correlation between heartbeats [13]. This is represented by an α coefficient close to 0.5. This observation suggests the possibility of utilizing it as a potential diagnostic tool. However, as the scales increase, the α coefficient rises and approaches the value 1, even surpassing the values for healthy subjects. This is clearly observable in Figure 2.8 with the curve having a low slope for small scales

but a higher one as scales grow. Finally, for congestive heart failure patients, although the values are relatively constant, they exhibit slightly more variability than healthy subjects across different time scales. The values are also higher than those for healthy subjects, indicating that the time series becomes more similar to Brownian noise.

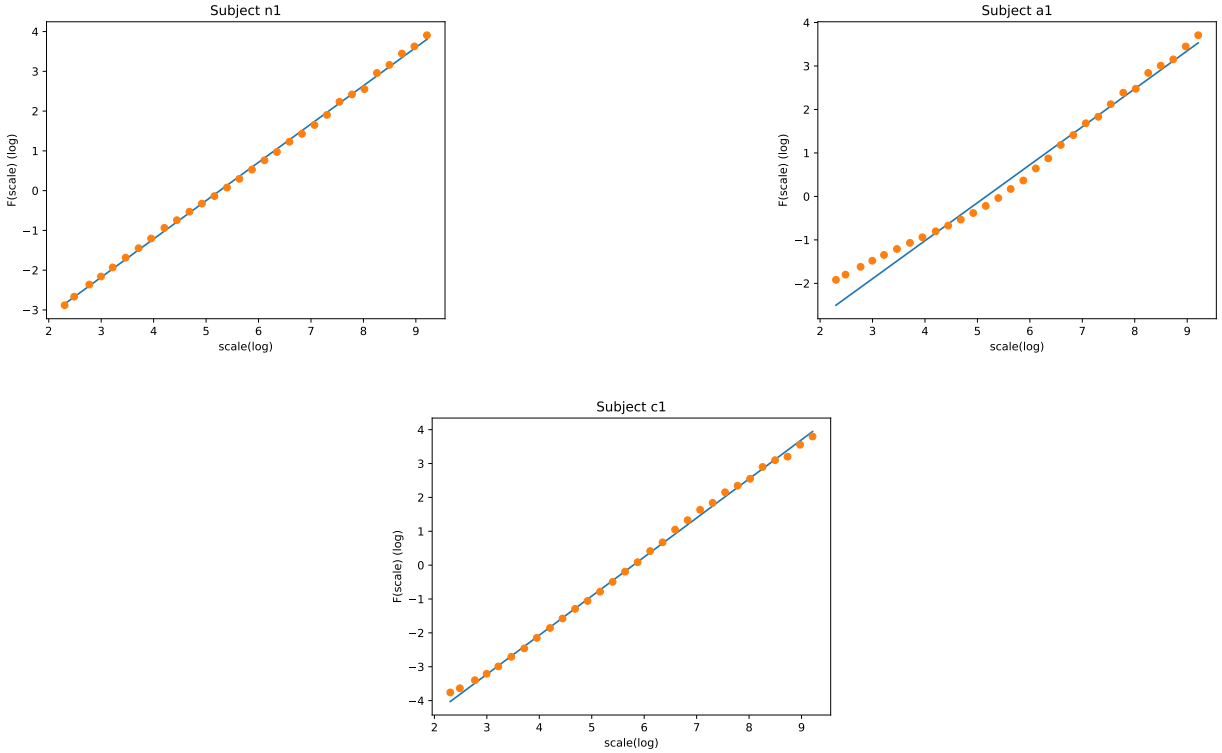


Figure 2.8: The plot above shows the DFA curve for the first subjects in each category. The data for group n (healthy) is represented on the top left side, group a (atrial fibrillation) on the top right side, and group c (congestive heart failure) on the bottom. The scales range from 10 to 10.000. The plots for all subjects are available in appendix B.

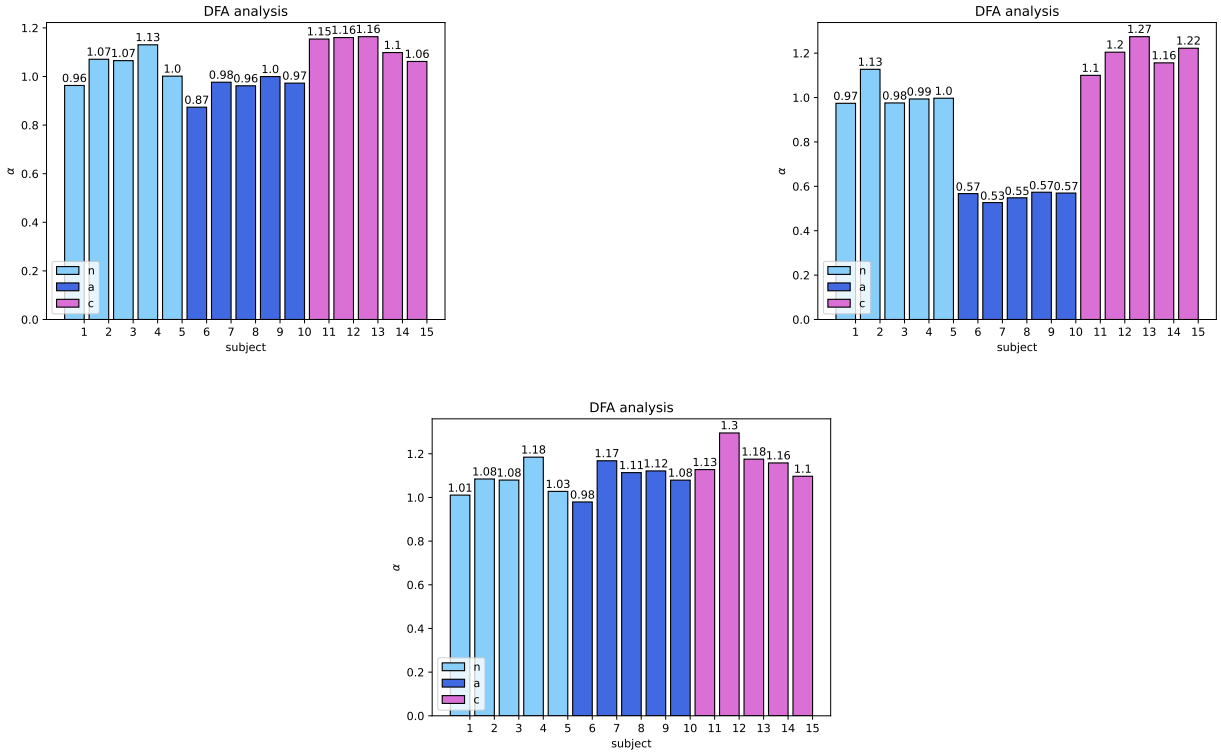


Figure 2.9: The bar charts display the α coefficient resulting from the DFA method performed on the entire time series of each subject. The first five bars represent group n (healthy), the next five represent subjects from group a (atrial fibrillation), and the last five represent those from group c (congestive heart failure). The results for three different scaling ranges are shown: the top left corresponds to scales between 10 and 10.000, the top right to scales between 10 and 100, and the bottom to scales between 100 and 10.000.

Analysis on 8192-beat sub-samples A second analysis was made based on the analysis of *Peng et al.* As pointed in that article, it isn't very practical to diagnose someone based on a 24 hours signal. Thus, DFA was conducted on sub-signals of the original one made of 8192 beats which corresponds to around 2 hours. Two different sets of scales were considered:

1. scales between 4 and 16 beats, the resulting value will be called α_1,
2. scales between 16 and 64 beats, the resulting value will be called α_2.

The results are available in table 2.1. It can be seen that the results for healthy people obtained are extremely close to the ones obtained in *Peng et al.*. On the other hand, for patients with congestive heart failure, even though it is quite close, there are still some small differences between our results and those from *Peng et al.*. We can make the assumption that this is due to a variation in severity of the disease between the subjects of the two datasets, but since we don't have access to the one used in *Peng et al.*, we have no way to confirm this assumption.

On Figure 2.10, these intervals are plotted. We can clearly distinguish the subjects with atrial fibrillation from the rest using α_2. However, due to the intervals overlapping with each other, there is no way to distinguish healthy subjects from the ones with congestive heart failure.

	Healthy	Congestive heart failure	Atrial fibrillation
α_1	1.2103 ± 0.1754	0.9969 ± 0.2854	0.6039 ± 0.0492
α_1 in *Peng et al.*	1.201 ± 0.178	0.803 ± 0.259	/
α_2	0.9915 ± 0.1167	1.2254 ± 0.119	0.5453 ± 0.0393
α_2 in *Peng et al.*	0.998 ± 0.124	1.125 ± 0.216	/

Table 2.1: Results of DFA made on time series made of 8192 beats. These time series come from the second dataset (section 1.3.2) and a division into sub-signals of the correct size of the provided signals. The results are presented as *mean± standard deviation.*

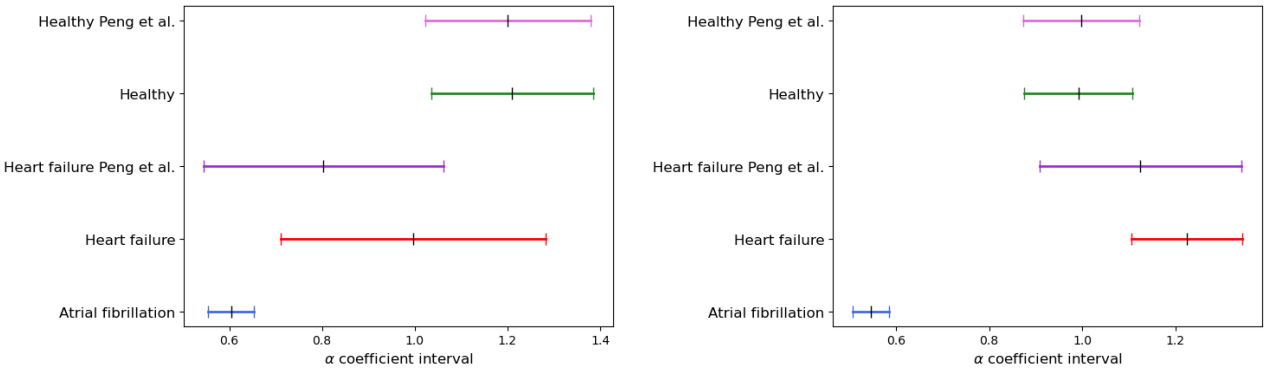


Figure 2.10: Intervals of the α coefficient obtained from DFA for healthy and diseased patients. The intervals on the left correspond to α_1, and those on the right correspond to α_2.

Analysis on a per hour basis A final analysis was performed by looking at the evolution of the α coefficient in time. It was conducted on a per hour basis. For that purpose, the time series were divided into signals that correspond to an hour long and DFA was performed on each of these sub-signals. The results are available on Figure 2.11. The first observation is that no trend can be noticed with the evolution of time of the α coefficient. This means that DFA won't be affected by the time at which the cardiac signal will be recorded. The second observation is that the results confirm what was said earlier: the subjects with atrial fibrillation can clearly be distinguished from the other subjects but healthy subjects and patients with congestive heart failure can't be significantly differentiated based on DFA.

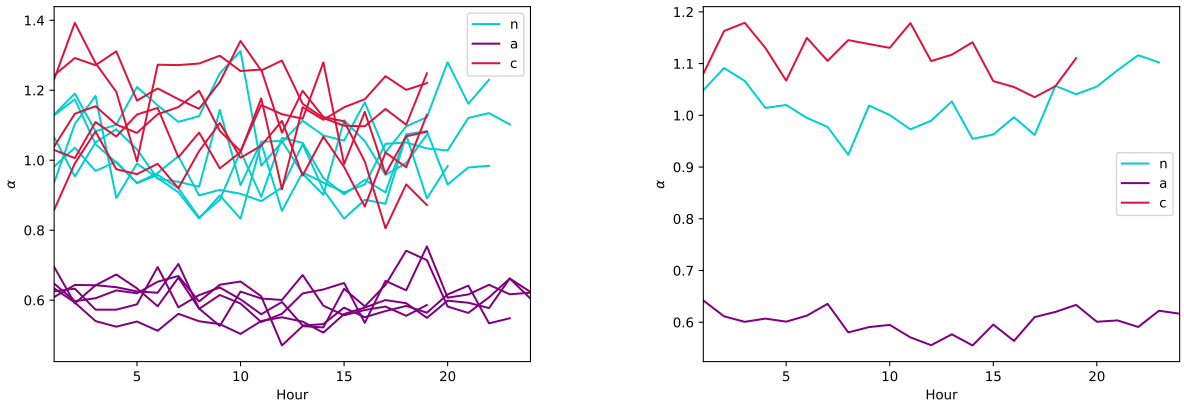


Figure 2.11: The plot shows the hourly evolution of the α coefficient from the DFA method for each subject in the second dataset (refer to section 1.3.2) on the left and the mean of each group (n for healthy, a for atrial fibrillation and c for congestive heart failure) on the right.

2.3 Conclusion

The results obtained with the DFA method have demonstrated its potential as a diagnostic tool for identifying atrial fibrillation. However, the same cannot be said for congestive heart failure, as the DFA method has shown mixed results in differentiating between patients with this condition and healthy individuals.

The Detrended Fluctuation Analysis (DFA) method is a reliable and accurate tool for analyzing fractal patterns in the heart rate signal (section 2.1.1.1). However, the size of the signal is a critical factor for obtaining significant results with DFA, and it is important to ensure that the signal is of sufficient length before conducting the analysis (section 2.2.1).

Finally, the results also suggest that the time at which the heart rate signal was measured does not appear to have a significant impact on the fractal pattern detected by the DFA method. This implies that the fractal properties of the heart rate signal may be a stable feature of the cardiovascular system, even in the presence of disease (section 2.2.2).

Chapter 3

The Ivanov Model: a Stochastic Model for Heart Rate Fractal Dynamics

3.1 Model Description and Analysis

3.1.1 Model Overview

This chapter focuses on a stochastic model proposed by *Ivanov et al.* [21] to describe the heart rate variability of healthy individuals. We examine this model to gain insights into its mechanism for capturing the fractal dynamics of the heart rate. A good understanding of this model is necessary because we attempt to fit some of its parameters to the subjects of the second dataset (see section 1.3.2) to determine if the model can be extended to diseased subjects and to identify any differences between healthy and sick patients.

The Ivanov model [21] is a stochastic model which considers that the heart rate is influenced by 3 elements (more details in section 1.2.2):

- the sinoatrial node (SA) which tends to keep the heart rate at a constant rate,
- the sympathetic systems (SS) which speeds up the heart rate,
- the parasympathetic system (PS) which slows down the heart rate.

The idea is that the time interval between two heartbeats, denoted as $\tau(n)$, behaves like a random walk process. Different factors influence this process, and each factor has a specific preferred level, denoted as τ_k. The process tends to move towards these preferred levels. However, the levels themselves change over time. To illustrate this idea, a simplified representation of the model's behavior is shown in Figure 3.1.

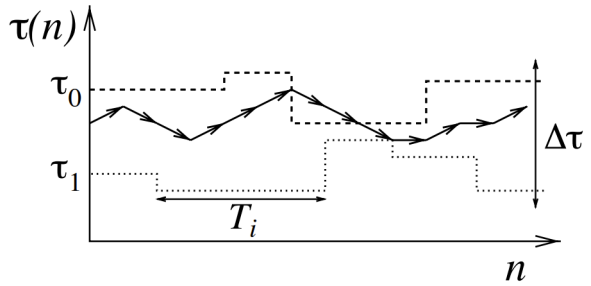


Figure 3.1: Schematic representation of the dynamics of the Ivanov model. This model consists in a random walk with stochastic feedback controls. In this specific representation, only two preferred levels are used: τ_0 and τ_1. *Figure from ref [21].*

In this model, it considers that the time $\tau(n+1)$ between the heart beats n and $n+1$ can be computed in the following way:

$$\tau(n+1) - \tau(n) = I_{SA}(n) + I_{PS}(n, \tau_{PS}(n)) + \sum_{j=1}^{N} I_{SS}^{j}(n, \tau_{SS}^{j}(n)) \tag{3.1}$$

where

- $$I_k(n) = \begin{cases} w_k(1+\eta) & \text{if } \tau(n) < \tau_k \\ -w_k(1+\eta) & \text{otherwise,} \end{cases} \tag{3.2}$$

  with I_k being the feedback input biasing the walker to a preferred level for a given system (SA, SS or PS),

- τ_k is the preferred level for $\tau(n)$ and remains at the same value for a certain amount of time which follows a probability distribution with a mean of T_{lock}^{k},
- w_k is the strength of the feedback input I_k,
- η is white noise.

3.1.2 Implementing and Assessing the Model

Implementation For the probability distribution of the white noise η, the preferred levels and the time of an unchanged preferred level T, the same distributions than the ones used in the paper [21] were used: a Laplace distribution for the first, a uniform distribution for the second and a normal distribution for the latest. The method to generate a signal takes as parameters:

1. the input strengths w_{SA}, w_{SS}, w_{PS},
2. the number of sympathetic inputs N,
3. the preferred level τ_{SA},
4. the upper and lower bounds on τ_{PS} and τ_{SS},
5. the mean and standard deviation of the white noise η (note that the mean should be 0 as it is white noise),
6. the mean and standard deviation of the time T,
7. the number of beats to be generated.

Assessing An example of a signal generated by this model is available in Figure 3.2. It can be seen that the signal indeed presents a fractal pattern as the signal has a similar pattern at different scales. Furthermore, the generated signal is similar to a signal of a healthy subject as can be seen on Figure 3.3.

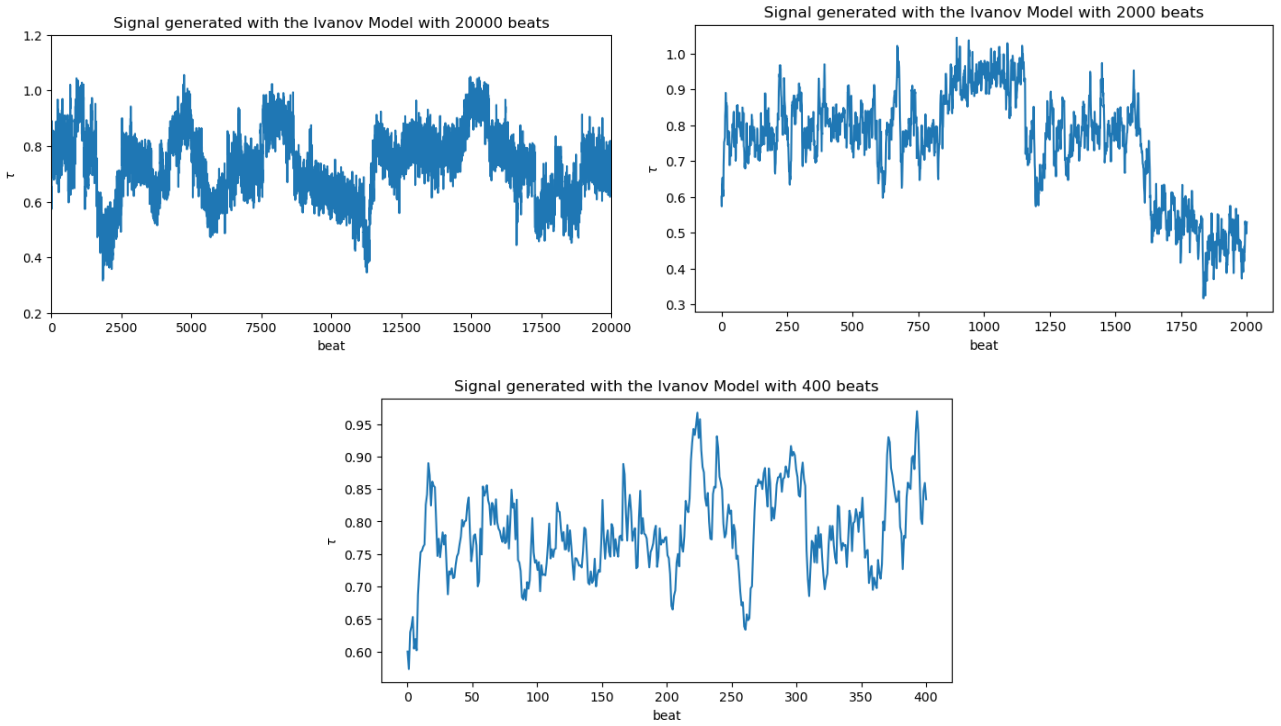


Figure 3.2: Signal generated from the Ivanov model at various time scales. The signal was generated with the same variable values as in the paper, i.e $N = 7$, $w_{SA} = w_{SS} = w_{PS}/3 = 0.01s$. η is a symmetrical exponential distribution (Laplace distribution) with 0 mean and 0.5 scale. $\tau_{SA} = 0.6s$, and both τ_{SS}^{j} and τ_{PS} are drawn from a uniform distribution, the first being bound by $[0.2, 1.0]$ and the latter by $[0.9, 1.5]$. Finally, T follows a normal distribution with a mean of 1000 beats.

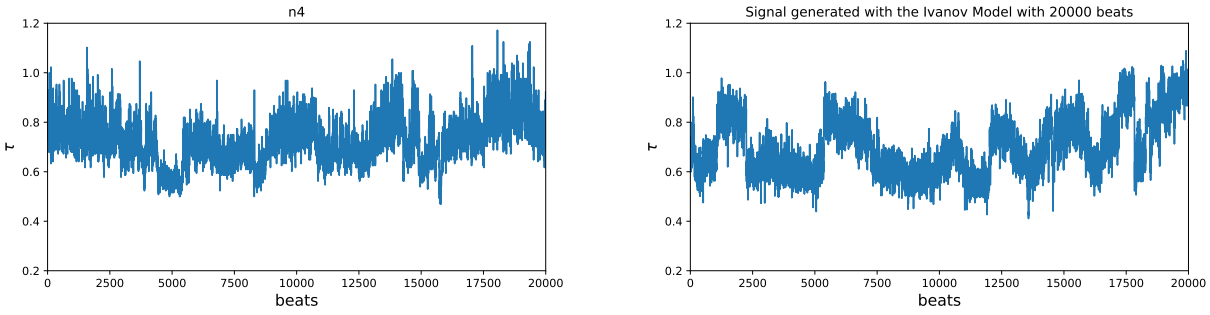


Figure 3.3: On the left, 20.000 beats of the signal of the forth healthy subject in the second dataset (see section 1.3.2). On the right, a signal generated with the Ivanov model. The parameters value are the same as in Figure 3.2.

However, this isn't enough to assess the model. Thus a Power spectrum analysis along with a DFA (see section 2.1) were performed on 1000 signals generated with that model. As can be seen on Figure 3.4, the signals generated from the Ivanov model are indeed fractals. A mean value of $\alpha \simeq 1.06006$ was found with the DFA algorithm while $\beta \simeq 1.156$ for the power spectrum analysis. In *Ivanov et al.* paper [21], the value for β was ~ 1.1 which is quite similar to our results.

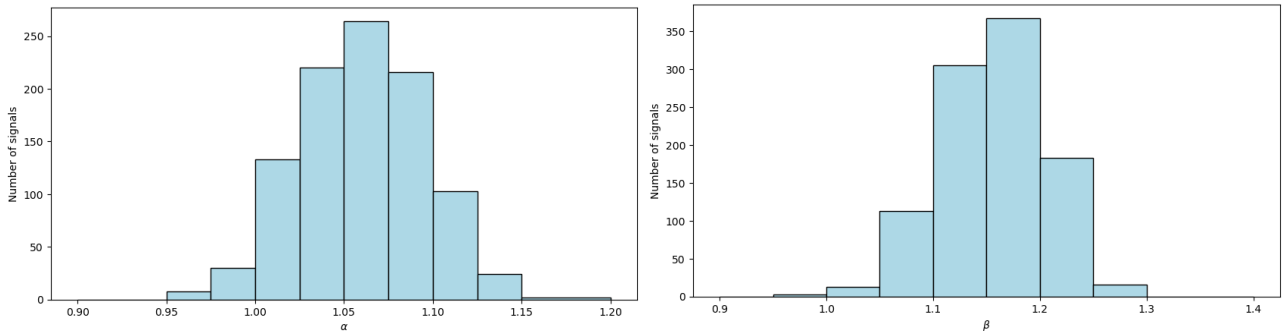


Figure 3.4: Histogram of the values of α for DFA (scales between 5 and 500) and β for the spectrum analysis on 1000 signals generated with the Ivanov model. The parameters value are the same as in figure 3.2.

3.1.3 Selection of Default Parameter Values

The default parameter values, which are the same as those used in section 3.1.2, are as follows:

1. $w_{SA} = 0.01$,
2. $w_{SS} = 0.01$,
3. $w_{PS} = 0.03$,
4. $N = 7$,
5. $\tau_{SA} = 0.6$,
6. $\tau_{SS} \in [0.2, 1]$,
7. $\tau_{PS} \in [0.9, 1.5]$,
8. η has a standard deviation of $0.5 \times \sqrt{2}$,
9. T has a mean of 1000 and a standard deviation of 500.

Most of these parameters have the same values as in the paper [21]. However two parameters are defined in a different way:

1. The first one is the standard deviation of the time T. The reason is simply that that standard deviation wasn't specified in the paper. However, as explained in section 3.1.5.5, this parameter doesn't have much importance.

2. On the other hand, the standard deviation of the white noise η was initially set to 0.5. However, after generating a signal with the Ivanov model, it turned out that the signals didn't look as smooth as the ones generated with a scale of 0.5 (with the Laplace distribution, the standard deviation is equal to $b \times \sqrt{2}$ with b being the scale) and looked more like a signal made of "blocks", thus not looking quite like a healthy patient heart rate signal (see Figure 3.5).

   On top of this shape less "smooth", the fractal pattern is also lost. Indeed, 1000 signals were generated with a standard deviation of 0.5 and DFA was performed on these signals. The obtained mean was $\alpha \sim 1.18279$ which is quite far from the required value of 1 (see Figure 3.6).

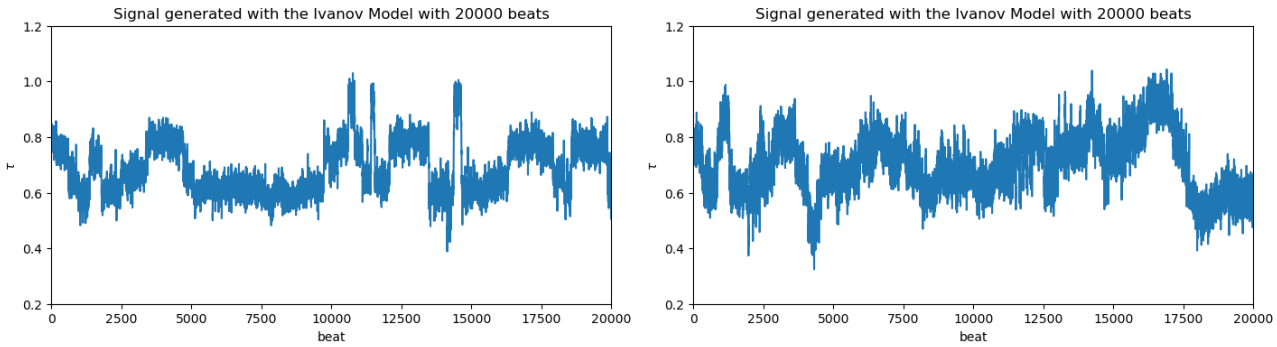


Figure 3.5: Generated signal with the Ivanov model. The parameters are the ones by default but the standard deviation of η is 0.5 on the left and $0.5 \times \sqrt{2}$ on the right.

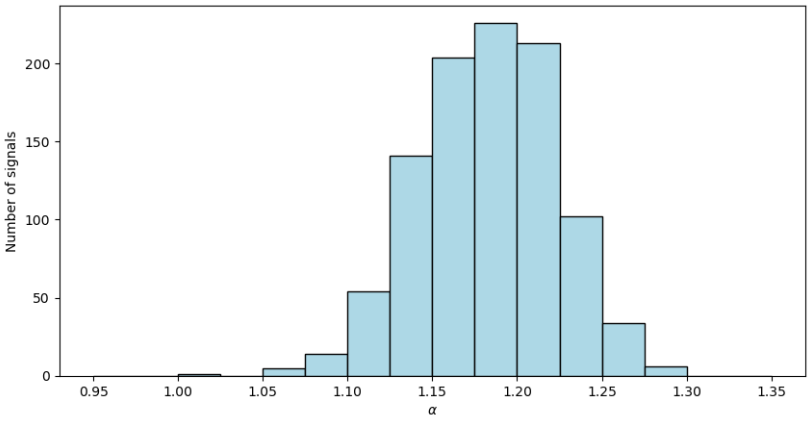


Figure 3.6: Histogram of the values of α for DFA (scales between 5 and 500) for 1000 signals generated with the Ivanov model. The parameters are the ones by default except for the standard deviation of the white noise η which is 0.5.

3.1.4 Mechanisms Underlying Fractal Pattern in the Model

The fractal patterns found in some biological signals may be explained by the influence of different inputs that change at varying time scales, as discussed in previous studies [14, 23]. *Haudorff et al.* demonstrated that the combination of random processes with different time scales can produce the 1/f noise pattern. This model shares similarities with the present study, as it involves the combination of various inputs that change over time and have their own preferred levels.

Furthermore, if we have a look at the various statistical characteristics of the 1/f noise (section 1.2.1.1), this model respects them:

- **Randomness:** there is randomness mainly with the parameter η to which we also add the time T and the preferred levels τ_k;
- **Stationarity:** the generated time series aren't stationary since the statistical properties (especially the mean of the signal) will change with the preferred levels τ_k;
- **Persistence:** the model indeed has long-lasting fluctuations that persist. In this model, the time in which all the τ_k remain the same is the time during which the signal is in a same state;
- **Correlation/Memory:** each $\tau(n)$ depends on the value of $\tau(n-1)$ which itself depends on $\tau(n-2)$, ...

However, this model works as long as the parameters remain in a certain range of values. In the next section, we have a look at how each parameter can influence the fractal pattern of the model.

3.1.5 Parameter Effects on Fractal Pattern in the Model

3.1.5.1 Strength of the Feedback Input I_k

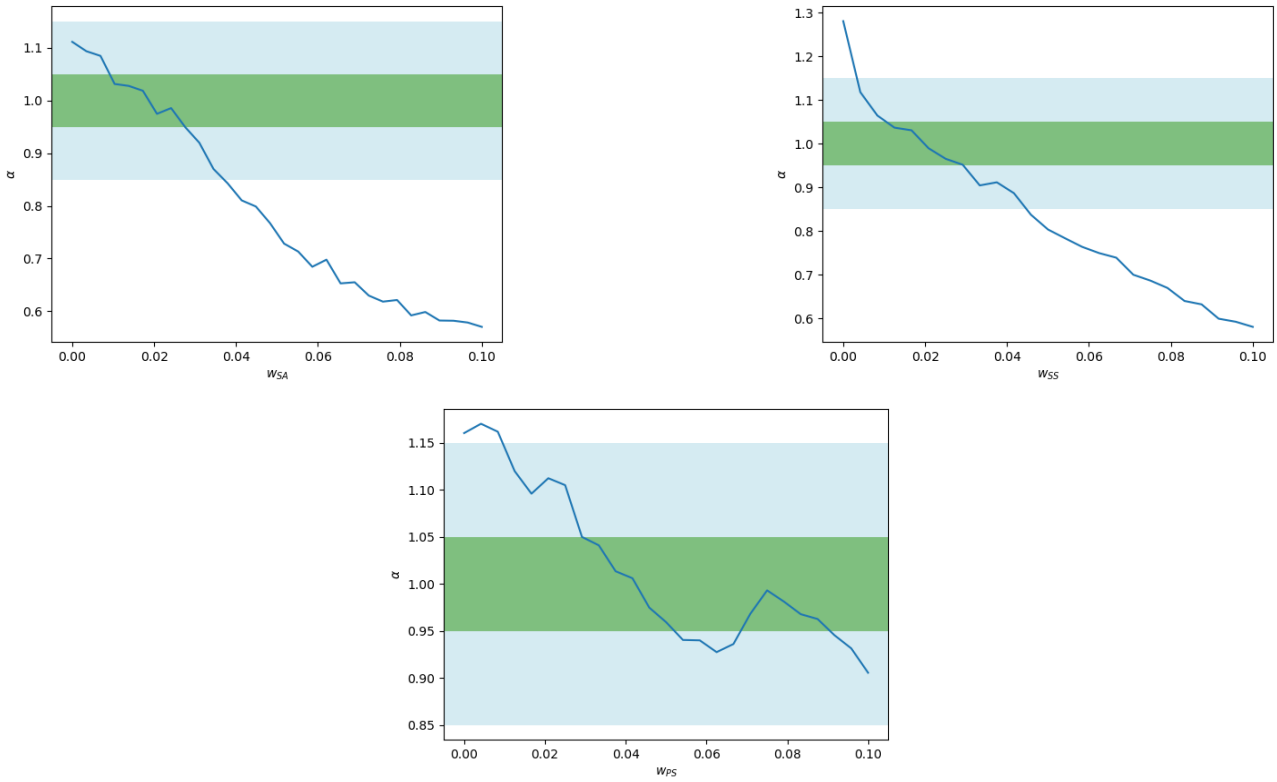


Figure 3.7: Evolution of the α coefficient obtained with DFA for the value of parameters w_{SA} (top left), w_{SS} (top right) and w_{PS} (bottom). The lags considered were between 5 and 500 and the generated signals were each of 20.000 points. The values considered for the parameters are between 0 and 0.1. 50 different parameter values were considered and 10 trials were done for each value. The mean of these 10 trials is what is plotted. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

On Figure 3.7, it can be seen that the three parameters tend to act in the same way. However, there are differences in their impact.

As the parameter values increase, the α coefficient approaches 0.5, making the signal more similar to white noise. This is because, as the strength of the input increases, so does the value of I_k (see Equation 3.2). For the SA node, at some point, I_{SA} becomes much larger, causing the signal to vary only around the straight line of τ_{SA}. A similar effect occurs with w_{SS}, where the values of I_{SS} become so large that changes in τ^j_{SS} are barely noticeable compared to the signal variation. When this happens, the signal loses its non-stationarity property and appears more like white noise.

On the other hand, w_{PS} does not seem to significantly affect the fractal pattern, as the larger values of I_{SA} and I_{SS} compensate for the increased input strength. We can however see that the α values do decrease and with higher values considered, we will see the same curve as for w_{SA} and w_{SS}.

Finally, when the value of w_{SS} is too small, the α coefficient increases, indicating that the signal is approaching Brownian noise. This is comparable to having a small value for N, and this is further explained in the following section.

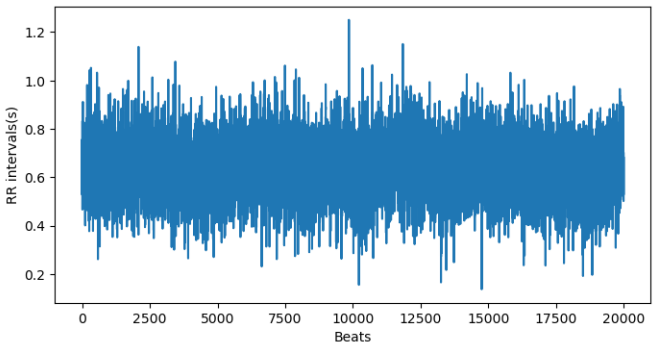


Figure 3.8: Signal generated with the Ivanov model with $w_{SA} = 0.1$

3.1.5.2 Number of Sympathetic Systems N

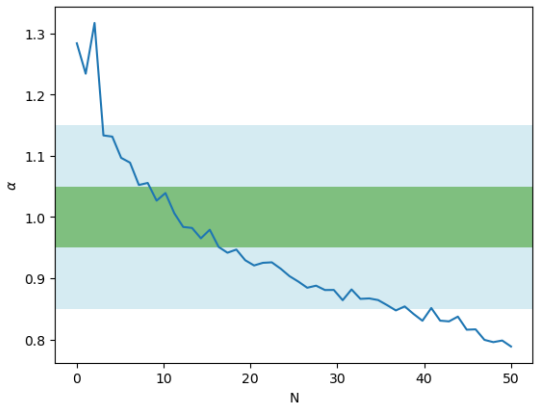


Figure 3.9: Evolution of the α coefficient obtained with DFA for the parameter N. The lags considered were between 5 and 500 and the generated signals were each of 20.000 points. The values considered for the parameter are between 0 and 50. 50 different parameter values were considered and 10 trials were done for each value. The mean of these 10 trials is what is plotted. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

Figure 3.9 illustrates that having a small value for N results in signals that are closer to Brownian noise. This is due to the fact that I_{PS} will account for most of the variation in $\tau(n+1)$. As a result, the signal will be made of "blocks" where the mean of the signal only varies with changes in τ_{PS}, leading to drifts in the signal (as depicted in Figure 3.10). The value N thus needs to be large enough to compensate I_{PS}.

On the other hand, as long as the value for N doesn't get too large, the model is not too impacted by the value of N. This is most likely because the I_{SS}^j with higher values for τ_{SS}^j will compensate the ones with lower values for τ_{SS}^j, and thus, there won't be very large variations in $\tau(n)$ as it was observed with the strengths of the feedback inputs.

The signal tends to white noise with very large values of N because when there are too many I_{SS}^j inputs, a change in one preferred level will have little effect on the final value. As a result, the signal becomes stationary (see Figure 3.10).

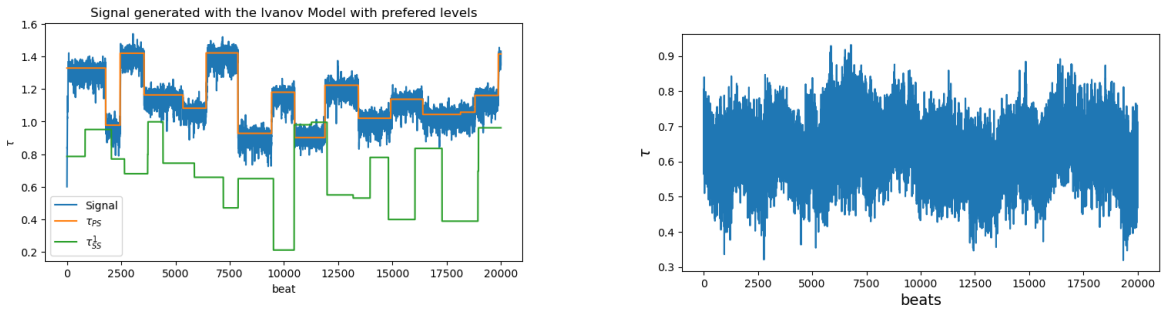


Figure 3.10: Generated signals with the Ivanov model. On the left, the signal is generated with $N = 1$ and the preferred levels τ_{PS} and τ^1_{SS} are also plotted. On the right, the signal is generated with $N = 50$.

3.1.5.3 White Noise η

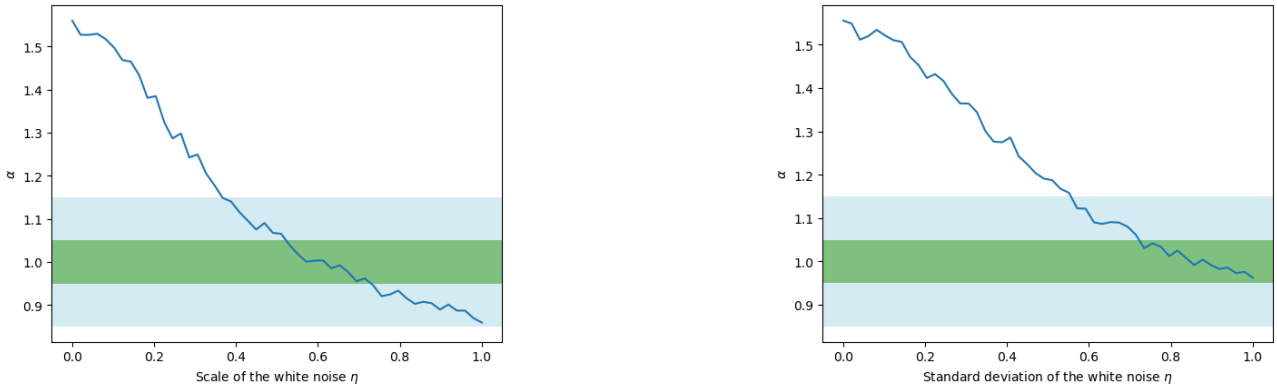


Figure 3.11: Evolution of the α coefficient obtained with DFA for the scale (left) and the standard deviation (right) of the white noise η. The lags considered were between 5 and 500 and the generated signals were each of 20.000 points. The values considered for the parameters are between 0 and 1. 50 different parameter values were considered and 10 trials were done for each value. The mean of these 10 trials is what is plotted. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

Figure 3.11 demonstrates that when the scale of the parameter η is too low, the resulting signal tends towards Brownian noise. This is because when the scale is very small, η approaches zero, which means that the noise has no effect on the signal. In this case, the signal will primarily consist of drifts, which leads to a signal that is closer to Brownian noise.

On the other hand, when the scale grows, larger values will be observed for η, thus taking more importance in the Ivanov equation. This can be seen in the decreasing values of α with higher scale values.

Another observation from Figure 3.11 is that using a scale of 0.5 instead of the standard deviation is appropriate, as *Ivanov et al.* [21] mention in their paper that the values should fall between 0.4 and 0.6, which is generally the case for the α coefficient when using the scale, but not when using the standard deviation.

The mean of η isn't considered here as by definition, white noise has a 0 mean.

3.1.5.4 Bounds for the Preferred Levels τ_k

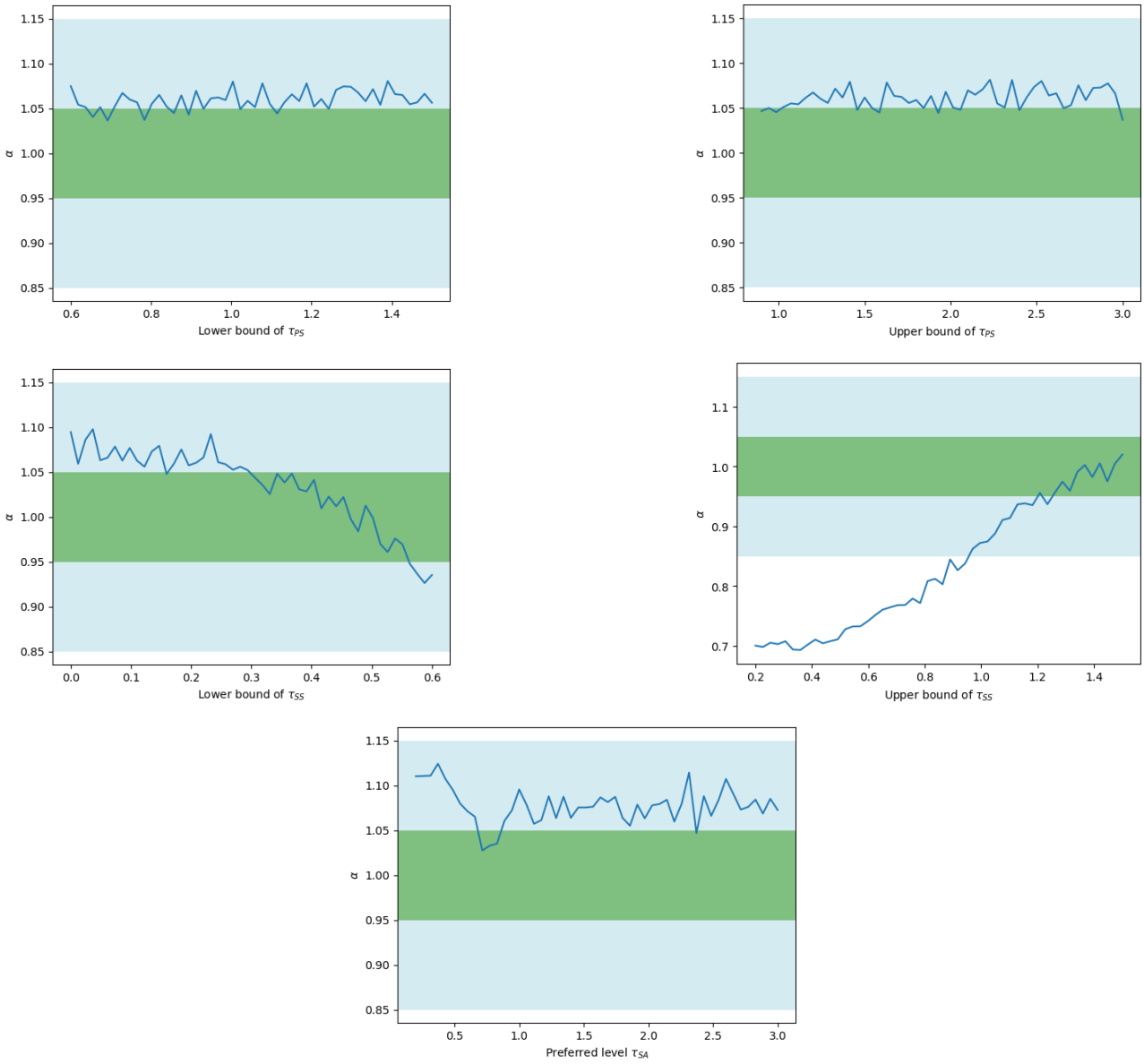


Figure 3.12: Evolution of the α coefficient obtained with DFA for the parameters τ_{PS} (on the top, lower bound on the left and upper bound on the right), τ_{SS} (in the middle, lower bound on the left and upper bound on the right) and τ_{SA} (bottom). The considered lags were between 5 and 500 and the generated signals were each of 20.000 points. The range in parameters values must make sense biologically and each level will have its own bounds. For τ_{PS}, the lower bound must not be lower than the value of τ_{SA} and must obviously remain under the upper bound, thus the values were considered between [0.6, 1.5]. The upper bound must not be too long as two heart beats can't be separated for more than 10 seconds normally. The considered values were between [0.9, 3]. For τ_{SS}, the lower bound must not be higher than τ_{SA} (values considered between [0, 0.6]) and the upper bound must not be larger than the upper bound of τ_{PS} (values considered in the interval [0.2, 1.5]). Finally, for τ_{SA}, the considered values were in the interval [0.2, 3]. 50 different parameter values were considered and 10 trials were done for each value. The mean of these 10 trials is what is plotted. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

The results on Figure 3.12 show that the bounds of the preferred levels seem to not have much impact apart from the upper bound of τ_{SS}.

In a general way, the value of the bounds will not impact the fractal pattern much. However it has an impact on the value of $\tau(n)$. For example, if we have two different subjects that have a different heart rate average, then the preferred levels in the model will need to be adjusted to account for these differences.

The reason why the upper bound of τ_{SS} has an influence is because it is too close to the lower bound, leading to τ_{SS}^{j} evolving more as a straight line. In that case, we fall back to cases similar to the ones in section 3.1.5.1. The reason why this wasn't observed with the lower bound is that we didn't consider values close enough to the chosen upper bound in order to remain biologically consistent, but we can see that the curve was already getting down.

With this logic, we could argue that this phenomena wasn't observed with τ_{PS}. There are several reasons why the bounds have little impacts. The first one is that most of the final value of Equation 3.1 will greatly depend on the I_{SS} value since it has more importance (7 inputs against one for PS) even if $w_{PS} > w_{SS}$. This is for that same reason that the value for τ_{SA} hasn't much impact on the fractal pattern.

A second reason, that concerns more the upper bound than the lower bound, would be that in a general way, $\tau(n)$ never reaches the bound of τ_{PS}, thus whatever the bound is, we will mostly be in the first case of Equation 3.2 (see Figure 3.13).

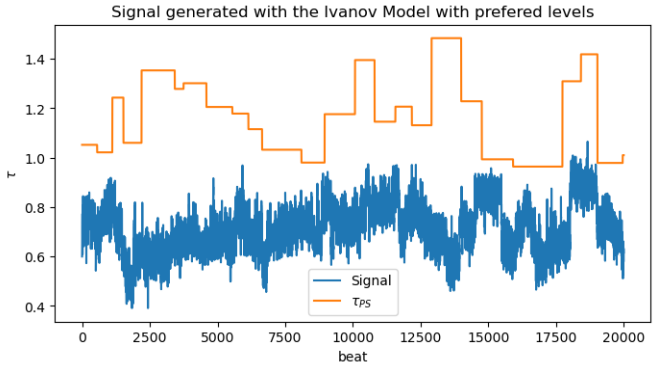


Figure 3.13: Generated signal with the Ivanov model along with the evolution of τ_{PS}

3.1.5.5 Time for the Preferred Levels T

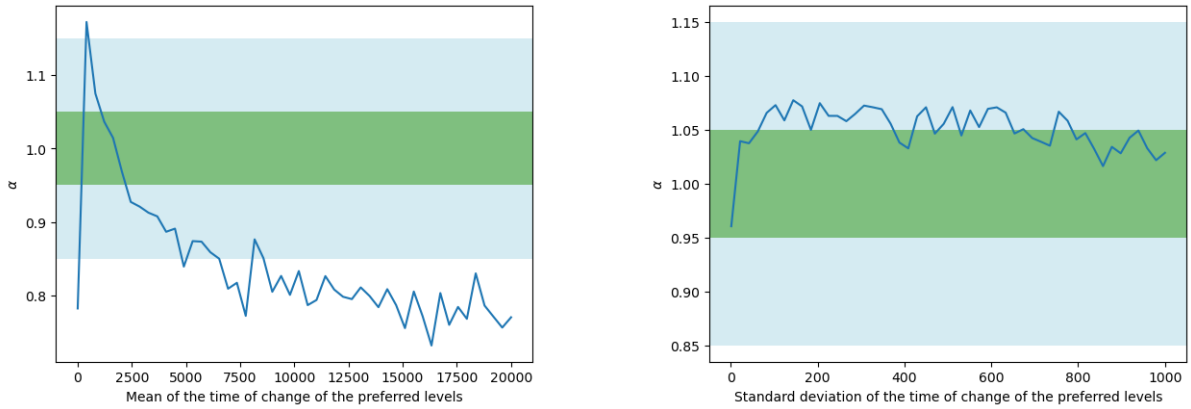


Figure 3.14: Evolution of the α coefficient obtained with DFA for the time T for which the preferred levels τ_k remain the same (evolution of the mean on the left and evolution of the standard deviation on the right). The values of the mean were considered between 0 and the length of the signal. The standard deviation was considered between 0 and the mean of T. The lags were between 5 and 500. The generated signals were each of 20.000 points. 50 different parameter values were considered. Ten trials were done for each value. The mean of these ten trials is what is plotted. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

From the results on Figure 3.14, it can be seen that the time T has an influence on the fractal pattern but only the mean of T has an impact, while its standard deviation has none.

The curve of the mean of T is different from the ones that were previously observed. Here, if the value is too small, the signal will be close to white noise but it will also happen if the value is too large. Let's first consider the case where the mean value is too small. In that case, the correlation between the points will be completely lost. Indeed, the I_k value will no longer depend on $\tau(n)$ but on the value of the preferred levels τ_k that will vary at almost each time step. This means that the signal will no longer try to remain to that preferred level since that level is completely random and changes at each time step. On the opposite case, if the value becomes too large, that means that the preferred levels will never change. We then reach cases where the preferred levels are straight lines which means that the non-stationarity property is lost.

The standard deviation of T on the other hand doesn't have any impact on the fractal pattern. However, larger values for the standard deviation are to be preferred as too small standard deviation will lead to the preferred levels changing at the same time. The generated signal will more look like "blocks", while a higher value of the standard deviation leads to "smoother" variations (see Figure 3.15).

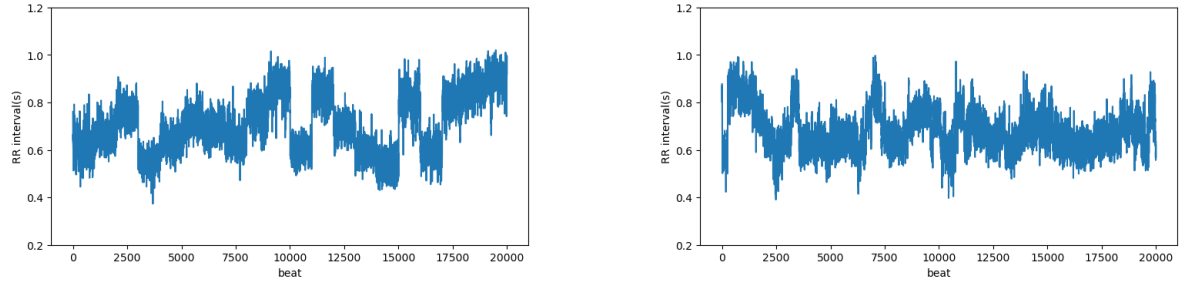


Figure 3.15: Signal generated with the Ivanov model with a standard deviation for T of 1 (on the left) and 1000 (on the right)

3.1.5.6 Parameters Impact Summary and Selection for Fitting

Parameters	Impact on the fractal pattern	Other impact
Strengths of the feedback inputs w_{SS}, w_{PS} and w_{SA}	Important impact when the value grows, especially for w_{SA} and w_{SS} leading to a signal close to white noise. Impact more negligible for w_{PS}. For w_{SS}, a low value leads to a signal made of "blocks".	Impact on the amplitude of the signal as well as its standard deviation.
Number of sympathetic signals N	Impact negligible as long as the values are within a reasonable range, as extremely small or large values may have more significant effects.	
White noise η	Important impact when the standard deviation is small as it removes the randomness of the signal.	Important impact on the standard deviation as well as the local amplitude in steady states.
Preferred levels τ_{SA}, τ_{PS} and τ_{SS}	Not an important impact as long as the upper and lower bound, especially the ones for τ_{SS}, are far enough.	Impact on the mean of the signal and the range of the values taken by the time series.
Time for constant preferred levels T	No impact of the standard deviation of T but very important impact of the mean value as extreme values (both too small and too large) will remove the non-stationarity propriety of the signal.	

Table 3.1: Summary of the impact of the various parameters of the Ivanov model

The analysis of table 3.2 led to the conclusion that the feedback input strengths w_{SA}, w_{PS}, and w_{SS} would be the most relevant parameters to adjust depending on the subject. These parameters have a significant impact on both the amplitude and fractal pattern of the signal, making them the most interesting ones to vary.

Let's examine the other parameters and understand why they were not chosen:

- The parameter N, representing the number of sympathetic signals, has a negligible impact, and therefore, it is not necessary to fit it to the heart rate time series.
- The white noise parameter η could also be a suitable option since it affects both the fractal pattern and the amplitude, similarly to the feedback input strengths. However, the impact of the standard deviation of η is only significant when it is small, and it has little to no impact when it is large. This makes the feedback input strengths more interesting since they impact the fractal pattern in a similar way and have a stronger influence on the amplitude.
- As the preferred levels have no significant impact on the fractal pattern, fitting these parameters to the heart rate time series would not be meaningful. However, this does not necessarily mean that these values must remain fixed. Since the impact of the preferred levels is mainly on the specific values taken by the time series, they could potentially be adjusted based on observations of the heart rate signals.
- While the time period T for which the preferred levels remain constant does have an impact on the fractal pattern, we didn't fit it to the heart rate signals. This is because its value plays a crucial role in preserving the persistent property of the model. Therefore, it is important to keep the value of T fixed, and not adjust it during the fitting process. Furthermore, it provides limited information regarding the impact of the systems on the heart rate, which is why the feedback input strengths were given preference.

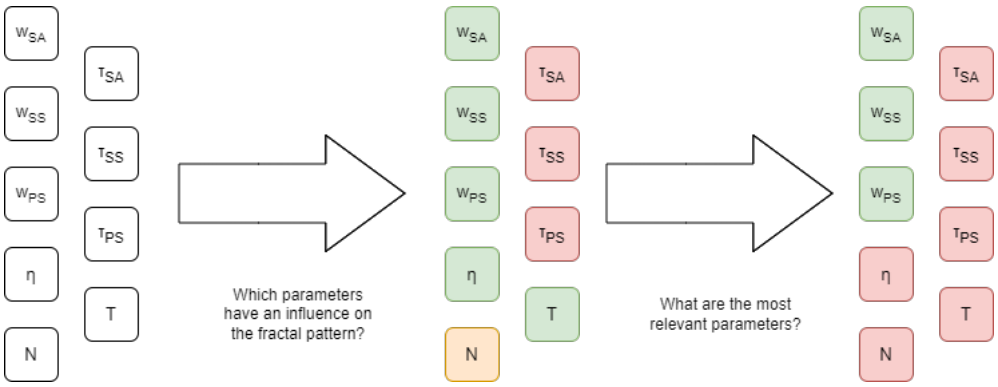


Figure 3.16: Methodology for selecting the most appropriate parameters to be fitted on heart rate signals.

3.1.6 Parameter Estimation on Real Data

With the most suitable parameters identified to be fitted on the heart rate of different subjects, we can now proceed to the actual parameter estimation in the subsequent sections. This allows us to determine the optimal parameter values for each subject. Three distinct methods were employed for this purpose:

1. maximum likelihood estimation (section 3.2),
2. approximate Bayesian computation (section 3.3),
3. Bayesian inference with neural networks (section 3.4).

As explained in section 3.1.5.6, we will focus on fitting the feedback input strengths w_{SA}, w_{PS}, and w_{SS}. The same methodology will be employed for each of the three methods, summarized in in Figure 3.17:

1. To assess the parameter estimation method, we will generate multiple signals using the Ivanov model with known parameter values;
2. Utilizing one of the three methods, we will perform the parameter estimation process;
3. The estimated parameters will be compared to their original values;
4. If the estimated parameters closely match the original values, we can conclude that the method is accurate and can be applied to estimate parameters in heart rate data.

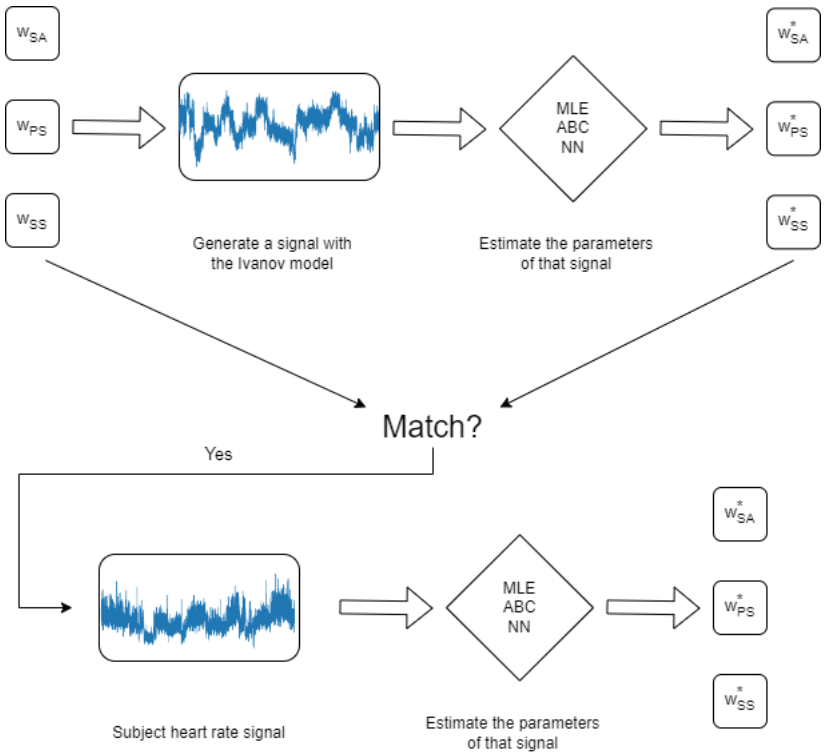


Figure 3.17: Methodology used to estimate the parameters w_{SA}, w_{SS} and w_{PS} of the Ivanov model for heart rate signals.

3.2 First Method for Parameter Estimation: Maximum Likelihood Estimation

3.2.1 Method Overview

The maximum likelihood estimation (MLE) is a method to estimate the optimal parameters of a model using available data. This approach involves selecting parameter values that maximize the likelihood, which is the probability of observing the given data based on the parameter values. Typically, the likelihood function is denoted as $\mathcal{L}(\theta|X)$, where θ represents the parameters of the model and X is the observed data. The likelihood function for n data points can be written as:

$$\mathcal{L}(\theta|X) = f(X = x_1|\theta) \times ... \times f(X = x_n|\theta)$$

where $f(X = x_i|\theta)$ represents the probability of observing the i-th data point x_i given the model parameters θ.

Often, it is more convenient to maximize the log-likelihood rather than the likelihood itself, as it results in a sum rather than a product.

$$\log(\mathcal{L}(\theta|X)) = \log(f(X = x_1|\theta)) + ... + \log(f(X = x_n|\theta)).$$

In the end, the estimation for the parameters θ is given by:

$$\theta^* = max_\theta \log(\mathcal{L}(\theta|X)).$$

3.2.2 Applying MLE to the Ivanov Model: Analytical Form of the Likelihood

To begin, we need to determine the probability distribution of the model. Equation 3.1 shows that we need to maximize the probability that the value $\tau(n+1) - \tau(n)$ is equal to the difference between two points in the time series. This involves determining the probability distribution of I_{SA}, I_{PS}, and I_{SS} individually. Subsequently, the probability distribution for the sum of all three can be computed. Since I_k is dependent on the value $\tau(n)$, we can simplify by maximizing the following probability:

$$f(X|\tau(n), \theta)$$

where $X = \tau(n+1) - \tau(n)$, and θ represents the model parameters.

3.2.2.1 Sinoatrial Node

The probability distribution of I_{SA} can be determined as follows, given that the value of the preferred level τ_{SA} is unique for the SA node:

$$\begin{cases} I_{SA} \sim \mathsf{Laplace}(w_{SA} \times (1+\mu), w_{SA} \times b) & \text{if } \tau(n) < \tau_{SA} \\ I_{SA} \sim \mathsf{Laplace}(-w_{SA} \times (1+\mu), w_{SA} \times b) & \text{otherwise,} \end{cases} \tag{3.3}$$

where μ is the mean of the parameter η in Equation 3.2 and b is its scale. Note that since η is white noise, μ will always be 0, thus for the next probability distributions definition, it will be ignored.

3.2.2.2 Parasympathetic System

The determination of the probability distribution for the parasympathetic system is more complex since τ_{PS} follows a uniform distribution between two values. To handle this, let us introduce the following parameters:

- $\tau_{PS_{max}}$ the maximum value for τ_{PS} ,
- $\tau_{PS_{min}}$ the minimum value for τ_{PS},
- p the probability that $\tau(n) < \tau_{PS}$.

Finally, we get

$$\begin{cases} I_{PS} \sim \mathsf{Laplace}(w_{PS}, w_{PS} \times b) & \text{if } \tau(n) < \tau_{PS_{min}} \\ I_{PS} \sim \mathsf{Laplace}(-w_{PS}, w_{PS} \times b) & \text{if } \tau(n) > \tau_{PS_{max}} \\ I_{PS} \sim p \times \mathsf{Laplace}(w_{PS}, w_{PS} \times b) + (1-p) \times \mathsf{Laplace}(-w_{PS}, w_{PS} \times b) & \text{otherwise.} \end{cases}$$

The last line of the equation is actually sufficient to cover all cases, with the first two lines being special cases for $p = 1$ and $p = 0$, respectively.

3.2.2.3 Sympathetic System

The input values I_{SS}^j for the sympathetic systems follow the same probability distribution as the parasympathetic system input, using their own parameter values such as $\tau_{SS_{min}}$, $\tau_{SS_{max}}$, and w_{SS}. However, since the values I_{SS}^j are added N times resulting in the sum $I_{SS} = \sum_{j=1}^{N} I_{SS}^j$, random variables with the same distribution are added, and the probability of this sum can be approximated by a normal distribution. This approximation is valid when N is sufficiently large, which in this case was determined experimentally to be $N > 3$. To simplify our analysis, we define the following terms:

- $\mu_{SS} = p \times w_{SS} + (1-p) \times (-w_{SS})$, the mean of the probability distribution of I_{SS},
- σ_{SS} the standard deviation of the probability distribution of I_{SS}^j. The distribution of I_{SS}^j can actually have two shapes:
  1. The first one is a Laplace distribution. In this case, the standard deviation will remain unchanged, thus $\sigma_{SS} = w_{SS} \times \sigma$ where σ is the standard deviation of the white noise η.
  2. The second case is the bimodal distribution, meaning the distribution has two trends. This is the result of the probability distribution being a linear combination of two Laplace distributions. In the present case, the distribution has two "peaks" which occur at the means of the two Laplace distributions. In this case, the standard deviation can be approximated by:

$$\sqrt{w_{SS}^2 + \sigma^2 - ((2p+1) \times w_{SS})^2}.$$

Finally, we have

$$I_{SS} = \sum_{j=1}^{N} I_{SS}^j \sim \mathcal{N}(N \times \mu_{SS}, \sqrt{N} \times \sigma_{SS}).$$

3.2.2.4 Convolution

Let's take a brief look at convolution before we proceed to derive the probability distribution of $\tau(n+1) - \tau(n)$.

In probability theory, convolution refers to a mathematical operation that involves two probability distributions. Specifically, given two random variables X and Y with probability distributions f and g respectively, the convolution of f and g, denoted $f * g$, is a new probability distribution that represents the probability distribution of the sum Z = X + Y.

The convolution operation can be thought of as a way to combine the probabilities of two events in order to compute the probability of their sum. The basic idea is to integrate the product of the probability density functions of X and Y over all possible values of X and Y that would result in a particular value of Z. This can be expressed mathematically as:

$$(f * g)(z) = \int f(x) g(z - x) dx.$$

3.2.2.5 Sum of the Inputs

Finding the probability density function of the sum of these three components is not a straightforward task. As we aim to calculate the probability density of the sum of random variables, we can employ the concept of convolution.

First, let's approximate the probability density function of $I_{SA} + I_{PS}$:

- if the probability density function of I_{PS} is a Laplace distribution, then

$$I_{SA} + I_{PS} \sim \mathsf{Laplace}(\mu_{SA} + \mu_{PS}, b_{SA} + b_{PS}),$$

- if the probability density function of I_{PS} is a bimodal distribution, then

$$f(I_{SA} + I_{PS}|\theta) = f(I_{SA}|\theta) * f(I_{PS}|\theta).$$

Finally, the probability density function can be estimated in the following way:

$$f(X|\theta) = f(I_{SA} + I_{PS}|\theta) * f(I_{SS}|\theta).$$

3.2.2.6 Problem of the Probability of $\tau(n) < \tau_k$

Now that we have the probability density function of $\tau(n+1) - \tau(n)$ given $\tau(n)$ and θ, the remaining step is to determine the probability p that $\tau(n) < \tau_k$, where τ_k represents the preferred levels.

For the sinoatrial node, this probability can only take values of 1 or 0 since τ_{SA} is constant and known. However, for PS and SS, the preferred levels vary within two bounds, making the computation of this probability more complicated. As the levels are selected using a uniform distribution, a naive approach could be to consider:

$$P(\tau(n) < \tau_k) = \frac{\tau_k - \tau_{lower\_bound}}{\tau_{upper\_bound} - \tau_{lower\_bound}}.$$

However, this approach overlooks the fact that the preferred levels remain constant for a significant amount of time. Therefore, $\tau(n)$ not only depends on $\tau(n-1), \tau(n-2), ...$ but also on these fixed preferred levels. When a time step is reached where the preferred levels have been constant for a while, $\tau(n)$ is likely to have reached a steady state. In such a scenario, it would be incorrect to assume that the probability p is the same as the one previously described. In other words, the proposed probability distribution is only valid if the preferred levels τ_k change at each time step.

In the current scenario, calculating the probability p appears impractical as it relies on several variables, such as the possibility that $\tau(n)$ may be in a steady state, but there is no ways of determining it as it also depends on the duration T for which a preferred level τ_k remains unchanged.

Therefore, it is unlikely to obtain precise outcomes by applying the maximum likelihood estimation method to the datasets' heart rate signals. However, for validation purposes, it was still conducted with the assumption that $T = 1$, to confirm that despite the model's significant deviation from reality, the maximum likelihood method doesn't provide outcomes that differentiate between healthy individuals and those with diseases.

3.2.3 Implementing and Assessing the Method

3.2.3.1 Method Implementation

The maximum likelihood implementation for the model was carried out in several steps. First, the probability distribution for each input I_x using the formulas defined in section 3.2.2 was constructed, after which they were gradually combined using convolution.

For the SA node, the required parameters to determine the likelihood are:

- w_{SA} the strength of the feedback input,

- the standard deviation of the white noise η,
- τ_{SA} the preferred level of the SA node,
- $\tau(n)$ the previous RR interbeat.

For the parasympathetic system, the required parameters to determine the likelihood are:

- w_{PS} the strength of the feedback input,
- the standard deviation of the white noise η,
- $\tau_{PS_{min}}$ the lower bound for preferred level,
- $\tau_{PS_{max}}$ the upper bound for preferred level,
- $\tau(n)$ the previous RR interbeat.

For the sympathetic system, the required parameters to determine the likelihood are:

- w_{SS} the strength of the feedback inputs,
- the standard deviation of the white noise η,
- $\tau_{SS_{min}}$ the lower bound for preferred level,
- $\tau_{SS_{max}}$ the upper bound for preferred level,
- $\tau(n)$ the previous RR interbeat,
- N the number of feedback inputs.

3.2.3.2 MLE with One Feedback Input

Method To verify the exactitude of the computation of the probability density function of each input I_x, the maximum likelihood estimation was performed but by considering that there is only one of the input of Equation 3.1. Thus, we have:

$$\tau(n+1) - \tau(n) = I_x.$$

In order to perform the maximum likelihood estimation, the first step is obviously to determine the likelihood for a given signal. For that we can use the formulas for I_x (SA, SS and PS) described in section 3.2.2. We compute the likelihood of each points in the provided signal excepted the first one (since we don't have $\tau(-1)$). Then, we compute the log-likelihood by summing up the log.

To maximize the log-likelihood, we can use the `minimize` function from the `scipy.optimize` library, which is designed for minimization rather than maximization. To overcome this, we minimized the negative log-likelihood. In the end, we will have:

$$w_x^* = min_{w_x} - \sum_n \log(f(\tau(n+1) - \tau(n) | \tau(n), \theta))$$

with θ being the parameters needed to compute the probability density.

In order to check that it works, a signal is generated with a given value for w_x and the other values for the strengths of the feedback inputs are set to zero. For example, if we are considering SA, then we generate a signal with $w_{SA} = y$ with y being a given value and $w_{SS} = w_{PS} = 0$.

For all three inputs, 500 signals composed of 5000 beats were generated with a random value for their strength w_x. A maximum likelihood estimation on these signals was performed to estimate these values of w_x. The absolute difference between the estimated w_x^* and the actual value w_x was then computed in order to assess the precision of the estimation.

This was performed twice:

1. once with the time T varying as specified in the model (see section 3.1.3),
2. once with $T = 1$ such that the naive probability distribution p is correct (see section 3.2.2.6 for more details).

Results Results of the two experiments are available in Figures 3.18 and 3.19. The parameter estimations for SA and PS were accurate in both cases. However, the estimations for SS were only accurate in the second case. The absolute difference between the estimated value of w^*_{SS} and the actual value could reach up to 0.012, which is significant considering that the values of w_{SS} were chosen between 0 and 0.1. Additionally, a loss in precision can be expected when combining multiple inputs. Therefore, in the following analysis, we will only consider the second case where the value of T is fixed at 1.

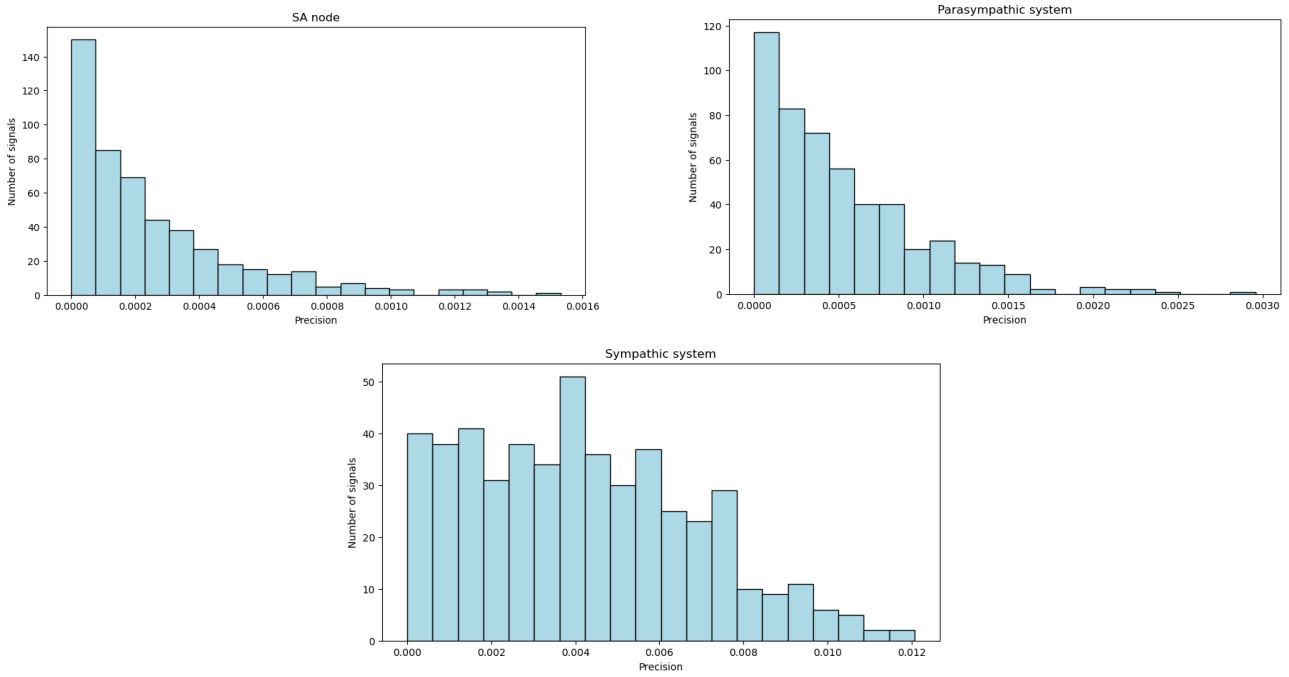


Figure 3.18: The histograms display the frequency of signals based on the absolute difference between the estimated w^*_x with MLE and the actual parameter value of a signal generated with the Ivanov model. The experiment generated 500 signals using the Ivanov model with w_x chosen randomly between 0 and 0.1. The remaining parameters used in this experiment are consistent with those specified in section 3.1.3, except for the two other feedback input strengths, which are set to 0 (e.g., $w_{SA} = y$ and $w_{PS} = w_{SS} = 0$).

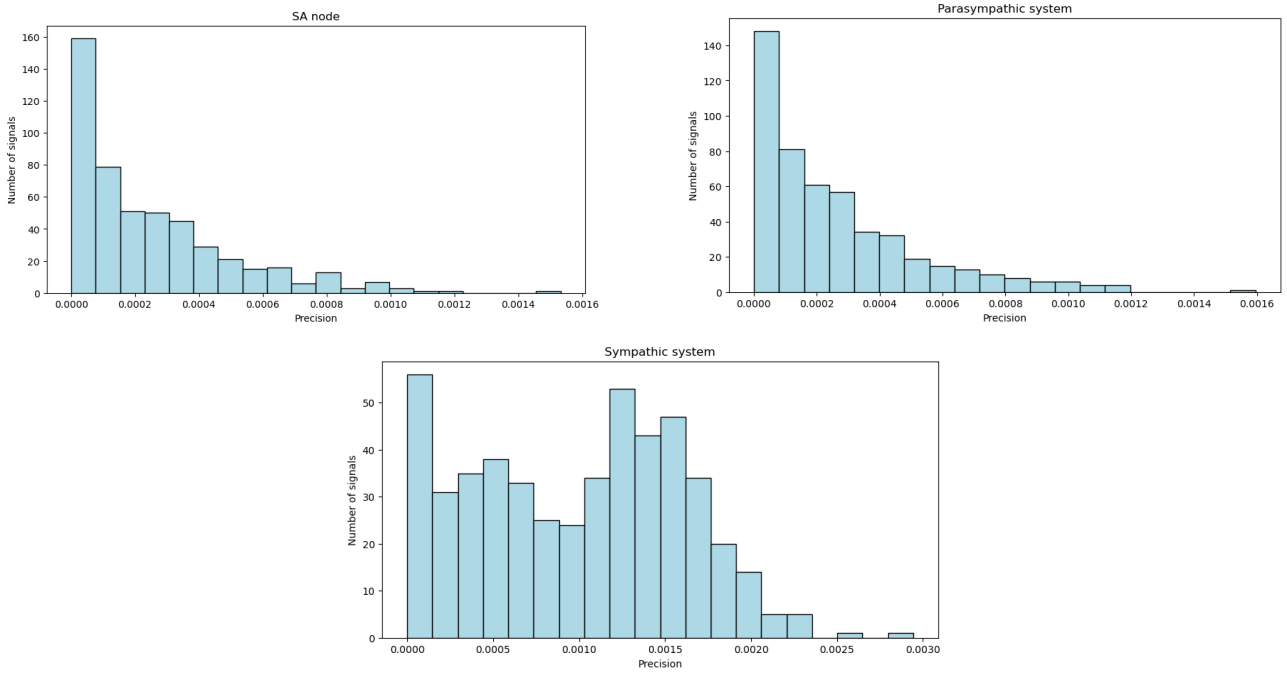


Figure 3.19: The histograms display the frequency of signals based on the absolute difference between the estimated w_x^* with MLE and the actual parameter value of a signal generated with the Ivanov model. The experiment generated 500 signals using the Ivanov model with w_x chosen randomly between 0 and 0.1. The remaining parameters used in this experiment are consistent with those specified in section 3.1.3, except for the two other feedback input strengths, which are set to 0 (e.g., $w_{SA} = y$ and $w_{PS} = w_{SS} = 0$) and the time T which is set to 1.

3.2.3.3 MLE with Two Feedback Inputs

Method Now that we have confirmed the accuracy of the maximum likelihood estimation for one input, we can now evaluate the same approach considering that the value $\tau(n+1) - \tau(n)$ is dependent on two inputs:

$$\tau(n+1) - \tau(n) = I_x + I_y.$$

The process of computing the separate likelihoods remains unchanged, but in order to determine the probability of $I_x + I_y$, we utilize the `convolve` function from the `scipy.signal` library. Before the convolution can be performed, we first need to compute $f_{I_x}(x|\theta)$ and $f_{I_y}(x|\theta)$ for all points, which is obviously unfeasible. Thus, we compute the values within a chosen interval that is defined by the bounds $mean \pm 3 \times std$, with a precision of 0.0001. In other words, the value of $f(x|\theta)$ is computed for every 0.0001 values in the specified interval.

The convolution operation is computationally expensive, and hence the maximum likelihood estimation was carried out on 50 signals with random values of w_x and w_y chosen between 0 and 0.1. Additionally, the length of the signals was decreased to 1000 points.

Results The results of these experiments can be found in Figure 3.20. The majority of the experiments produced accurate results, but there were some cases where the accuracy was poor. This can be attributed to the imprecision of the convolution process, which causes a loss of accuracy. The imprecision can create local minimums which the optimization algorithm `minimize` can get stuck in, leading to less accurate results.

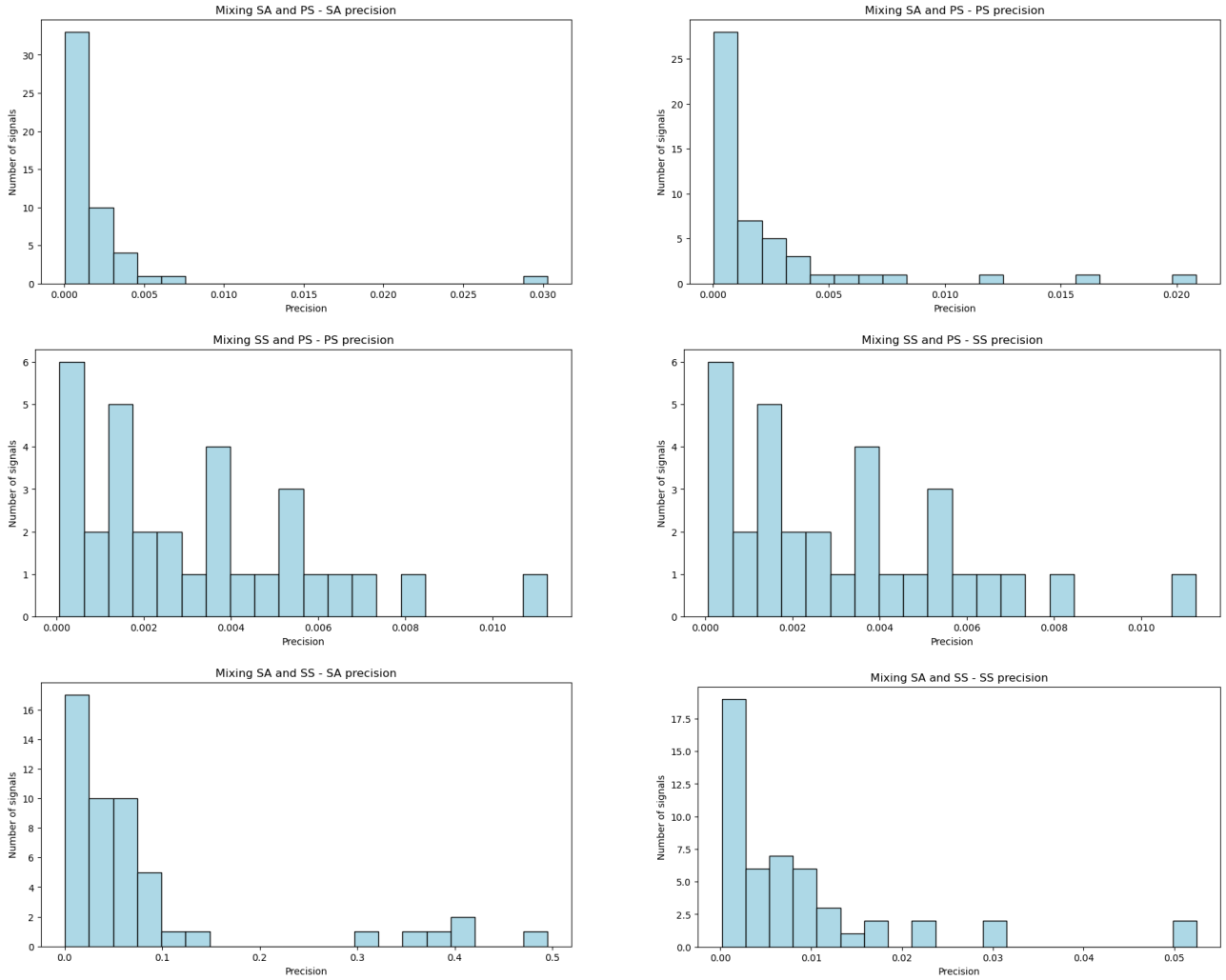


Figure 3.20: The histograms illustrate the frequency of precision values (absolute difference between estimated and actual values) for the estimated w_x^* and w_y^* (mix of 2 inputs) using MLE on signals generated with the Ivanov model. The top histograms show estimation results on signals not containing the SS inputs (w_{SS} fixed to 0), the middle ones show results for signals without the SA input (w_{SA} fixed to 0), and the bottom ones show results without the PS input (w_{PS} fixed to 0). T is set to 1, and the remaining parameters follow those specified in section 3.1.3. The estimations were performed 50 times on signals with 1000 points. The values of w_x and w_y were randomly selected between 0 and 0.1. The abscissa axis of the middle histograms is limited to 0.015 for clarity, but 8 precision values exceeded this limit, with 5 around 0.08.

3.2.3.4 MLE on the Whole Model

Method Finally, using convolution, the likelihood estimation was performed by estimating the likelihood of the sum of all inputs as in Equation 3.1. The parameter estimation was conducted on 50 signals consisting of 1000 points, with the strengths of the feedback input randomly selected between 0 and 0.1.

Results The results are available on Figure 3.21. They show that the estimation of the feedback input strengths is often inaccurate, with over half of the experiments yielding an absolute difference between the estimated value w_x^* and the true parameter value w_x greater than 0.1. This inaccuracy can be attributed to the loss of precision incurred during the convolution, which causes the minimization to get trapped in a local minimum.

Ideally, to avoid this issue, one would aim for a precision level lower than the tolerance of the

`minimize` algorithm. However, achieving such a level of precision would require substantial computational resources.

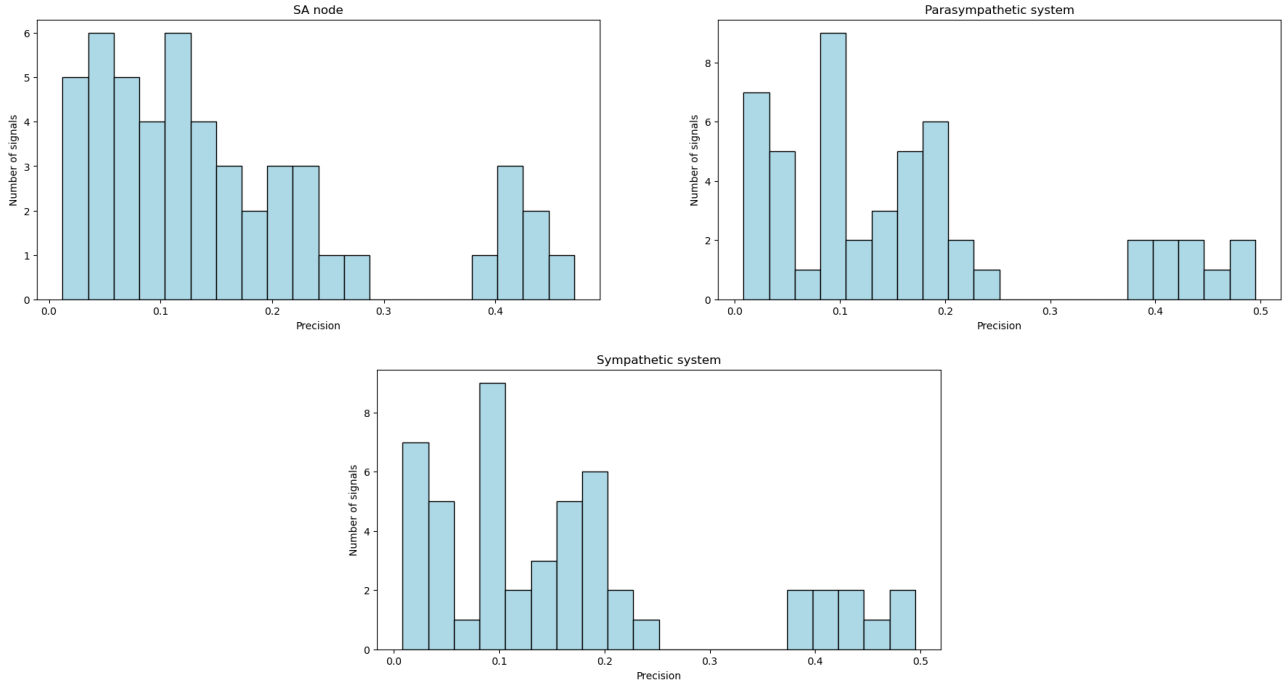


Figure 3.21: Histograms representing frequency of signals that had a given precision (absolute difference between the estimated value for the estimated parameter and the actual value) for the estimated w_x^*. The signals generated with the Ivanov model on which the maximum likelihood estimation was performed are made of 1000 points and 50 trials were performed. T is set to 1 and the values for w_{SA}, w_{SS} and w_{PS} were randomly selected between 0 and 0.1. The remaining parameters are the same as the ones described in section 3.1.3.

3.2.4 Conclusion

Although the maximum likelihood estimation is a commonly used method for estimating parameters from data, it is not a suitable approach for the Ivanov model due to several issues. Firstly, it is highly challenging to obtain an analytical form of the model's likelihood, as explained in section 3.2.2.6. While a naive expression can be derived and tested, a second problem arises from the fact that the Ivanov model is a stochastic process composed of a sum of inputs, which means it is also a sum of random variables. This requires the use of convolution to compute the likelihood of the model, which is highly computationally intensive and results in a loss of precision. Therefore, we can conclude that the maximum likelihood estimation is not a suitable method for parameter estimation in the Ivanov model.

3.3 Second Method for Parameter Estimation: Approximate Bayesian Computation

3.3.1 Method Overview

In the previous section, it was shown that accurately estimating the parameters based on the likelihood is not feasible. An alternative method is to use approximate Bayesian computation (ABC) [40]. This approach is often employed when the likelihood function is either intractable or computationally expensive. Instead of computing the likelihood function, ABC approximates the posterior distribution of parameters $p(\theta|X)$ by simulating the system and comparing the simulated data to the observed data using a distance function. The basic concept of ABC is to accept parameter values that generate simulated data that are similar to the observed data, up to a specified tolerance level. The most commonly used algorithm is the rejection algorithm, which follows the following steps [43]:

1. Sample a candidate parameter θ^* from a prior distribution $p(\theta)$;
2. Simulate a dataset X^* from the model;
3. Compare the simulated dataset X^* with the experimental data X_0, using a distance function d and a tolerance ϵ. If $d(X_0, X^*) \leq \epsilon$, accept θ^*;
4. Repeat until N parameter values have been accepted.

The choice of distance function is not limited to any specific type as long as it effectively measures the similarity between the observed and simulated data. Examples include the Euclidean distance or Manhattan distance.

Assessing the similarity between the observed and the simulated data solely by measuring the distance between them can be inadequate, especially in the case of time series where it is often challenging to replicate the exact same series. Instead, a common approach is to measure the distance between a summary statistic $S(X)$ that reduces the data to a more manageable set of features, such as mean, standard deviation, autocorrelation, etc. This allows for a more effective comparison between the observed and simulated data.

Selecting the appropriate summary statistics, distance metric, and threshold value is crucial for the performance of the ABC algorithm and depends on the specific case at hand. Hence, these choices should not be overlooked.

The rejection algorithm has a major drawback: if the prior distribution $p(\theta)$ differs significantly from the posterior distribution $p(\theta|X)$, the algorithm will generate many θ^* that are far from the true value of θ. As a result, the algorithm will have a high rejection rate, which makes it highly inefficient.

To address this issue, we introduce an alternative version of the ABC algorithm based on sequential Monte Carlo (SMC) [43, 38]. The ABC SMC algorithm aims to iteratively refine the approximation of the posterior distribution of parameters by selecting multiple thresholds, denoted as $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T > 0$. The algorithm operates as follows:

1. Initialize $\epsilon_1, \epsilon_2, ..., \epsilon_T$ and set the population indicator to t=0;
2. Set the particle indicator $i = 1$ (a parameter estimation value is called a particle);
3. If $t = 0$, sample θ^{**} independently from $p(\theta)$. Else, sample θ^* from the previous population θ_{t-1} with the weights w_{t-1} and perturb the particle to obtain $\theta^{**} \sim K_t(\theta|\theta^*)$ where K_t is a perturbation kernel.

   If $p(\theta^{**}) = 0$, return to step 3.

   Simulate a candidate dataset X^*.

   If $d(X^*, X_0) \geq \epsilon_t$, return to step 3;

4. Set $\theta_t^{(i)} = \theta^{**}$ and compute the weight for particle $\theta_t^{(i)}$ in the following way

$$w_t^{(i)} = \begin{cases} 1 & \text{if } t = 0 \\ \frac{p(\theta_t^{(i)})}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(j)})} & \text{if } t > 0. \end{cases}$$

   if $i < N$, set $i = i + 1$ and go to step 3.

5. Normalize the weights.

   If $t < T$, set $t = t + 1$, go to step 2.

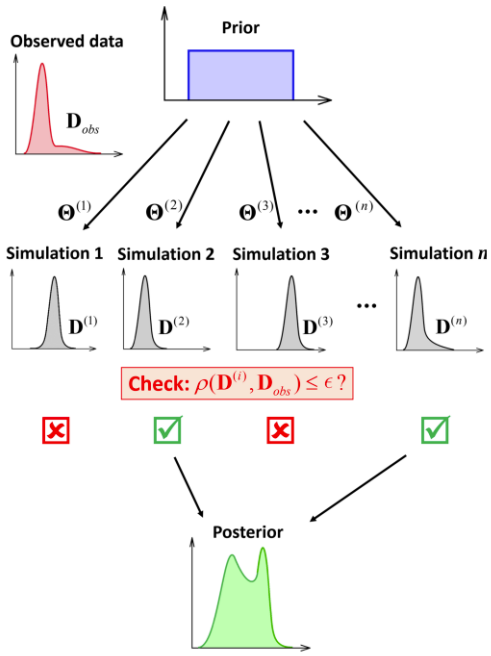


Figure 3.22: Conceptual overview of the ABC method with the rejection algorithm. *Figure from ref [16].*

3.3.2 Applying ABC to the Ivanov Model: Tools Selection

3.3.2.1 Summary Statistics

Three summary statistics are chosen for the ABC algorithm on the heart rates:

- **The DFA coefficient:** since the fractal dynamic of the heart rate is being considered, this summary statistic is an obvious choice for parameter estimation;
- **The autocorrelation:** long-range correlation is a statistical characteristic of 1/f signals, as described in section 1.2.1.1, so it makes sense to use this statistic;
- **The variance:** the parameters to be estimated are the strengths of the feedback inputs (see section 3.1.5.6), which have an impact on the signal amplitude and, consequently, the variance of the signal. The variance is relatively small compared to the two previous statistics, so a factor of 50 (determined experimentally) is used to increase its importance.

3.3.2.2 Distance Function

For the ABC algorithm on the heart rates, the Euclidean distance was chosen to measure the similarity between the observed and simulated data. This is a common choice in ABC as it is easy to compute and

interpret. Moreover, it provides a natural way to use the summary statistics that we selected. Indeed, since the summary statistics are real-valued vectors, the Euclidean distance naturally quantifies the distance between them. Other distance functions could have been chosen, but the Euclidean distance is a reasonable first choice and often yields good results in practice.

3.3.2.3 Thresholds

The chosen thresholds are generated in three stages as follows:

1. The first 20 thresholds are evenly spaced between 1 and 0.1;
2. The next 20 thresholds are evenly spaced between 0.1 and 0.01;
3. The final 20 thresholds are evenly spaced between 0.01 and 0.001.

However, it was found that some of the parameter estimation computations were excessively slow at certain thresholds, with little difference in the resulting parameter distributions. Therefore, a time limit was imposed on the parameter estimation computation, and it may not always complete at the final threshold.

3.3.2.4 Prior Distribution

The chosen prior distribution for the parameters w_{SA}, w_{SS}, and w_{PS} is a uniform distribution given by $w_x \sim \mathsf{Uniform}(0, 0.1)$.

3.3.2.5 Perturbation Kernel

The perturbation kernel for parameter estimation is commonly implemented using either a normal distribution or a uniform distribution. For this particular project, a normal distribution is deemed more suitable, and thus we selected θ^{**} according to the following procedure:

$$\theta^{**} \sim \mathcal{N}(\theta^*, 0.01).$$

The choice of a standard deviation of 0.01 is considered appropriate because the original distribution of $p(\theta)$ is a uniform distribution between 0 and 0.1. This value is small enough to avoid considering an excessively large range of values, yet large enough to prevent getting trapped in a narrow range of values if an iteration yields parameter estimates far from the correct value.

3.3.2.6 Number of Points for the Time Series

The length of the time series was set to 5000 points, which strikes a balance between efficiency and accuracy. The ABC algorithm produces a substantial number of time series, so the longer the time series, the more time the parameter estimation will take. However, the fractal pattern of the signals generated by the Ivanov model is due to the signal varying based on preferred levels that themselves vary based on a time T (see sections 3.1.5.5 and 3.1.4). This time T follows a normal distribution with a mean of 1000. Therefore, a value of 5000 seems appropriate as it will capture various modifications in the preferred levels but will not generate excessively large signals.

Since the time series used in this project for the Ivanov model have a size of 20.000, the subset of 5.000 points to be analyzed will be selected randomly, starting from a randomly chosen initial point.

3.3.3 Implementing and Assessing the Method

The implementation of the method utilized the tools outlined in the preceding section. As explained in section 3.1.6, the evaluation of the method involved generating signals with predetermined parameter values with the Ivanov model. The estimated parameters obtained through the ABC method are then compared to the original parameter values to assess the performance of the method.

The Ivanov model parameters to be estimated are the feedback input strengths w_{SA}, w_{SS}, and w_{PS}, as discussed in section 3.1.5.6. Figure 3.23 shows that the ABC algorithm does not provide accurate results. However, it shows promising results for estimating a single feedback input strength, even for w_{SS}, which was not the case for maximum likelihood estimation, as described in section 3.2.3.2. Therefore, a revised model will be presented.

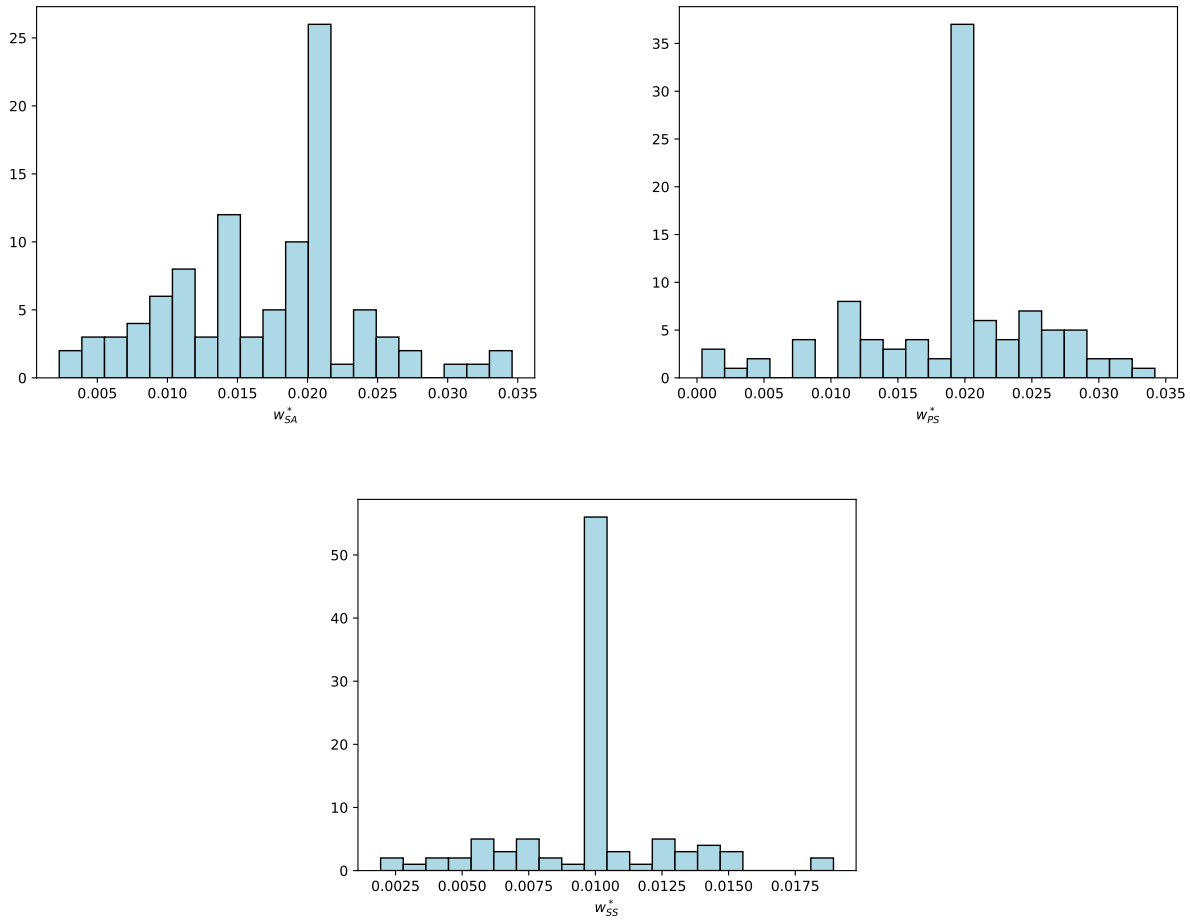


Figure 3.23: The histograms depict the parameter estimation distribution obtained using ABC SMC for a signal generated by the Ivanov model, with the parameter values set as described in section 3.2.3.2. The same tools described in section 3.3.2 were used, and 100 parameter estimations were generated for each considered threshold. The top left panel shows the estimations for w_{SA}, the top right panel shows the estimations for w_{PS}, and the bottom panel shows the estimations for w_{SS}.

3.3.4 Revisited Ivanov Model

3.3.4.1 Model Overview

In this section, we examine a modified version of the Ivanov model (previously discussed in section 3.1). Although the underlying stochastic process remains the same, the interpretation will be slightly different due to simplifications. Specifically, we still assume that the time between beats $\tau(n)$ follows a random walk process that is biased towards preferred levels. However, in contrast to the previous model, we omit the inputs for the sinoatrial (SA) node and the parasympathetic system. We then have:

$$\tau(n+1) - \tau(n) = \sum_{j=1}^{N} I^j(n, \tau^j(n)) \tag{3.4}$$

with

$$I(n) = \begin{cases} w(1+\eta) & \text{if } \tau(n) < \tau^j \\ -w(1+\eta) & \text{otherwise.} \end{cases} \tag{3.5}$$

In the revised version of the Ivanov model, the inputs are not assigned to any particular system but are rather considered as inputs that can be influenced by both the sympathetic and parasympathetic systems. The preferred levels change over time, as in the original model, and their values determine the impact of the sympathetic or the parasympathetic systems. If a preferred level at a particular time is close to the center of the range, it will be considered as an input that isn't influenced by any system and follows the rhythm of the SA node.

Upon reflection, this version appears to be more coherent, particularly for the inputs of the sympathetic system. This is because the sympathetic system is intended to accelerate the heart rate, but its preferred level, representing the desired interbeat time, could be greater than the preferred level of the SA node. Additionally, the uniform distribution for this input was centered on τ_{SA} (the preferred level of SA).

3.3.4.2 Parameters Value Choice

The majority of the parameters in this model are analogous to those in the original model. To avoid repetition, we only provide a brief overview of them. For a more in-depth understanding of the reasoning behind each parameter, please refer to sections 3.1.3 and 3.1.5. The parameters in this model are as follows:

- For the number of inputs considered N, the original value was set to 7. Considering that we removed two inputs and that I_{PS} had a bigger impact than the other inputs, we will now set N to 10.
- The white noise η remains unchanged (Laplace distribution with $scale = 0.5$ and $mean = 0$).
- The preferred level τ^j will remain the same than for the SS inputs of the original model, i.e. centered around the SA level and bounded between 0.2 and 1. These levels follow a uniform distribution.
- The preferred levels will remain constant for T beats before changing. The distribution of T remains unchanged (normal distribution with $mean = 1000$ and $standard\ deviation = 500$)
- The parameter value for w was the most challenging to determine as it was the parameter we aimed to fit to certain subjects in the model. To select an appropriate value for this parameter, we conducted an analysis similar to that in section 3.1.5.1, and the outcomes are presented in Figure 3.24. Based on these results, w was set to 0.015.

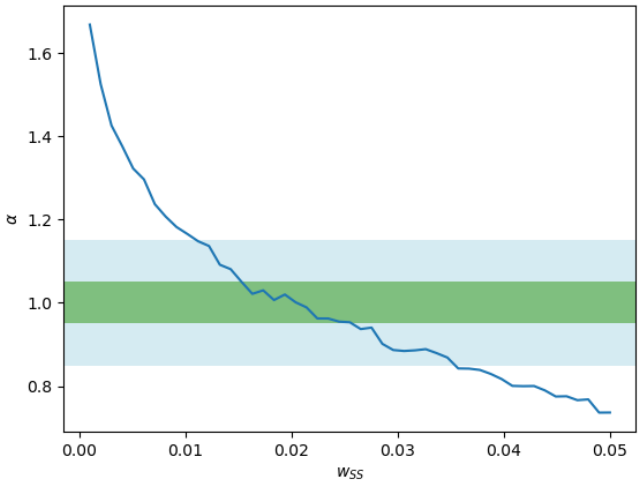


Figure 3.24: This graph shows the variation of the α coefficient calculated using the DFA method with respect to the parameter w. The lag values used for the analysis range from 5 to 500 and the signals used for the calculation are generated with the revisited Ivanov model and are of 20.000 points. The parameter w varies between 0 and 0.05, with 50 different values considered. For each value of w, 10 trials were performed, and the mean of the results is displayed in the graph. The green strip represents α coefficients that fall within the range of 0.95 to 1.05, while the blue strips represent coefficients that are 0.5 to 0.1 away from the significant value of 1.

3.3.4.3 Implementing and Assessing the Model

Implementation No implementation of the model is necessary as it can be modeled by changing the parameters of the original model in the following way:

1. $w_{SA} = w_{PS} = 0$,
2. $w_{SS} = 0.015$,
3. $N = 10$,
4. $\tau_{SS_{lower}}$, $\tau_{SS_{upper}}$, $\tau_{PS_{lower}}$, $\tau_{PS_{upper}}$ and τ_{SA} are unchanged,
5. the mean and standard deviation of T are unchanged,
6. the mean and scale of η are unchanged.

Assessing Figure 3.25 shows that the signal generated by the model is very similar to the heart rate of a healthy subject, indicating that the model is a reliable approximation. Furthermore, DFA was conducted on 1000 signals (see Figure 3.26), and the mean α coefficient was found to be around 1.073, which is quite close to the significant value of 1. Thus, this model generates a signal that successfully captures the fractal pattern of the heart rate.

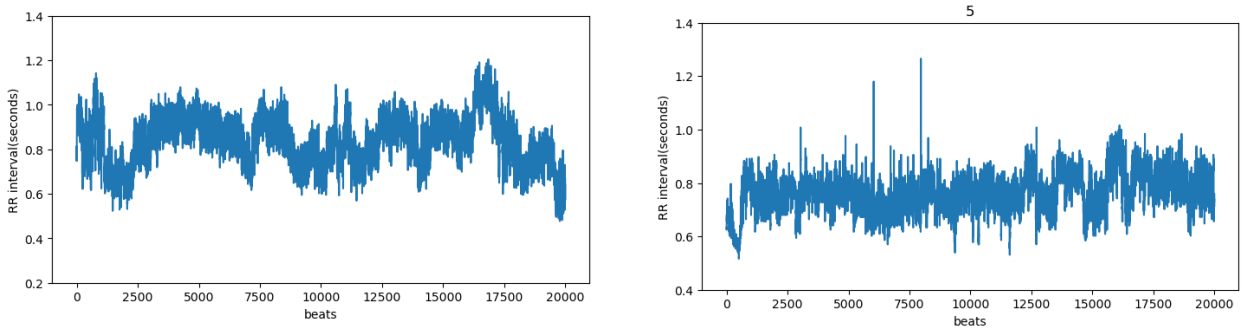


Figure 3.25: The figure on the left shows the heart rate signal generated by the revisited Ivanov model, with parameter values of $w = 0.015$, $\tau^j \in [0.2, 1]$, and $\eta \sim \mathsf{Laplace}(0, 0.5)$. On the right, we can see the first 20.000 interbeats of the heart rate signal from the fifth healthy subject in the second dataset (section 1.3.2).

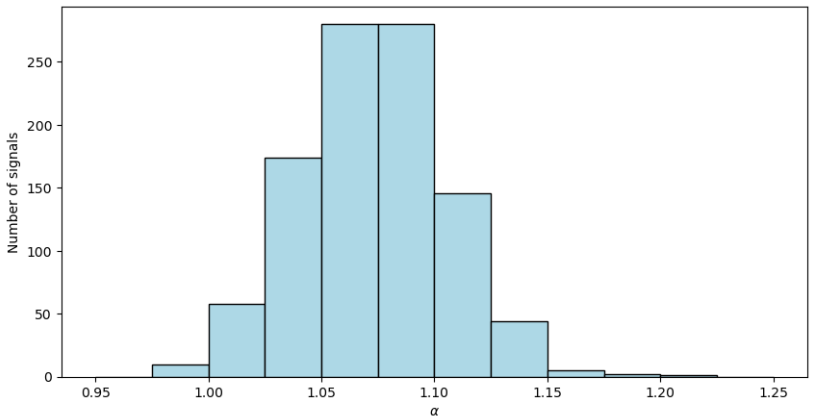


Figure 3.26: The histogram shows the distribution of the α coefficients computed using the DFA method on 1000 signals generated with the revisited Ivanov model. The lags considered are between 5 and 500. The parameter values used for the model are $w = 0.015$, $\tau^j \in [0.2, 1]$, and $\eta \sim \mathsf{Laplace}(0, 0.5)$.

3.3.5 ABC on the Revisited Model

Similar to the approach taken for the Ivanov model, we choose to fit the feedback input strength w for various signals. This is motivated by the same reasons discussed in section 3.1.5.6.

3.3.5.1 Assessing ABC on the Ivanov Model Revisited

Before proceeding with parameter estimation on the heart rate signals using the ABC algorithm, it is important to verify the reliability of the method, just like for the maximum likelihood estimation (see section 3.2.3). To assess the accuracy of the ABC algorithm, the parameter w using signals generated from the revisited Ivanov model was estimated. Three different estimations of the parameter w for signals were performed with the following values:

1. $w = 0.005$,
2. $w = 0.01$,
3. $w = 0.025$,
4. $w = 0.05$,
5. $w = 0.075$.

The parameter estimation results obtained using the ABC algorithm are depicted in Figure 3.27. It can be observed that the ABC method performs quite well in estimating the parameter w, particularly for small values. Although the algorithm appears to be slightly less accurate for higher values of w, the relative difference between the actual value and the estimated value remains similar to the relative difference observed for smaller values (see Figure 3.28). Therefore, while the precision of the estimation may decrease for higher values of w, the ABC method can still be considered a reliable estimator for the revisited Ivanov model parameter w.

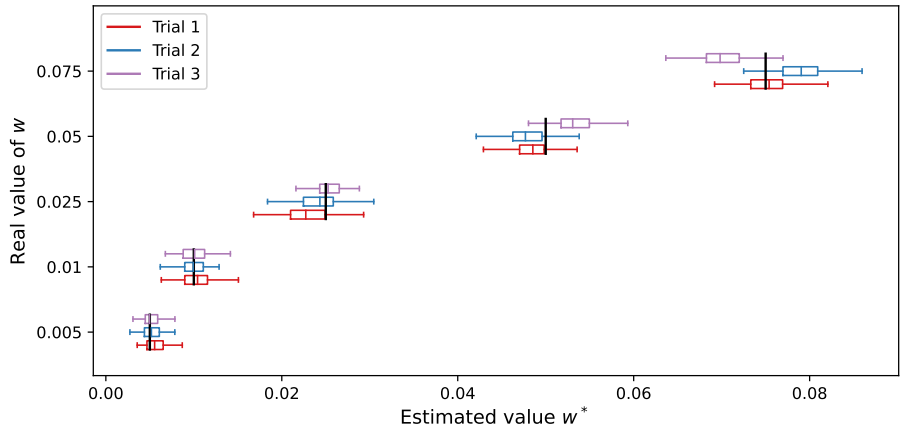


Figure 3.27: This figure presents boxplots displaying the distribution of parameter estimations w^* obtained using the ABC algorithm on signals generated with the revisited Ivanov model. The parameter w was set to 0.005, 0.01, 0.05, and 0.075, and for each value, three signals were generated and their corresponding w parameter was estimated 100 times for each threshold. All parameters of the Ivanov model are the same as described in section 3.3.4.2, except for the feedback input strength. The ABC SMC algorithm tools used are the same as in section 3.3.2. The computation was terminated after a specified time limit, with the last threshold considered falling within the range of 0.01463158 and 0.08410526.

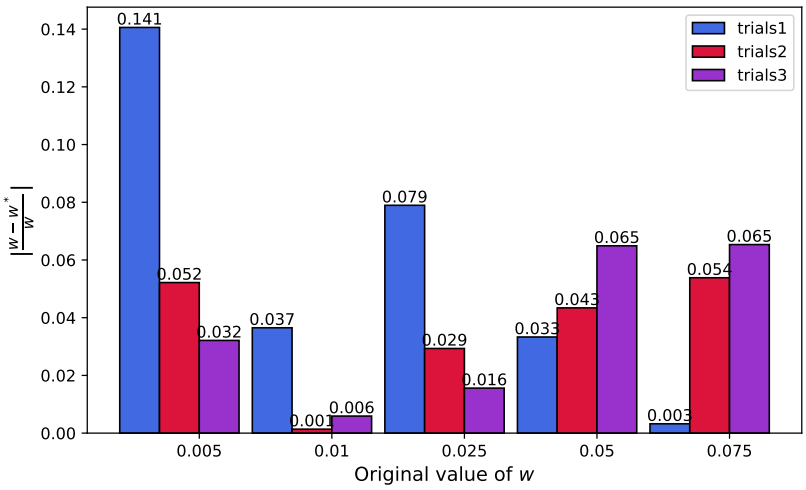


Figure 3.28: This figure serves as a summary of figure 3.27. For each parameter value considered, the relative difference between the estimated parameter value w^* and its actual value w is computed for each trial.

3.3.5.2 Parameter Estimation on Heart Rate Signals

Method Having established the reliability of the ABC method for parameter estimation in the revisited Ivanov model, we can now apply it to estimate the parameter w for real subjects. The Ivanov model, which was previously considered for time series of 20.000 points, will now be applied to the signals in the dataset described in section 1.3.2. Since all the signals in this dataset consist of at least 70.000 points, they were divided into four separate time series, each containing 20.000 points apart for the last one that might be smaller.

The analysis was conducted in two stages. In the first stage, a fitting was performed on the initial 20.000 points of each subject's signal, allowing for a longer computation time and thereby achieving greater precision. The second stage involved the remaining three time series, corresponding to the points in the intervals [20.000, 40.000], [40.000, 60.000], and [60.000, 80.000]. Although the computation time was shorter in the second stage, it still provided sufficiently accurate parameter estimations.

Results The results of the first fitting can be seen in Figure 3.29, while the results of the second fittings are shown in Figure 3.30. Although there is some overlap between the healthy subjects and those with congestive heart failure, a clear distinction can still be observed between the two groups based on the parameter w. The parameter w tends to be smaller for subjects with congestive heart failure, indicating that the influences on the heart rate from the components (SA node, sympathetic and parasympathetic systems) are weakening. This suggests a decrease in the effectiveness of these components in regulating the heart rate.

In contrast, subjects with atrial fibrillation can be easily identified by their significantly larger estimations of the feedback input strength w^*. According to the model, this suggests a substantial influence of the components in affecting the heart rate.

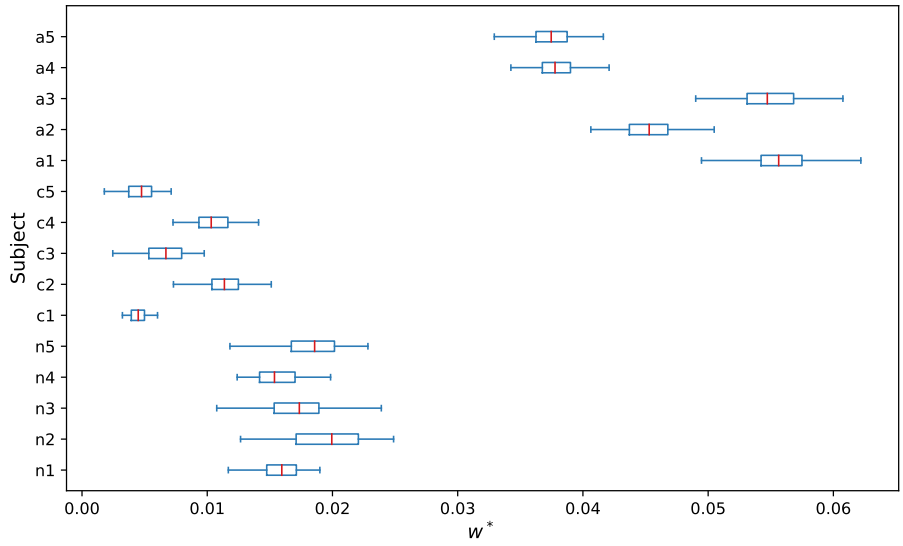


Figure 3.29: The boxplots illustrate the distribution of parameter w (of the revisited Ivanov model) estimations obtained using ABC on real heart rate signals from subjects in the second dataset (refer to section 1.3.2). The subjects are categorized into three groups: the first five subjects (group n) are healthy, the middle five subjects (group c) have congestive heart failure, and the last five subjects (group a) have atrial fibrillation. The ABC tools described in section 3.3.2 were employed for the analysis. The parameter fitting was conducted on the first 20.000 points of each time series. The ABC algorithm was executed for a specific time limit, resulting in a final threshold range between 0.05631579 and 0.098.

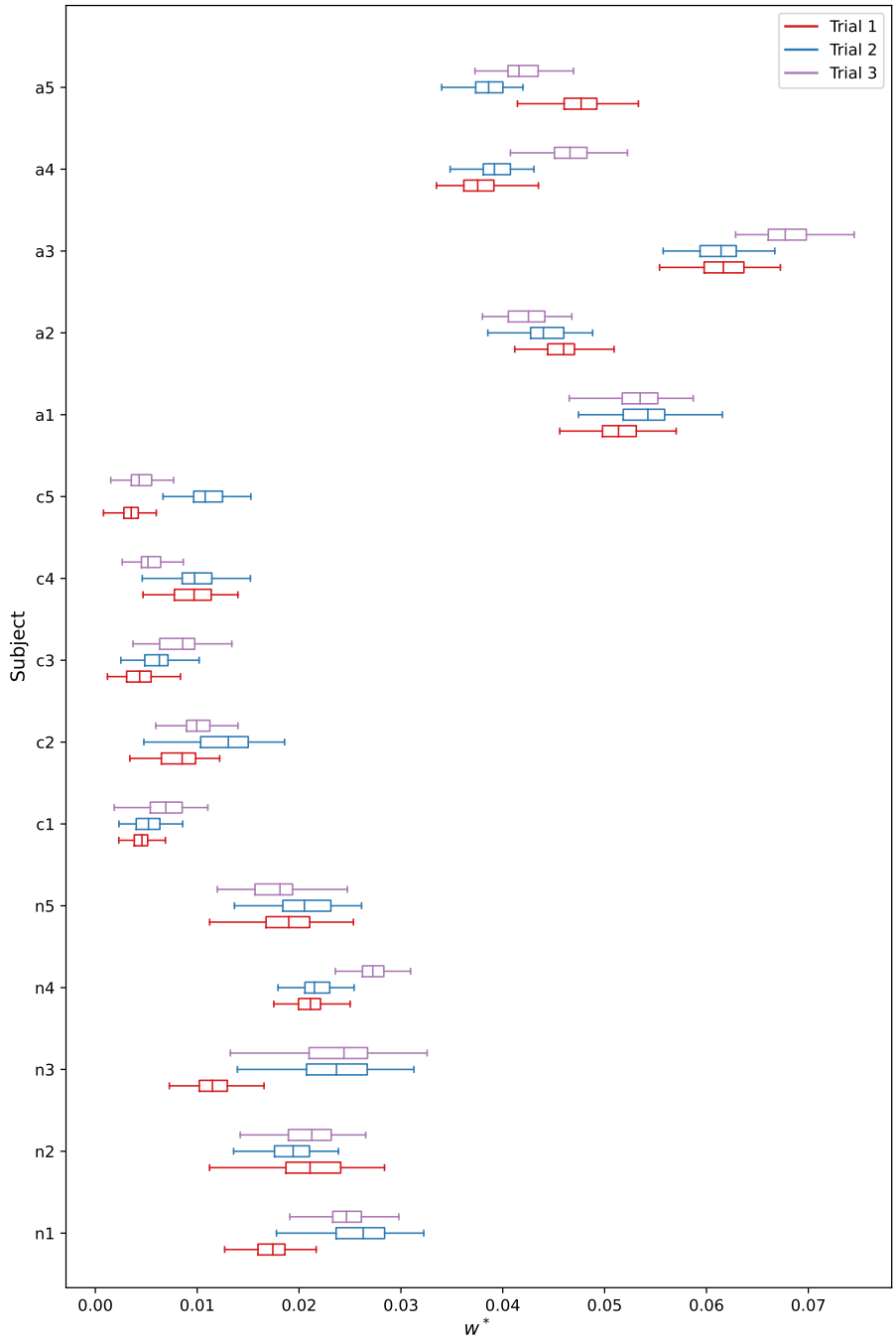


Figure 3.30: The boxplots illustrate the distribution of parameter w (of the revisited Ivanov model) estimations obtained using ABC on real heart rate signals from subjects in the second dataset (refer to section 1.3.2). The subjects are categorized into three groups: the first five subjects (group n) are healthy, the middle five subjects (group c) have congestive heart failure, and the last five subjects (group a) have atrial fibrillation. The ABC tools described in section 3.3.2 were employed for the analysis. The fitting process was conducted three times, each time on a different portion of the subject signals. Trial 1 focused on beats within the interval [20.000, 40.000], trial 2 on [40.000, 60.000], and trial 3 on [60.000, 80.000]. The ABC algorithm was executed with a specified time limit, resulting in a final threshold range between 0.09336842 and 0.1463158.

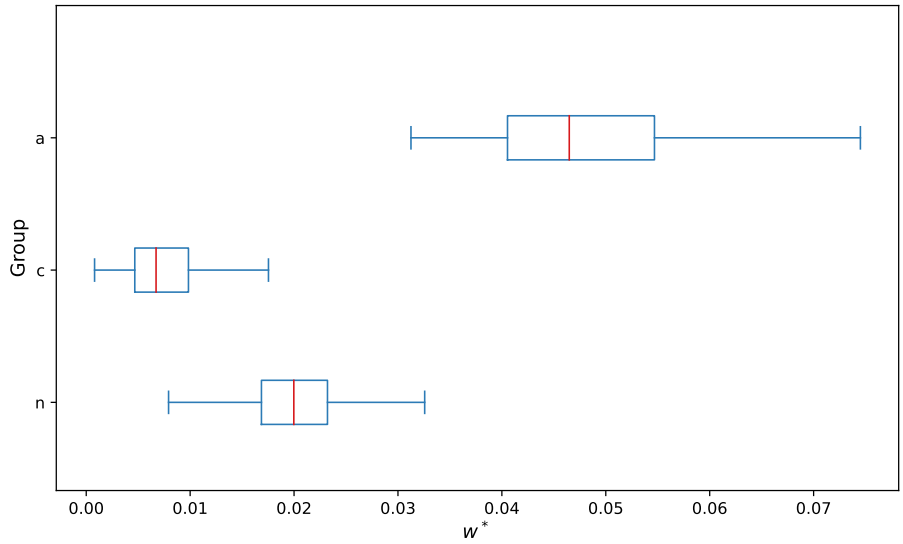


Figure 3.31: The boxplots illustrate the distribution of parameter w (of the revisited Ivanov model) estimations obtained using ABC on real heart rate signals from subjects in the second dataset (refer to section 1.3.2). The boxplots provide a summary of the parameter estimation distribution within each group of the dataset (n for healthy, a for atrial fibrillation, and c for congestive heart failure), serving as a concise representation of the results depicted in Figures 3.29 and 3.30.

3.3.6 Conclusion

Similar to the MLE method, the ABC method demonstrated limitations in accurately estimating the parameters of the Ivanov model. However, unlike the previous method, the ABC method can be employed to estimate the parameters of a revised model (section 3.3.4).

The ABC method specifically estimates the feedback input strength parameter w. The estimations of this parameter for the subjects in each group yielded interesting results, as noticeable distinctions were observed between the groups. This shows the potential for modeling each group separately using the Ivanov model.

3.4 Third Method for Parameter Estimation: Neural Networks

In the preceding sections, we explored two different methods for parameter fitting of the Ivanov model on real heart rate time series. However, neither of these methods yielded accurate parameter estimations. In this section, we introduce a final approach using neural networks to address this challenge.

3.4.1 Applying neural networks to the Ivanov Model

3.4.1.1 Libraries

The time series will be embedded using `PyTorch`.

The `lampe` package[1] was employed to estimate the parameters. This package provides methods that, given the data, estimates the posterior distribution of the parameters. It is implemented using `PyTorch`.

3.4.1.2 Datasets

Three distinct datasets were created, each serving a specific purpose:

- the training set: This set consists of 2^{16} time series,
- the validation set: This set consists of 2^{12} time series,
- the testing set: This set consists of 2^{10} time series.

All time series within these datasets contain 20.000 data points and are generated using the Ivanov model. The parameters used to generate the signals are set to their default values (see section 3.1.3) except for the parameters under estimation, namely the feedback input strengths w_{SA}, w_{SS}, and w_{PS}. These strengths are randomly chosen within the range of 0 to 0.1 for each individual time series.

To efficiently store and load the dataset, the `lampe.data` module provides a class called `HD5Dataset`. The use of this process of generating and storing the datasets was heavily influenced by the `lampe` tutorial [2].

3.4.1.3 Embedding

Prior to parameter estimation, the dimensionality of the time series needs to be reduced. To accomplish this, an embedding was constructed using convolutional neural networks (CNN). CNNs have demonstrated remarkable effectiveness in analyzing images and uncovering significant patterns within them. Subsequently, they have been applied to time series analysis and proven to be a dependable tool for extracting patterns and facilitating classification [46]. The embedding architecture employed in this study is specifically designed to convert the input time series data into a learned feature representation.

Theoretical background A convolutional network, also known as a convolutional neural network (CNN), is a type of deep learning model designed for processing structured grid-like data, such as images or time series data [28, 1].

The key component of a convolutional network is the convolutional layer, which applies a set of learnable filters (also called kernels) to the input data. These filters slide over the input, computing a dot product at each location and producing a feature map. This process allows the network to capture local patterns and spatial dependencies present in the input. It has several features:

- The number of channels which refers to the depth dimension of the input and output data. It represents the number of feature maps or filters used in that layer. Each channel focuses on capturing different types of features in the input data.

[1] `https://github.com/francois-rozet/lampe`

[2] `https://lampe.readthedocs.io/en/stable/tutorials/simulators.html`

- The kernel size, also known as the filter size, which determines the spatial extent of the receptive field used for the convolution operation. It specifies the dimensions (width and height) of the sliding window that moves over the input data.

- Padding which is an optional technique used to preserve spatial information during the convolution operation. It adds extra pixels or values around the input data, effectively increasing its size.

Typically, convolutional networks also include pooling layers, which reduces the spatial dimensions while retaining important information.

Convolutional networks often conclude with fully connected layers, which connect all the neurons from the previous layer to the next layer, and finally end with a softmax layer for classification or a regression layer for regression tasks.

During training, convolutional networks optimize their parameters through backpropagation and gradient descent, aiming to minimize a chosen loss function. This process allows the network to learn to recognize complex patterns and features in the input data, making them well-suited for tasks such as image classification, object detection, and image segmentation.

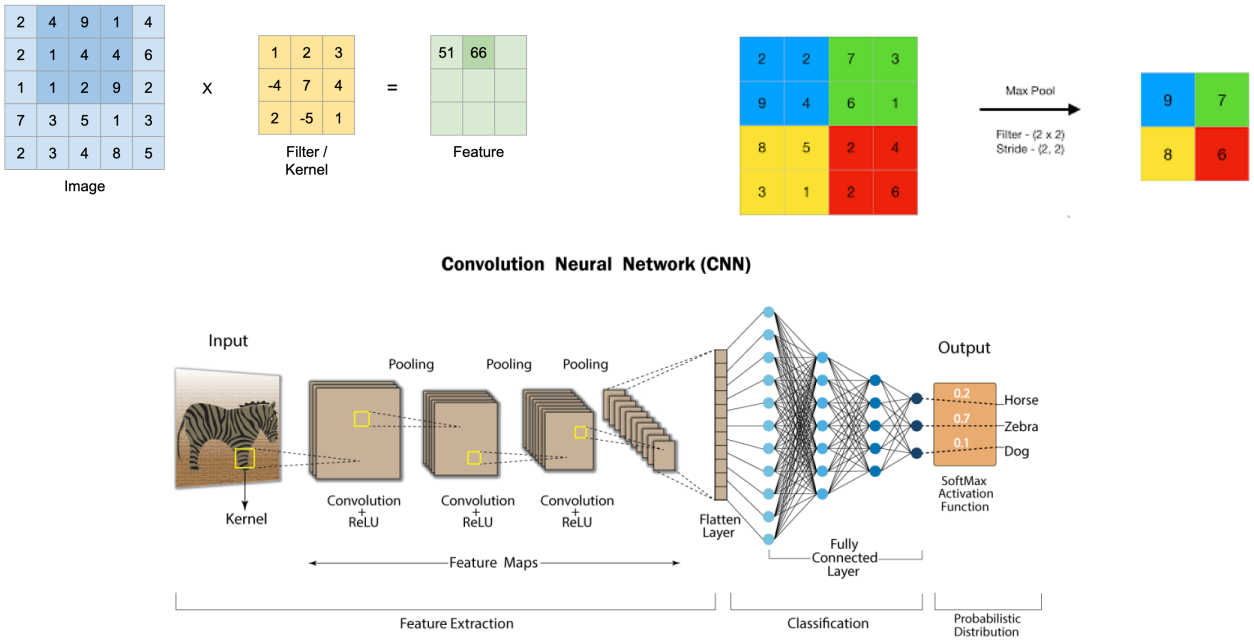


Figure 3.32: These illustrations depict the underlying principles of a convolutional neural network (CNN). The figure on the top left demonstrates the operation performed by the convolutional layer, while the figure on the top right represents the operation carried out by the pooling layer. The bottom figure provides an overview of the general architecture of a CNN. Source: `https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-understand-the-basics/`.

Embedding architecture

- **Convolutional layers:** The embedding starts with a 1D convolutional layer that takes as input a single channel, representing the time series data. This layer convolves the input with 16 filters, each with a kernel size of 1.

  Following the initial convolutional layer, a stack of 10 additional convolutional layers is employed. Each of these layers uses the same number of channels as the previous layer (16), a kernel size of 3 and a padding of 1.

- **Activation function:** After each convolutional layer, a scaled exponential linear unit (SELU) activation function is applied.
- **Pooling layers:** To reduce the spatial dimensions, max pooling is applied after each convolutional layer. Max pooling is performed with a kernel size of 3 and a stride that varies depending on the layer position.
- **Flatten and fully connected layer:** Following the stack of convolutional and pooling layers, a flattening operation is applied to convert the multi-dimensional feature maps into a one-dimensional vector. This flattened representation is then passed through two fully connected layers which reduces the data to a vector of size 8.

3.4.1.4 Model Architecture

After creating the embedding, the remaining architecture follows a straightforward approach. The `lampe` package provides the `NPE` module, which takes the embedded time series as input and returns an approximate posterior distribution $p(\theta|x)$, where θ represents the parameters to be estimated and x represents the time series.

The implementation closely follows the approach outlined in the `lampe` tutorial [3], with similar steps and procedures.

3.4.2 Training and Assessing the Method

Training The neural network is trained using the training set and validated using the validation set. Training is performed using batches of size 256, and the `torch.cuda` library is utilized to accelerate the process on compatible hardware.

Expected coverage To evaluate the accuracy of the neural network estimation, the `lampe.diagnosis` module provides methods to calculate the expected coverage of the model.

The concept of expected coverage, as defined in the `lampe` tutorial, can be described as follows: "The expected coverage of a posterior estimation $p(\theta|x)$ at a credible level $1-\alpha$ is the probability of a set of parameters θ^* to be included the highest density region of total probability $1-\alpha$ of the posterior estimator, given an observation $x^* \sim p(x|\theta^*)$" [4].

The highest posterior density region corresponds to the credible region with the smallest volume. A credible region is a space Θ that satisfies the condition [17]:

$$\int_{\Theta} p(\theta|\mathbf{x} = \mathbf{x_O})d\theta = 1 - \alpha.$$

If the expected coverage at a credible level $1-\alpha$ is equal to $1-\alpha$, the estimator is considered calibrated. In other words, the plot of expected coverage should align along the diagonal. If the curve falls below the diagonal, it indicates overconfidence of the model, while a curve above the diagonal suggests underconfidence.

Testing the model The expected coverage of the trained network was evaluated on the testing set. The results are available on Figure 3.33. It can be seen that even though the curve isn't totally aligned with the diagonal, it is pretty close, which means that the model is pretty calibrated. On Figure 3.34, the parameters estimations on two different time series of the testing set are available. We can observe that the first one is a very good approximation while the other one is not totally accurate. However, even though the parameter estimations won't be totally accurate, we can still conclude that the estimations remain quite close.

[3] `https://lampe.readthedocs.io/en/stable/tutorials/embedding.html`
[4] `https://lampe.readthedocs.io/en/stable/tutorials/coverage.html`

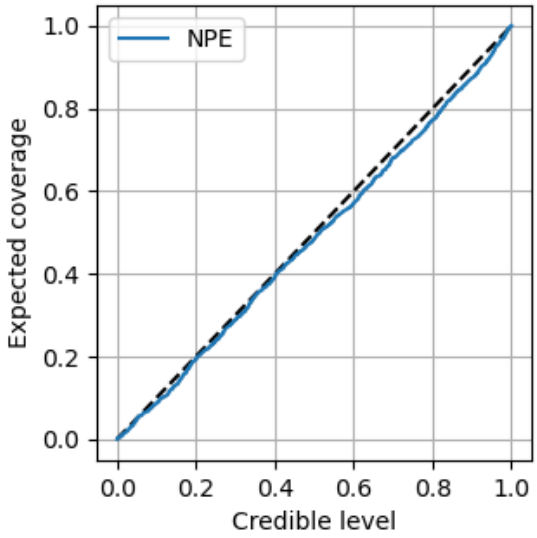


Figure 3.33: Expected coverage of the trained neural network. A curve bellow the diagonal means an overconfident model while a curve above it means a underconfident model.

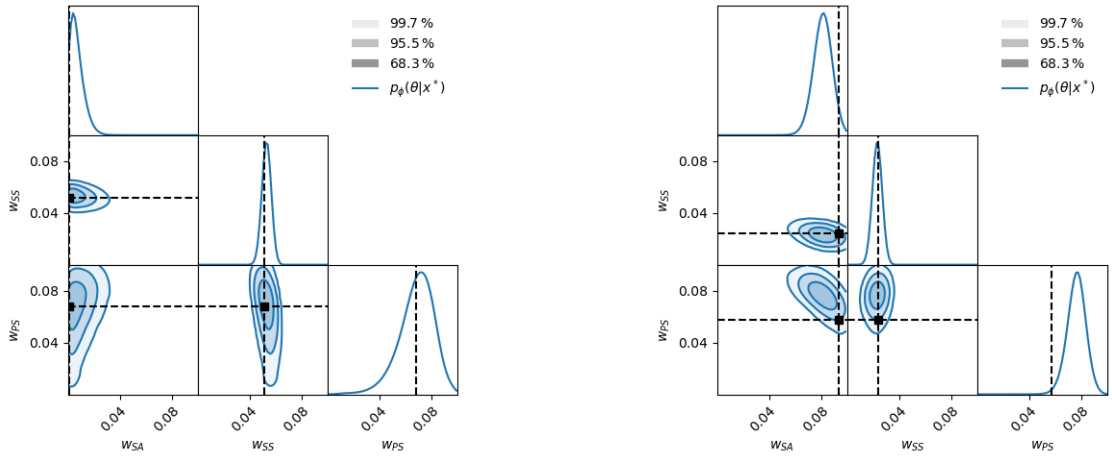


Figure 3.34: Parameters estimation for two time series of the testing set with the trained neural network. The estimation on the left is the one of the first time series of the training set and the one on the right is for the 25^{th} time series of the training set. The parameters estimated are the feedback input strengths of the Ivanov model (more details in section 3.1.5.6). The cells located on the rightmost side depict the posterior probability distribution $p(w_x|X)$ of the parameter value for the respective time series. The remaining cells represent the joint probability distribution for two parameters.

3.4.3 Parameters Estimation of Heart Rate Signals

Given that the model was trained on time series consisting of 20.000 points, the parameter fitting for the Ivanov model on real heart rate signals also needed to be performed on time series of the same length. To achieve this, the first 60.000 points of each signal from each subject of the second dataset (see section 1.3.2) were considered, and the fitting process was conducted three times on different segments of the signals.

The obtained results can be analyzed in two steps. Firstly, we can examine the results for healthy subjects and assess the proximity of the estimated values to those chosen in this study. Secondly, we can investigate any differences that may exist among the various groups.

Results of healthy subjects The original model for healthy subjects [21] established the reference values as $w_{SA} = w_{SS} = w_{PS}/3 = 0.01$, which were also utilized in this project. Figures 3.35, 3.36,

and 3.38 demonstrate that the estimations for at least w_{SA} and w_{PS} in healthy subjects' signals are remarkably close to these initial values apart from the first healthy subject which exhibits a relatively high value for w_{SA}. Nevertheless, referring to Figure 3.7 in section 3.1.5.1, where the impact of the parameters was analyzed, this value is not high enough to have a substantial impact on the fractal dynamic of the heart rate.

On the other hand, the parameter w_{SS} tends to be slightly smaller, especially for subjects n1, n2, and n5. In section 3.1.5.1, it was observed that a small value for w_{SS} can affect the fractal dynamics of the heart rate signal. However, this effect is counterbalanced by the higher estimations of w_{PS}.

Overall, despite some differences between the estimations and the original model values, the estimations remain quite close. This finding further strengthens the claim that the Ivanov model can be effectively employed to model the heart rate of healthy subjects.

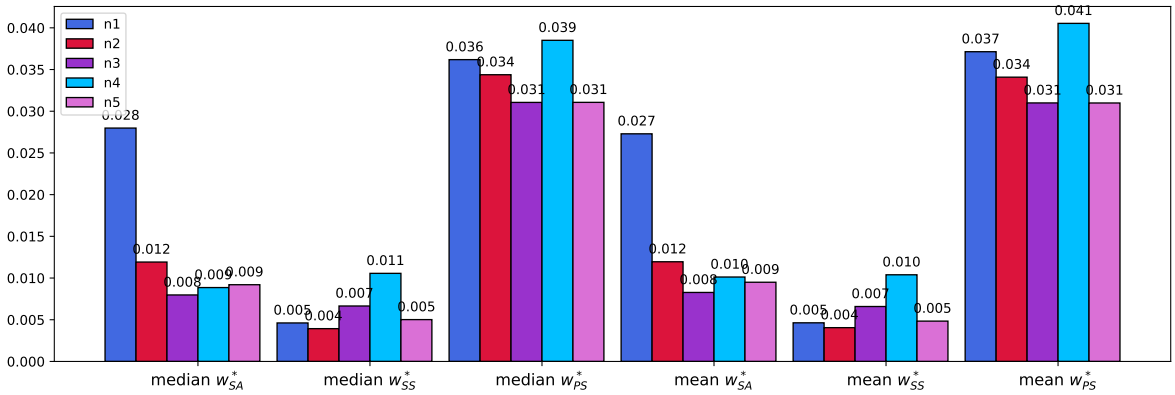


Figure 3.35: The figure illustrates the median and mean of the parameter estimations for the three parameters (w_{SA}, w_{SS}, and w_{PS}) of the Ivanov model, specifically for each healthy subject in the second dataset (refer to section 1.3.2).

Results of all groups From Figures 3.36, 3.37 and 3.38, we can observe a similar trend to the fitting using ABC (refer to section 3.3.5.2): the strengths of the feedback inputs tend to decrease for congestive heart failure and increase significantly for atrial fibrillation.

However, several additional observations need to be mentioned. Firstly, for congestive heart failure, the decrease in strengths is mainly noticeable for w_{SA} and w_{PS}, while the decrease for w_{SS} is less apparent.

Secondly, it can be observed that the predictions for subjects with atrial fibrillation have a higher variance (indicating greater uncertainty) compared to subjects in the other two groups. It could be hypothesized that atrial fibrillation, being closer to white noise, exhibits a highly randomized signal, making it challenging to accurately model. However, this increased uncertainty is primarily observed for w_{SA}, indicating that this parameter have a lower importance in the case of atrial fibrillation. Upon closer examination of Figure 3.38, it can be seen that for w_{PS}, the uncertainty is primarily driven by subjects a5 and a4, while the predictions for w_{PS} among the three remaining subjects have the lowest variance of all subjects. Furthermore, although the variance for w_{SS} remains relatively high, it is still small enough to differentiate subjects with atrial fibrillation from those in the other two groups. In conclusion, while w_{SA} cannot be clearly confined to a specific interval for modeling the heart rate of subjects with atrial fibrillation, w_{SS} can be placed within a distinct interval, and caution should be exercised when interpreting w_{PS} due to its large variance in two of the subjects.

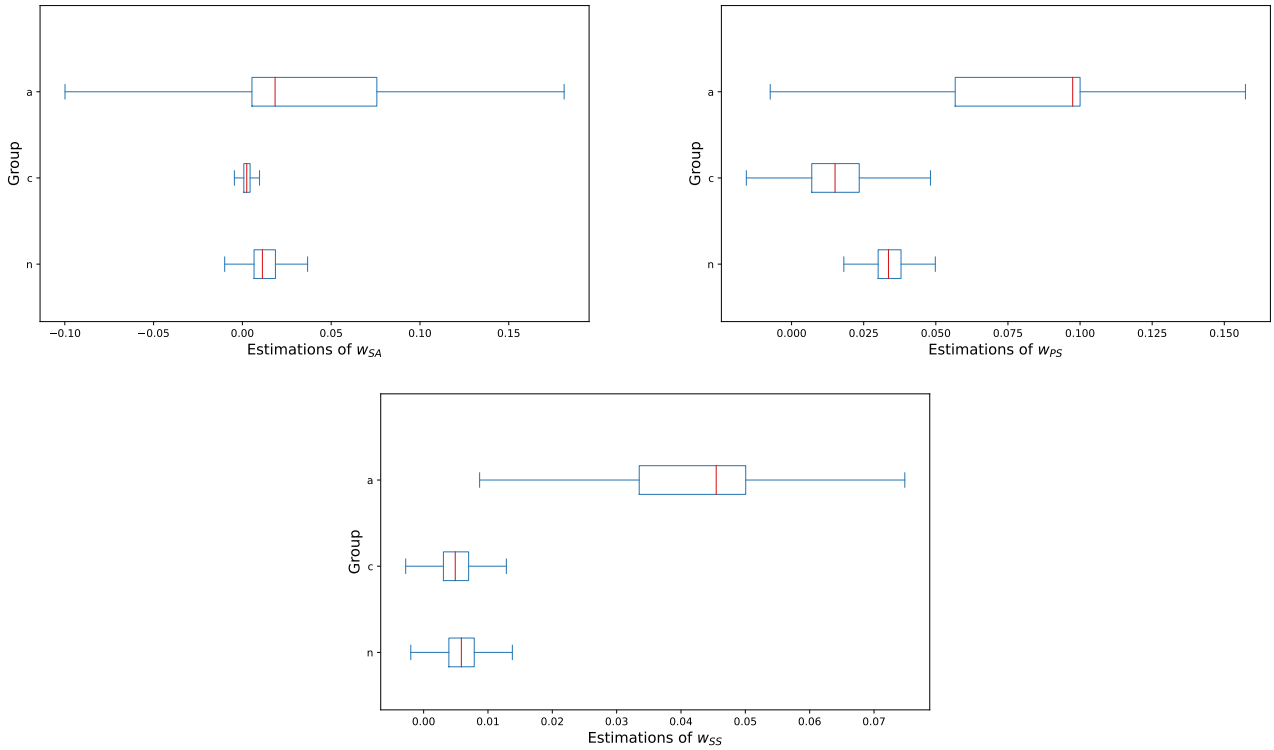


Figure 3.36: The boxplots display the distribution of parameter estimations for w_{SA} (top left), w_{PS} (top right), and w_{SS} (bottom) within each group (n for healthy, a for atrial fibrillation, c for congestive heart failure) of the second dataset (see section 1.3.2). The parameter fitting was performed using the neural network architecture with three trials on the first 60.000 points of each signal (20.000 points for each trial). Each boxplot represents the distribution of estimations from the three trials for each subject within the group.

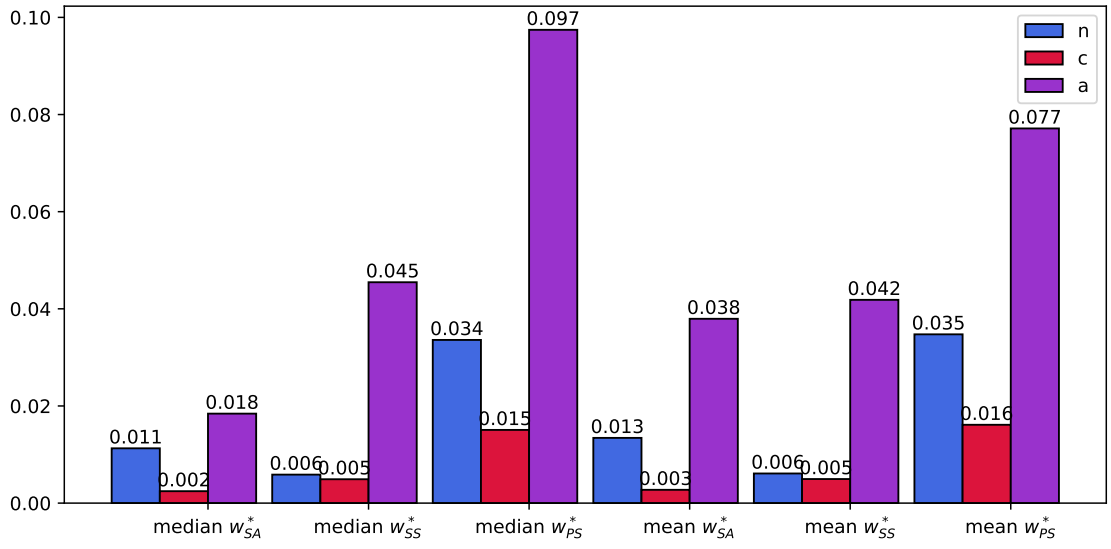


Figure 3.37: The mean and median of the parameter estimations for w_{SA}, w_{SS}, and w_{PS} of the Ivanov model are shown for each group (n for healthy, a for atrial fibrillation, and c for congestive heart failure) in the second dataset (see section 1.3.2).

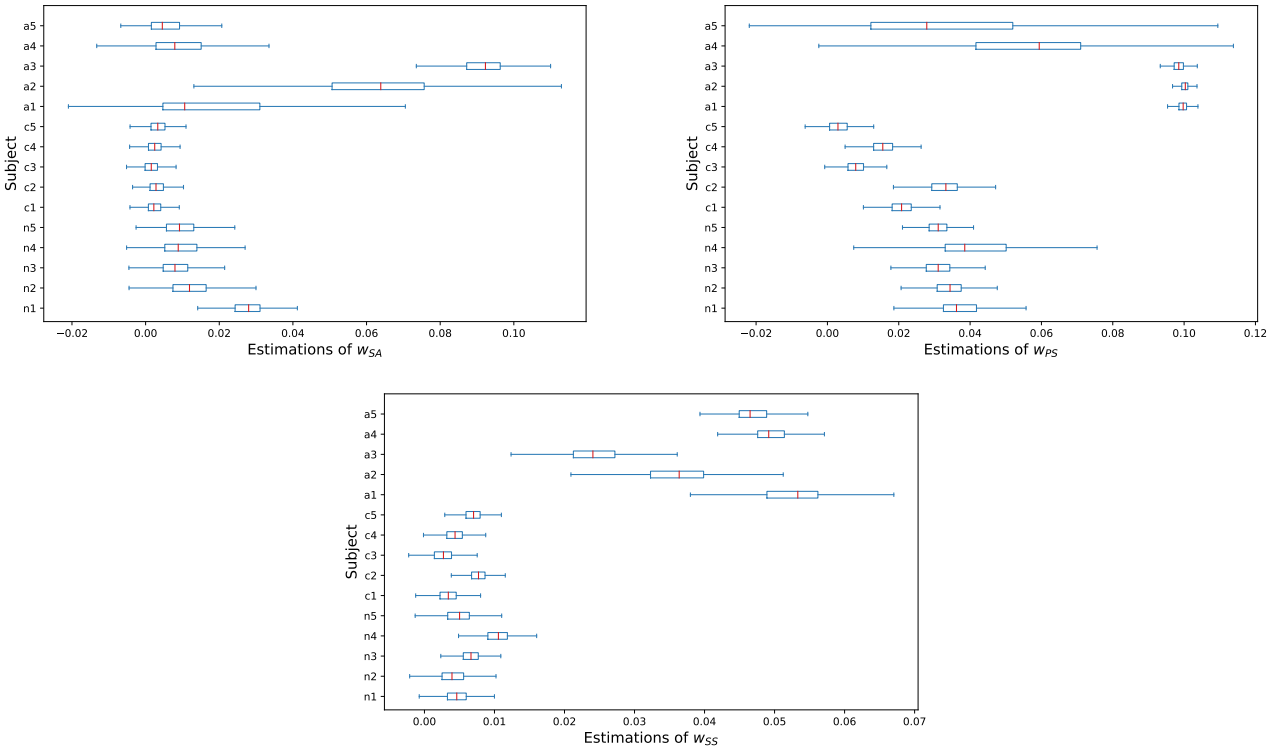


Figure 3.38: The boxplots depict the distribution of parameter estimations for w_{SA} (top left), w_{PS} (top right), and w_{SS} (bottom) across subjects in the second dataset (see section 1.3.2). The parameter fitting was conducted using the neural network architecture with three trials on the first 60.000 points of each signal (20.000 points for each trial). Each boxplot represents the distribution of estimations from the three trials. A more detailed distribution of each trial is available in appendix C.

3.4.4 Conclusion

Among the methods explored, the utilization of a neural network emerged as the only reliable approach to estimate the desired parameters of the Ivanov model.

The parameter estimations obtained from the heart rate signals in the second dataset (section 1.3.2) yielded interesting insights. Firstly, the estimations for healthy subjects closely aligned with those of the original model, providing reassurance regarding the effectiveness of the original model in simulating the heart rate of healthy individuals. Secondly, notable distinctions were observed among the different groups, indicating the potential for individualized modeling of heart rate based on the specific disease.

3.5 Results Summary and Interpretation

3.5.1 Results Summary

In the previous sections, we investigated several methods to estimate the feedback input strengths parameters of the Ivanov model. The objective was to apply these methods to actual heart rate signals in order to obtain the best parameter fit for the model. Among these methods, two emerged as particularly noteworthy:

- the ABC method applied to a revisited model (refer to section 3.3.4),
- a neural network utilizing the `lampe` package.

Both of these approaches provided sets of parameter estimations that could be utilized to approximate the posterior distribution $p(\theta|x)$ of the parameters. The outcomes of the fittings for each subject can be found in sections 3.3.5.2 and 3.4.3.

The key findings were as follows:

- The neural network fitting for the parameters produced values similar to those in the original work for healthy subjects, indicating that the model is effective for simulating heart rate signals.
- For subjects with congestive heart failure, the strengths had lower values than those for healthy subjects, suggesting a reduction in the impact of the sympathetic and parasympathetic systems.
- In contrast, for subjects with atrial fibrillation, the strengths were very high, indicating an over influence of the sympathetic and parasympathetic systems.

	Group n	Group c	Group a
w with ABC	0.0201 ± 0.0046	0.0074 ± 0.0033	0.0482 ± 0.0091
w_{SA} with the neural network	0.0134 ± 0.009	0.0027 ± 0.0028	0.0379 ± 0.0367
w_{SS} with the neural network	0.0061 ± 0.0032	0.005 ± 0.0027	0.0418 ± 0.0113
w_{PS} with the neural network	0.0347 ± 0.0073	0.0161 ± 0.0112	0.0771 ± 0.0313

Table 3.2: For each group (n for healthy, c for congestive heart failure, and a for atrial fibrillation), this table presents the intervals of parameter values that can be utilized to generate a heart rate signal using the (revisited) Ivanov model. The intervals were computed as $mean \pm std$, using parameter estimations obtained through the ABC method (section 3.3) and the neural network (section 3.4).

3.5.2 Assessing the Fit to Real-World Data

Comparing time series Now that we have performed fittings using several methods, it is interesting to evaluate their relative effectiveness. However, comparing time series poses a challenge as it involves techniques from the field of time series data mining, which is an active area of research [44]. Most existing methods for comparing time series involve measuring the "closeness" of two time series, either by directly computing the Euclidean distance between them or by applying alignment techniques to superimpose them and then assessing the difference [9, 45, 4]. However, these methods have limitations, including the fact that many proposed techniques are specific to particular datasets [22, 8]. Additionally, in the context of this project, we are dealing with noisy data, which means that identifying similar trends among the time series is not feasible. Consequently, aligning the time series and calculating distances between them won't provide meaningful insights.

Hence, our main objective is to evaluate the model's ability to generate signals that exhibit similar characteristics to the original ones. Specifically, we will analyze the fractal dynamics of the generated signals in comparison to their corresponding subjects.

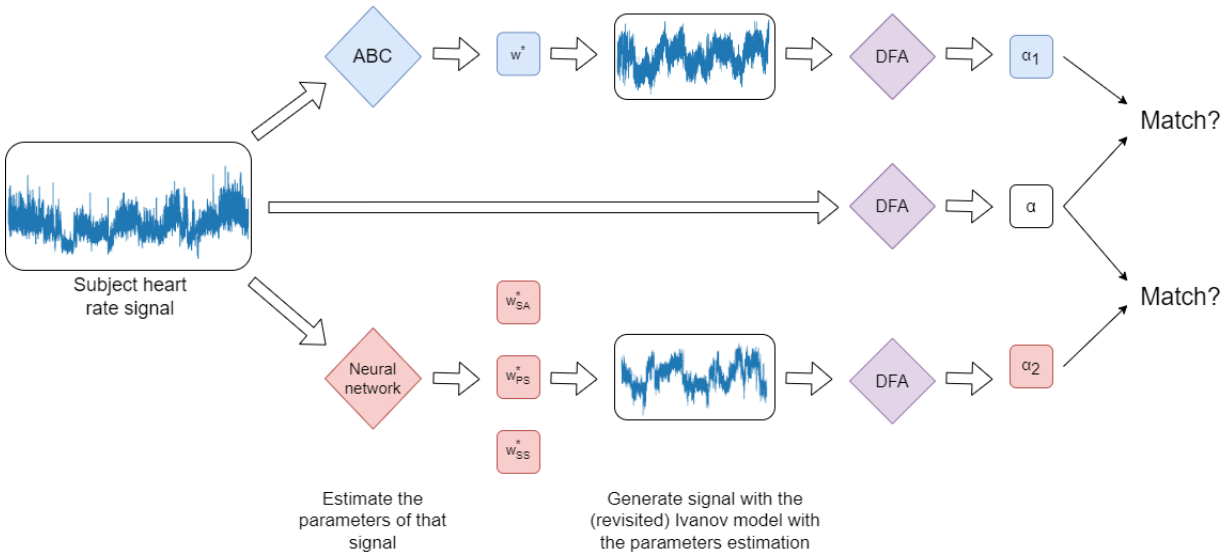


Figure 3.39: Methodology for assessing the Ivanov model's capacity to replicate the fractal signal of each subject, utilizing the parameter estimations obtained from the ABC method for the revisited model (see section 3.3.4) and the neural network.

Visual aspect An initial comparison can be made by examining the visual characteristics of the time series. In Figure 3.40, the first 20.000 data points of the time series for the first subject are displayed for each category, along with simulated signals generated using the Ivanov model and the revisited Ivanov model (see section 3.3.4). Upon visual inspection, it is apparent that the simulated signals closely resemble the original signal, demonstrating a strong visual similarity between them.

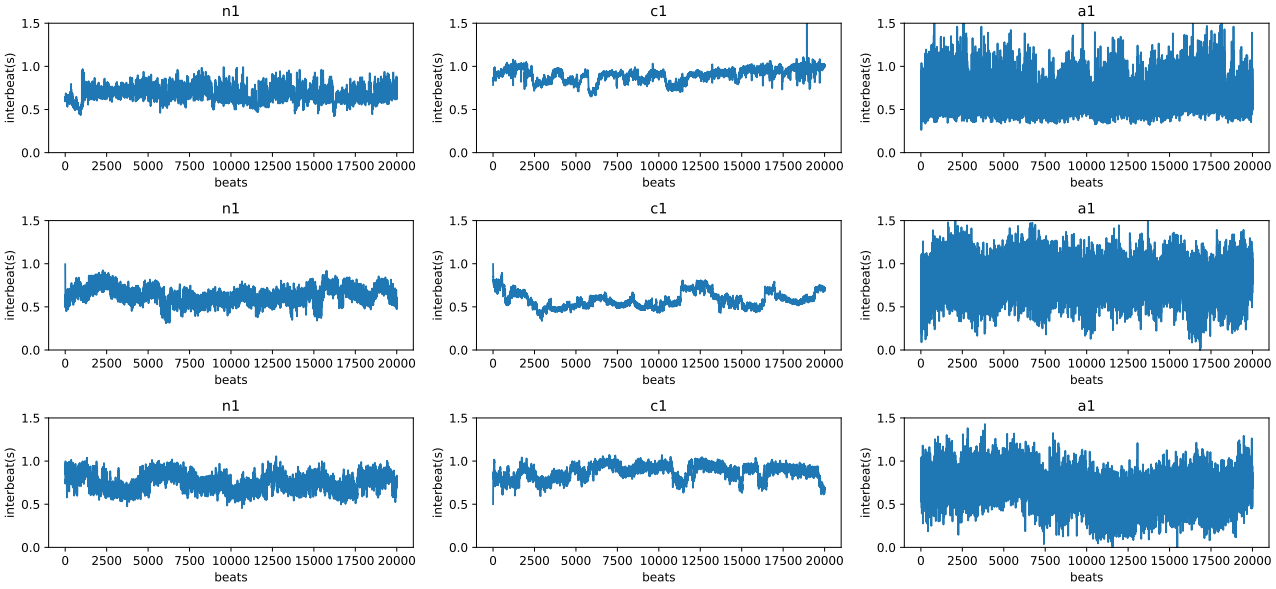


Figure 3.40: These figures depict the heart rate signals of the first subjects from each group in the second dataset (n for healthy, c for congestive heart failure, and a for atrial fibrillation). The top signals display the first 20.000 data points from the dataset. The middle signals represent signals generated using the revisited Ivanov model (section 3.3.4), where the parameter w is set to the mean value obtained from parameter estimations using the ABC method (see section 3.3). Lastly, the bottom signals correspond to signals generated using the Ivanov model with default parameters (see section 3.1.3), except for the feedback input strengths w_{SA}, w_{SS} and w_{PS}, which are set to the mean value obtained from parameter estimations using the neural network (see section 3.4). A simulation for all subjects is available in appendix D.

Detrended fluctuation analysis After visually confirming the similarity between the simulated and original signals, the next step is to assess whether the fractal pattern or its loss is accurately reproduced by the model with the fitted parameters for each subject. To accomplish this, 100 simulated signals were generated for each subject, and the resulting α coefficient was computed. Figure 3.41 presents a summary of the obtained results.

The overall results indicate that both models have successfully preserved the presence or absence of fractal dynamics. However, a few subjects, particularly those in the congestive heart failure group, have slightly deviated results, which may affect the reliability of their heart rate simulations. Nonetheless, these difference are not significant enough to dismiss the model altogether, but they indicate room for improvement.

In conclusion, this model is capable of accurately representing the heart rate of healthy subjects and can be extended to individuals with both congestive heart failure and atrial fibrillation.

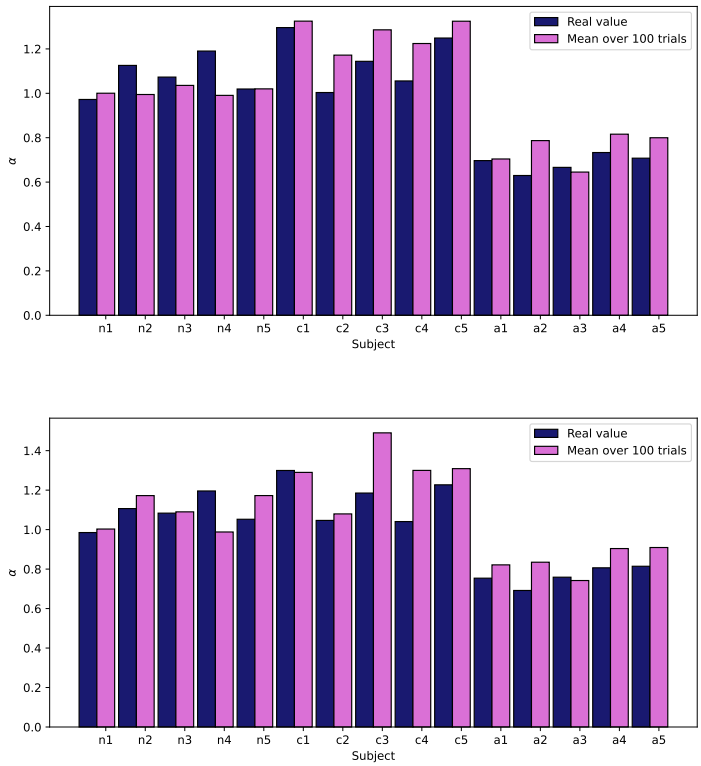


Figure 3.41: This figure compares the α coefficient obtained using the DFA method for the real heart rate signals of the subjects in the second dataset (see section 1.3.2) with the simulated signals generated using the (revisited) Ivanov model. The analysis considered scales ranging from 10 to 500, and the time series consisted of 20.000 points each. For the real signals, the DFA method was applied to the first 20.000 points. For the simulations, 100 signals were generated for each subject using parameters randomly selected from the estimations made specifically for that subject within the (revisited) Ivanov model. The value displayed represents the mean α coefficient computed across all these simulated signals. The top figure presents the results obtained using the revisited Ivanov model and the parameter values obtained through ABC (see section 3.3), while the bottom figure illustrates the results obtained with the Ivanov model and the parameter values obtained through the neural network.

3.5.3 Improvement Suggestions

Considering more parameters There are two potential strategies for improving the current model. Firstly, it may be worthwhile to investigate the preferred levels bounds of the Ivanov model, which were not previously considered in the fitting process as it was not deemed impactful on the fractal dynamic. However, since subjects with congestive heart failure have smaller amplitude compared to healthy

subjects, adjusting the bounds and expanding or reducing the potential values for the preferred levels may have a greater impact in this group. This approach could be explored using the ABC method on the revisited model initially.

To evaluate the feasibility of this suggestion, a similar approach to the previous section was employed for DFA with the parameter estimations obtained earlier using ABC. The only difference is that the signals generated for the estimation assessment now have preferred level bounds based on the values observed in the specific subject under consideration for the α coefficient comparison.

The results, depicted in Figure 3.42, indicate that this proposition holds merit and shows promising potential. The improvements achieved with subjects who initially had poor results, such as n4, c3, and c4, are substantial. However, this preliminary experiment merely confirms the usefulness of adapting the preferred level bounds based on the heart rate signal values during parameter estimation. To conduct a more comprehensive analysis, the entire process described in Figure 3.17 must be repeated, taking into account the dynamic adjustment of the preferred level bounds. The poor results observed for n1 and the atrial fibrillation group highlight the necessity of undertaking this complete process to obtain a deeper understanding of the parameter estimation and model fitting.

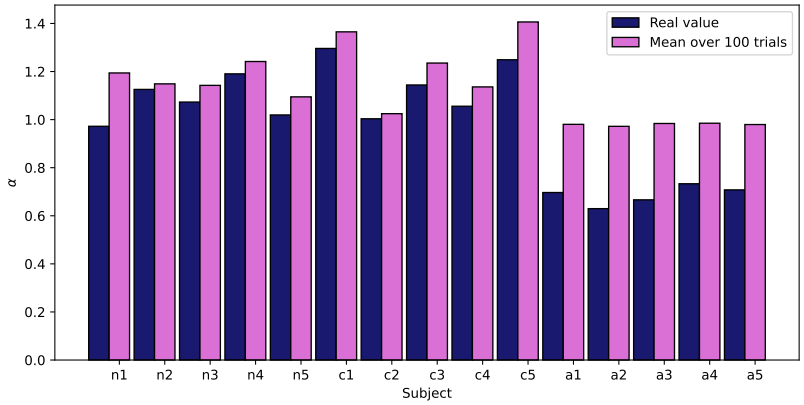


Figure 3.42: This figure compares the α coefficient obtained using the DFA method for the real heart rate signals of the subjects in the second dataset (see section 1.3.2) with the simulated signals generated using the revisited Ivanov model. The analysis considered scales ranging from 10 to 500, and the time series consisted of 20.000 points each. For the real signals, the DFA method was applied to the first 20.000 points. For the simulations, 100 signals were generated for each subject using parameters randomly selected from the estimations made specifically for that subject within the revisited Ivanov model. Furthermore, the bounds of the preferred level are adapted to the subject considered.

In a broader sense, it is possible to consider fitting all the parameters of the Ivanov model for all subjects. However, it is important to recognize that incorporating more parameters significantly increases the complexity of parameter estimation, potentially leading to a decrease in precision.

Another suggestion for incorporating new parameters in the model or parameter estimation is to introduce additional components that account for other factors influencing the heart rate, such as blood pressure or exercise.

LSTM embedding Secondly, an alternative embedding technique could be applied in the fitting of parameters with the neural network. The current choice of CNN for embedding was motivated by its ability to recognize various patterns in time series data. However, an alternative approach worth considering is the use of LSTMs (Long Short-Term Memory). LSTMs possess the capacity to capture important information from a sequence while disregarding less significant details. Given the long-range correlation exhibited by fractal time series, LSTMs could prove to be particularly advantageous in this context.

3.5.4 Results Interpretation

The outcomes of these parameter fittings indicate that in the case of congestive heart failure, the influence of the sympathetic and parasympathetic systems appears to diminish significantly. Conversely, for atrial fibrillation, there is a notable increase in the impact of these systems accompanied by a diminished significance in the Ivanov model of the sinoatrial (SA) node, as indicated by its exceptionally high variance for the parameter estimations. This observation can be attributed to the absence of a sinus rhythm in individuals with atrial fibrillation, which implies that the SA node no longer regulates the heart rate. These findings highlight the distinct physiological mechanisms underlying these two cardiac conditions, shedding light on the potential role of autonomic regulation in their pathophysiology.

According to Dr. Philippe Marcelle, a cardiologist at "CHC asbl Liège", and Dr. Sebastien Robinet, a cardiologist at "CHU Uliège", these results are consistent with the existing knowledge about congestive heart failure and atrial fibrillation.

According to Dr. Robinet, these findings align with previous studies conducted on heart rate variability [31], which is a closely related area of research. He emphasizes that the results obtained for congestive heart failure closely correlate with existing prognostic indicators. However, he notes that these results may be less applicable to atrial fibrillation due to its absence of a sinus rhythm. While still meaningful, the relevance of these findings in the context of atrial fibrillation is somewhat diminished.

Dr. Marcelle also highlights the progress made in recent years in understanding these diseases, particularly the increasing recognition of subgroups within congestive heart failure. Dr. Marcelle suggests that while the overall findings are accurate, further exploration of these subgroups would be valuable to ensure the applicability of the results to all cases, considering the heterogeneous nature of congestive heart failure. Regarding atrial fibrillation, Dr. Marcelle recommends considering additional factors such as the health of the atrioventricular node. Overall, while the model demonstrates its effectiveness, Dr. Marcelle also highlights the importance of physiological justifications for the number of feedback inputs for the parasympathetic and sympathetic systems.

This insight suggests new opportunities for future research, indicating that a more detailed differentiation of the signals based on disease subgroups could be conducted to observe any potential variations.

3.6 Conclusion

It has been demonstrated that the Ivanov model is an accurate model for representing the fractal pattern of the heart rate in healthy individuals and can be extended to diseased subjects.

Several methods were explored to adapt the Ivanov model for individuals with congestive heart failure and atrial fibrillation by using parameter estimation. Among these methods, only neural networks (section 3.4) demonstrated reliability, while the ABC method (section 3.3) was applicable only to a revised version of the model (section 3.3.4). In contrast, maximum likelihood estimation (section 3.2) proved to be highly inaccurate and unsuitable for this particular case.

The parameter estimations resulting from these methods allowed for differentiation between the three groups (healthy, congestive heart failure, and atrial fibrillation). Furthermore, the results demonstrated accuracy, as the estimations successfully replicated the correct fractal dynamics of the signals (section 3.5).

Nevertheless, perfect accuracy was not achieved for all subjects, with slight differences observed in the fractal pattern. This indicates potential avenues for further improvement and exploration of the model (section 3.5.3).

Chapter 4

Going Beyond: an Introduction to Generalized Linear Model

4.1 GLM overview

In this project, we focused on analyzing a stochastic model specifically designed for modeling the heart rate. However, the fractal dynamics observed in the heart rate signal can also be found in many other physiological signals. It would be intriguing to explore the possibility of extending this model to a more generalized framework that could be applied to model various fractal signals. One potential avenue for such an extension is the use of generalized linear models. In the following section, we provide a brief overview of generalized linear models and their potential for heart rate modeling.

A Generalized linear model (GLM) is an extension of the linear regression model [27, 30]. Unlike the linear regression model, GLMs allow for a wider range of probability distributions, making it more flexible. The linear regression model assumes that the conditional expectation of the response variable Y is a linear function of the predictor variables:

$$E(Y|\mathbf{X}) = \mathbf{X}^T\beta.$$

The linear regression model is based on a normal distribution assumption, so that

$$Y = \mathbf{X}^T\beta + \epsilon$$

with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Alternatively, we can say that

$$Y|\mathbf{X} \sim \mathcal{N}(\mu(\mathbf{X}), \sigma^2 I)$$

with $\mu(\mathbf{X}) = \mathbf{X}^T\beta$.

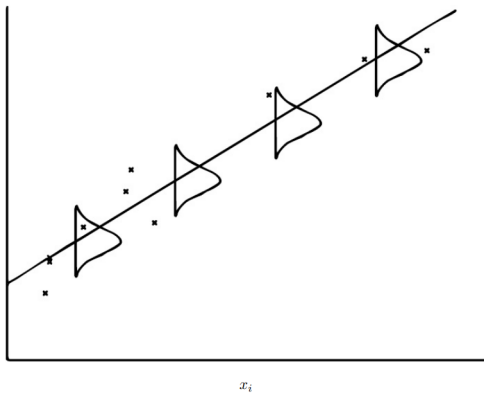


Figure 4.1: Linear regression schematic representation. *Figure from ref [27]*

In contrast to linear regression, the generalized linear model assumes that $Y|\mathbf{X}$ follows a distribution from the exponential family (more details in appendix E). Moreover, it assumes that a function of the mean is a linear combination of the predictors:

$$G(\mu(\mathbf{X})) = \mathbf{X}^T\beta$$

with G being called the *link function*.

GLMs consist of three components [37, 36]:

1. **Random component:** It refers to the probability distribution that describes the variation in the response variable.

$$Y \sim \text{distribution from the exponential family.}$$

2. **Linear predictor:** It is a linear combination of the predictor variables used to model the relationship between the predictor variables and the response variable. The linear predictor is specified as:

$$\eta = G(\mu) = \boldsymbol{X\beta}.$$

3. **Link function:** As mentioned earlier, it is a function used to transform the response variable to a linear combination of the predictor variables. Table 4.1 shows link functions for several distributions. The link function must satisfy the following requirements:

   - it must be continuously differentiable,
   - it must be strictly increasing,
   - the image must be in $\mathbb{R}$ (and thus must be invertible).

   In the case of the canonical exponential family, we can use the *canonical link*, which is obtained as:

$$g(\mu) = \theta = (b')^{-1}(\mu).$$

   In appendix E.1, we will discuss the intuition behind this formula.

Distribution	$f(y)$	Canonical link
Bernouilli	$\mu^y(1-\mu)^{1-y}$	$\eta = \log(\frac{\mu}{1-\mu})$
Poisson	$\frac{\mu^y}{y!}e^{-\mu}$	$\eta = \log(\mu)$
Exponential	$\frac{1}{\mu}\exp(-\frac{y}{\mu})$	$\eta = \frac{1}{\mu}$
Gamma	$\frac{1}{\Gamma(\phi)}(\frac{\phi}{\mu})^\phi \exp(-\frac{\phi y}{\mu})y^{\phi-1}$	$\eta = \frac{1}{\mu}$
Normal	$\frac{\exp[-(y-\mu)^2/(2\phi)^2]}{\sqrt{2\pi\phi}}$	$\eta = \mu$
Inverse Gaussian	$\sqrt{\frac{\lambda}{2\pi x^3}}\exp(-\frac{\lambda(x-\mu)^2}{2\mu^2 x})$	$\eta = \frac{1}{\mu^2}$

Table 4.1: Example of distributions from the exponential family

4.2 GLMs on Time Series

After reviewing the principles behind GLMs and the general overview of the method, we can proceed to model the heart rate of the second dataset (described in section 1.3.2) by estimating the parameters $\boldsymbol{\beta}$. To predict the interbeat for beat n, the previous $n-t$ interbeat observations will be used as the input variables $\mathbf{X}$. It should be noted that since fractals show long-range correlation, the value of t must be chosen large enough.

For time series, we can consider two cases for the GLMs:

1. keeping the signal unchanged,

2. converting the signal into a count time series.

4.2.1 Implementation

The fitting of the parameters $\boldsymbol{\beta}$ for GLMs was implemented using the `GLM` module from the `statsmodels.api` library [34]. This module enables the fitting of the parameters based on given observations and their corresponding response variable Y, along with the distribution considered.

For a given heart rate signal, we ignore the first t points and use all the remaining points as the Y values. The preceding t points in the signal are the observations for that given value.

After the fitting is completed, we can simulate a signal using the following steps:

1. Take the first t points of the original heart rate signal. These points will be the starting observations for the simulation;

2. Compute the scale of the exponential distribution using the `scale` attribute from the `GLM.fit()` method;

3. Compute the expected value μ with $\mu(\mathbf{X}) = G^{-1}(\mathbf{X}^T\boldsymbol{\beta})$;

4. Generate a new point randomly using the chosen exponential distribution;

5. Add the new point to the observations and remove the furthest point in the observations;

6. Go back to step 3 and generate a new point.

4.2.2 GLMs on the Heart Rate Signal

4.2.2.1 Normal Distribution

The first distribution that we considered is the normal distribution, which implies a simple linear regression model. We set $t = 1000$ for this model.

Fitting the GLM on the signal did not provide accurate results. As shown in Figure 4.2, a visual inspection reveals that the generated signal is significantly different from the signal on which the parameters were fitted. Nevertheless, this does not necessarily mean that the model is useless and should be discarded. Interestingly, a fractal analysis using the DFA method shows that even though the amplitudes of the signals are quite different, the fractal pattern of the signals is actually reproduced (see Figure 4.3). While this project will not undertake any further analysis, this result is encouraging with regards to the potential of GLMs for modelling heart rate signals.

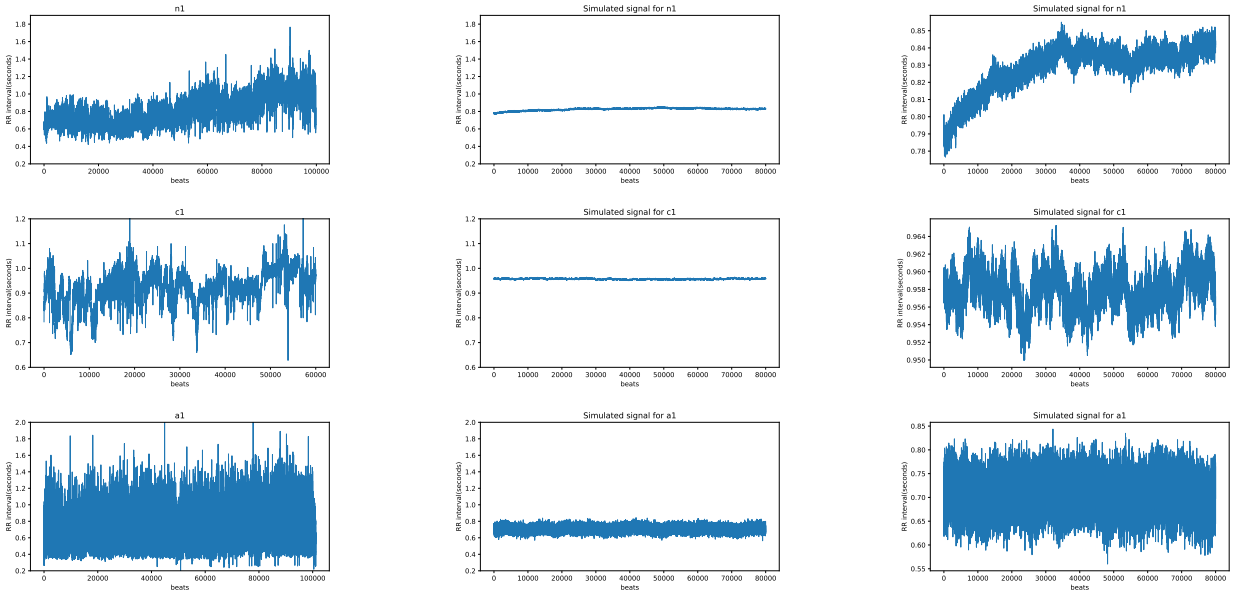


Figure 4.2: The figure shows heart rate signals. The left panel displays signals from the first subject of each group (n for healthy, a for atrial fibrillation and c for congestive heart failure) in the second dataset (described in section 1.3.2). The middle panel shows a generated signal with parameters fitted with GLM using a normal distribution on the signal of the subject with the same scale as the left panel. The right panel shows a signal generated by simulation of GLM with a different scale.

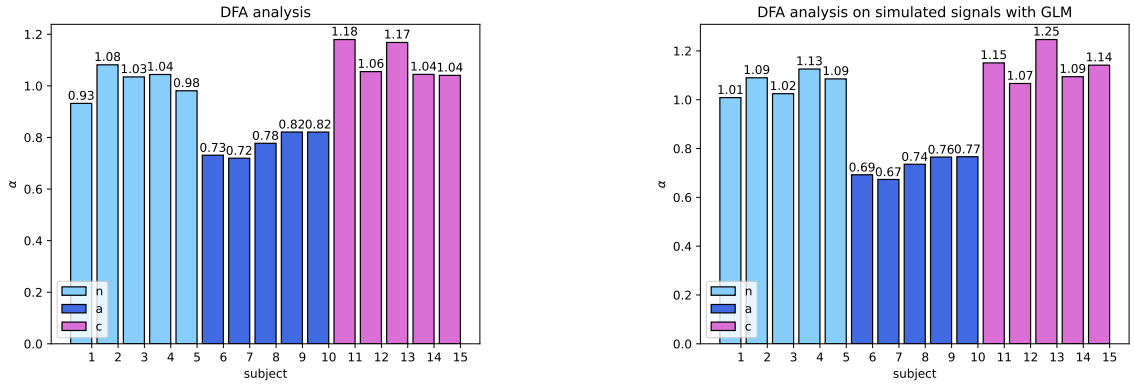


Figure 4.3: These bar plots illustrate the α coefficient for each subject, obtained from the DFA method with scales ranging from 10 to 1000. The first five bars correspond to the healthy subjects, the middle five represent the subjects with atrial fibrillation, and the last five bars correspond to the subjects with congestive heart failure. The plot on the left shows DFA performed on the real signals from the second dataset, while the one on the right shows DFA conducted on the signals generated using the parameters fitted with GLMs on the subjects of this dataset.

4.2.2.2 Inverse Gaussian Distribution

The inverse Gaussian distribution is the second distribution that is being considered. This choice was motivated by the fact that it was previously used to build a statistical model of the heart rate, which shares some similarities with a GLM [3].

Similar to the normal distribution, fitting GLMs with the inverse Gaussian distribution resulted in signals with very low amplitude compared to the original signals. However, unlike the normal distribution, the inverse Gaussian distribution failed to replicate the fractal pattern of each signal, and only produced signals that resemble white noise (see Figure 4.4). These results do not necessarily

indicate that the inverse Gaussian distribution should be rejected, given that it has been previously used for heart rate modeling. However, if one were to further explore the use of GLMs for heart rate modeling, it would be prudent to first investigate the use of the normal distribution.

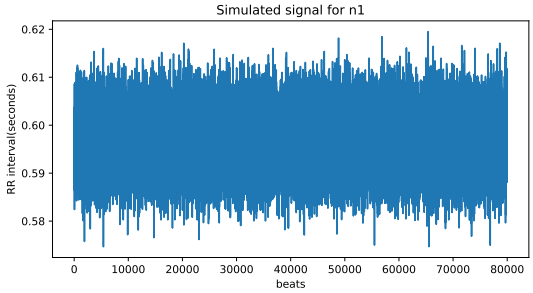


Figure 4.4: Simulation of heart rate signal for subject n1 (healthy) from the second dataset (see section 1.3.2) using parameters fitted through GLM with the inverse Gaussian distribution.

4.2.2.3 GLMs on Count Time Series

An alternative approach to consider is transforming the heart rate time series into count time series [7, 10, 25]. This involves creating time bins of a specific size in which the number of beats will be recorded. The choice of distribution to use will depend on the length of the bins:

- for small time bins, the count will either be 1 or 0, and a Bernoulli distribution should be used,
- for larger time bins, the count can be greater than 1, and a Poisson distribution should be used.

However, these approaches are not analyzed in this work. In the case of the Bernoulli distribution, a large number of beats needs to be considered, which means that an extremely large number of observations (time bins) are required, which requires powerful computational tools, and therefore, it wasn't considered in this project.

Although the Poisson distribution is more feasible, the fractal dynamics of the heart rate cannot be analyzed since we have no way of knowing how the beats are distributed within the bins. This causes us to lose the fractal properties of the heart rate that we aim to analyze in this work. However, this does not necessarily mean that count time series should be discarded altogether, and further analysis may lead to potential models with great promise.

4.3 Conclusion

We have only scratched the surface of the potential of GLMs for modelling the heart rate signal. Despite the inaccuracy of the results obtained with the normal distribution, the fact that a fractal pattern similar to the original signal was reproduced shows the promising potential of this approach.

Chapter 5

Conclusion

The objective of this project was to gain a deeper comprehension of the fractal nature of biological signals and to construct a model that could elucidate the elements responsible for this fractal pattern in such signals. Additionally, the project aimed to examine how these components may change in response to diseases.

The study primarily centers on the heart rate signal and seeks to investigate the influence of three principal components, namely the sinoatrial node (SA), the sympathetic signal (SS), and the parasympathetic signal (PS), on its variability. The investigation specifically compares three groups of subjects, healthy individuals, subjects with congestive heart failure and individuals afflicted with atrial fibrillation.

The initial focus lies in identifying the presence of a fractal pattern in the signals and conducting preliminary observations. The DFA method was identified as a suitable technique to identify the fractal pattern. The findings suggest that individuals with atrial fibrillation exhibit a loss of the fractal pattern at smaller scales, indicating its potential as a diagnostic tool. Although a slight distinction was observed between healthy subjects and those with congestive heart failure, it is not substantial enough to definitively differentiate between the two groups.

This project centers around the utilization of the Ivanov model, an existing stochastic model introduced by *Ivanov et al.* [21]. The Ivanov model is based on a random walk process that incorporates inputs influenced by the SA, SS, and PS components. To align the model with real datasets, the strengths of these inputs were calibrated using three distinct approaches: maximum likelihood estimation, approximate Bayesian computation (ABC), and neural networks. Among these methods, the neural network emerged as a reliable estimator, while the ABC method demonstrated reasonable accuracy when applied to a slightly adjusted version of the model.

The results of the model fitting were found to be intriguing. In the case of individuals with congestive heart failure, there was a noticeable reduction in the strength of the inputs, indicating that the heart rate is less influenced by the various systems. On the other hand, individuals with atrial fibrillation demonstrated an opposite reaction, with a significant increase in the strength of the inputs.

The results obtained in this study demonstrate a significant level of accuracy in modeling the heart rate and its fractal dynamics for both healthy subjects and those with congestive heart failure and atrial fibrillation. Although the model did not achieve total accuracy, there are various potential avenues for future work, such as exploring the possibility of fixing custom parameters, which were fixed in this study, or considering the potential of using LSTMs to better integrate the long-range dependency of the fractal dynamics in the parameters fitting.

Although the Ivanov model is specifically designed for heart rate signals, fractal dynamics can also be observed in other biological signals. Considering the model's ability to accurately reproduce this fractal dynamic, it could be interesting to generalize its application. In this regard, the potential of using generalized linear models (GLM) to model the heart rate was briefly investigated. Despite the modest accuracy of the generated signals, some of them successfully captured the fractal pattern observed in the original subjects. This finding highlights promising opportunities for further investigation into the fractal dynamics of biological signals.

The fractal pattern observed in biological signals is a complex structure that we have yet to fully

comprehend. This study has provided insight into the potential of modeling the fractal pattern of the heart rate, examining the elements that influence it and how they can evolve with diseases, ultimately affecting the fractal pattern. The field of research in this area is extensive, and the model developed in this study could be extended to many other horizons.

References

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a Convolutional Neural Network". In: *IEEE* (2017).

[2] Xianghong Arakaki1 et al. "The connection between heart rate variablility (HRV), neurological health, and cognition: a literature review". In: *Frontiers in Neuroscience* 17 (2023).

[3] Riccardo Barbieri et al. "A point-process model of human heartbeat intervals: new definitions of heart rate and heart rate variability". In: *Am J Physiol Heart Circ Physiol* 288 (2005), pp. 424–435.

[4] Gustavo E.A.P.A. Batista, Xiaoyue Wang, and Eamonn J. Keogh. "A Complexity-Invariant Distance Measure for Time Series". In: *Proceedings of the 2011 SIAM International Conference on Data Mining* (2013). Society for Industrial and Applied Mathematics, pp. 699–710.

[5] Richard Blender and Klaus Fraedrich. "Long time memory in global warming simulations". In: *GEOPHYSICAL RESEARCH LETTERS* 30.14 (2003).

[6] Anita Boardman et al. "A study on the optimum order of autoregressive models for heart rate variability". In: *Physiology measures* 23 (2002), pp. 325–336.

[7] Nicholas Bosowski, Vinay Ingle, and Dimitris Manolakis. "Generalized Linear models for count time series". In: *ICASSP* (2017), pp. 4272–4276.

[8] Hui Ding et al. "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures". In: *Proceedings of the VLDB Endowment* 1.2 (2008), pp. 1542–1552.

[9] Keaogh Eamonn. *Introduction to Time Series Mining*. `http://didawiki.cli.di.unipi.it/lib/exe/fetch.php/dm/time_series_from_keogh_tutorial.pdf`. Accessed: 06-05-2023.

[10] Uiri Eden. *Generalized linear models*. Presented at the Methods in Computational Neuroscience. 2016.

[11] Ary L. Goldberger, C.-K. Peng, and Lewis A. Lipsitz. "What is physiologic complexity and how does it change with aging and disease?" In: *Neurobiology of Aging* 23 (2002), pp. 23–26.

[12] L. Goldberger et al. "Fractal dynamics in physiology: Alterations with disease and aging". In: *PNAS* 99 (2002), pp. 2466–2472.

[13] S. Halvin et al. "Fractals in Biology and Medicine". In: *Chaos, Solitons Fractals* 6 (1995), pp. 171–201.

[14] Jeffrey M. Hausdorff and C.K. Peng. "Multiscaled randomness: A possible source of 1/f noise in biology". In: *Physical Review E.* 54.2 (1996), pp. 2154–2157.

[15] Jeffrey M. Hausdorff et al. "When human walking becomes random walking: fractal analysis and modeling of gait rhythm fluctuations". In: *Physica A.* 302 (2001), pp. 138–147.

[16] Indranil Hazra, Mahesh D. Pandey, and Noldainerick Manzana. "Approximate Bayesian computation (ABC) method for estimating parameters of the gamma process using noisy data". In: *Reliability Engineering System Safety* 198 (2020).

[17] Joeri Hermans et al. "Posterior Approximations Can Be Unfaithful". In: *Transactions on Machine Learning Research* 11 (2022).

[18] K. Hu et al. “The suprachiasmatic nucleus functions beyond circadian rhythm generation”. In: *Neuroscience* 149 (2007), pp. 508–517.

[19] Kun Hu et al. “Effect of trends on detrended fluctuation analysis”. In: *PHYSICAL REVIEW E* 64.011114 (2001).

[20] Kun Hu et al. “Non-random fluctuations and multi-scale dynamics regulation of human activity”. In: *Physica A.* 337 (2004), pp. 307–318.

[21] P.C. Ivanov et al. “Stochastic feedback and the regulation of biological rythms”. In: *Europhysics letters* 43 (1998), pp. 363–368.

[22] Eamonn Keogh and Shruti Kasetty. “On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration”. In: *Data mining and knowledge discovery* 7.4 (2003), pp. 349–371.

[23] Marvin S. Keshner. “1/f noise”. In: *Proceedings of the IEEE* 70.3 (1982), pp. 212–218.

[24] Philippe Kolh. *Physiologie des Systèmes.* Course syllabus. 2021.

[25] W.K. Li. “Time Series Models Based on Generalized Linear Models: Some Further Results”. In: *Biometrics* 50 (1994), pp. 506–511.

[26] Haijun Lin et al. “A new method for heart rate prediction based on LSTM-BiLSTM-Att”. In: *Measurement* 207 (2022).

[27] James K. Lindsey. *Applying Generalized linear models.* New York: Springer, 1997.

[28] Gilles Louppe. *Lecture 5: Convolutional networks.* Course of Deep Learning given at University of Liège. 2023.

[29] Jeffrey M.Haussdorff et al. “Fractal dynamics of human gait: stability of long-range correlations in stride interval fluctuations”. In: *Journal of Applied Physiology* 80 (1996), pp. 1448–1457.

[30] Marlene Müller. “Generalized linear model”. In: *Handbook of Computational Statistics* (2004), pp. 681–709.

[31] James Nolan et al. “Prospective Study of Heart Rate Variability and Mortality in Chronic Heart Failure”. In: *Circulation* 98.15 (1998).

[32] C.-K. Peng et al. “Mosaic organization of DNA nucleotides”. In: *PHYSICAL REVIEW E* 42.2 (1994), pp. 1685–1689.

[33] C.-K. Peng et al. “Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series”. In: *Chaos* 5.1 (1995), pp. 82–7.

[34] Jonathan Pillow. *Statistical models for neural data: From GLMs to latent variables.* Presented at COSYNE. 2018.

[35] Homer R.Warner and Albert Cox. “A mathematical model of heart rate control by sympathetic and vagus efferent information”. In: *Simulation* 3.1 (1962), pp. 349–355.

[36] Philippe Rigollet. *Chapter 10: Generalized Linear Models.* `https://ocw.mit.edu/courses/18-650-statistics-for-applications-fall-2016/dff89368051a5feae72b39c6422d0752_MIT18_650F16_GLM.pdf`. Statistics for Application course given at MIT. 2016.

[37] Pierre Sacré. *Lecture slides for Priniciples of Neuroengineering.* 2022.

[38] S. A. Sisson, Y. Fan, and Mark M. Tanaka. “Sequential Monte Carlo without likelihoods”. In: *PNAS* 104.6 (2007), pp. 1760–1765.

[39] James Robert Stirling et al. “A Model of Heart Rate Kinetics in Response to Exercise”. In: *Journal of Nonlinear Mathematical Physics* 15.2 (2002), pp. 325–336.

[40] Mikael Sunnaker et al. “Approximate Bayesian computation”. In: *PLOS Computational Biology* 9.1 (2013).

[41] Kuan Tao et al. “Estimation of Heart Rate Using Regression Models and Artificial Neural Network in Middle-Aged Adults”. In: *Frontiers in Physiology* 12 (2021).

[42] Fumiharu Togo and Yoshiharu Yamamoto. "Decreased fractal component of human heart rate variability during non-REM sleep". In: *Am J Physiol Heart Circ Physiol* 280 (2000), pp. 17–21.

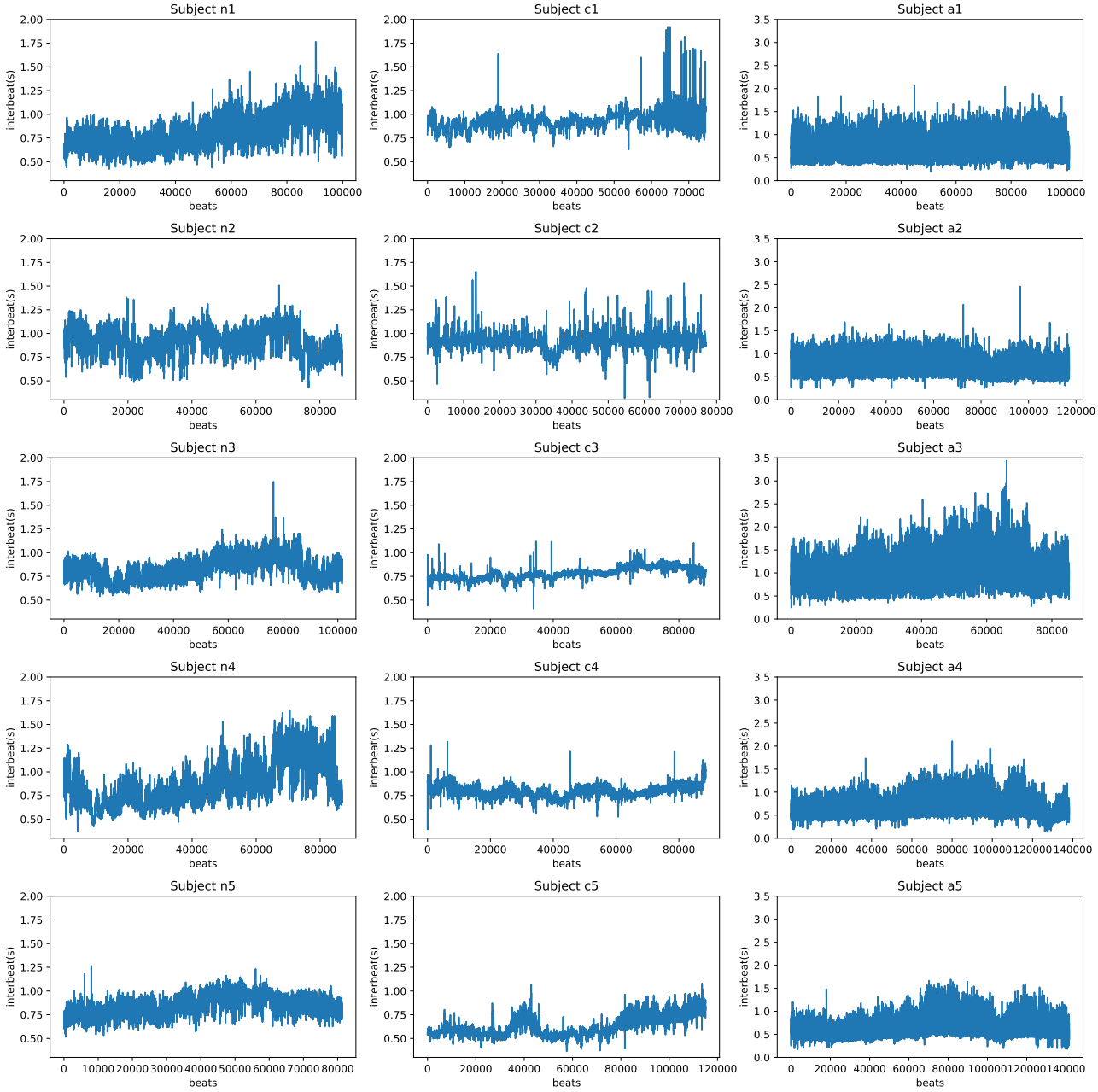
[43] Tina Toni et al. "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems". In: *Journal of the royal society* 6 (2009), pp. 187–202.

[44] Derrick TR, Bates BT, and Dufek JS. "Evaluation of time-series data sets using the Pearson product-moment correlation coefficient." In: *Medecine and Science in Sports and Exercise* 26.7 (1994), pp. 919–928.

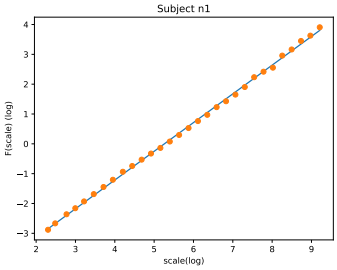
[45] Xiaoyue Wang et al. "Experimental comparison of representation methods and distance measures for time series data". In: *Data Mining and Knowledge Discovery* 26 (2013), pp. 275–306.

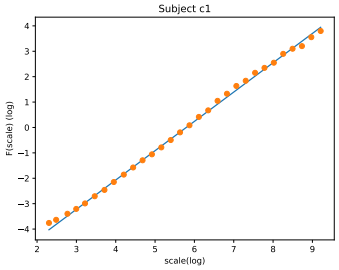
[46] Bendong Zhao et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.

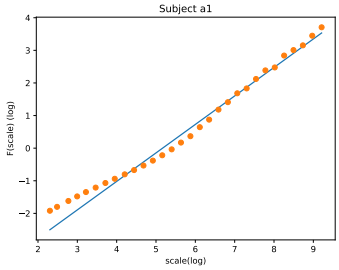
A Heart Rate Time Series of the Dataset with Healthy and Sick Patients

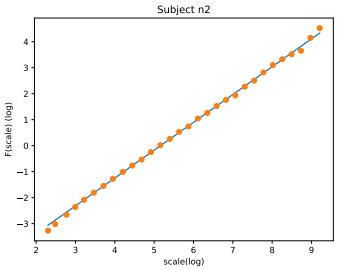


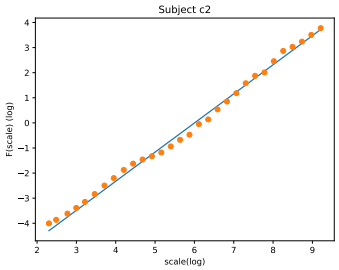
B DFA on Dataset with Healthy and Sick Patients

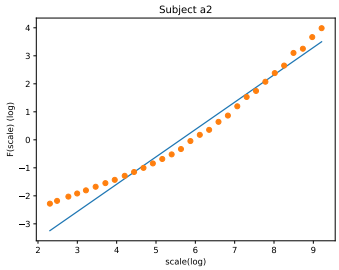


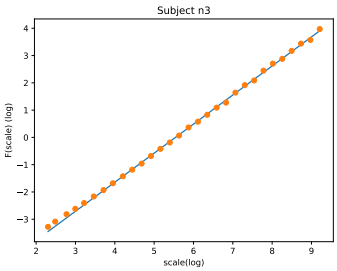


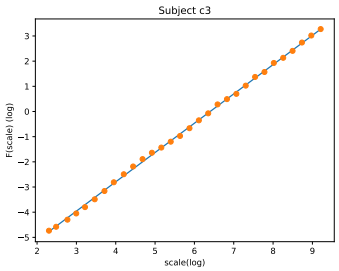


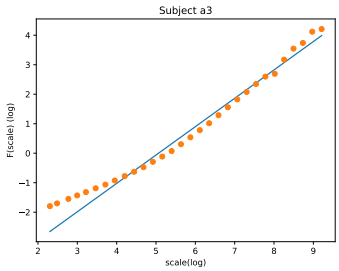


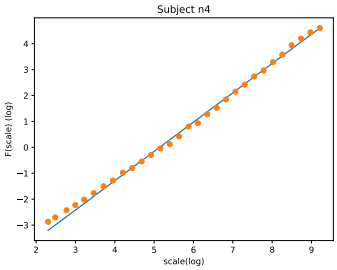


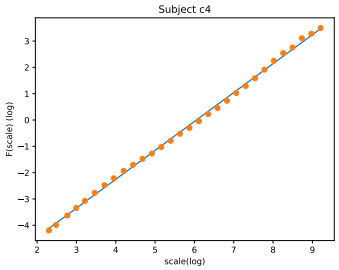


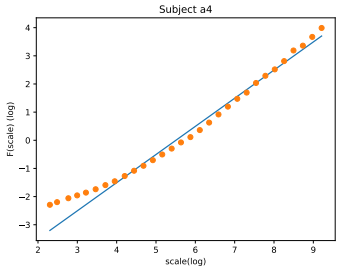


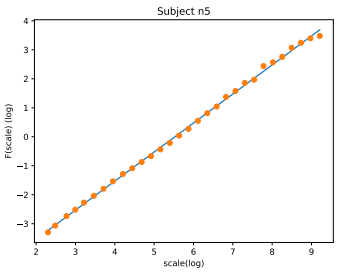


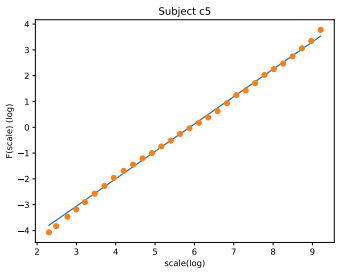


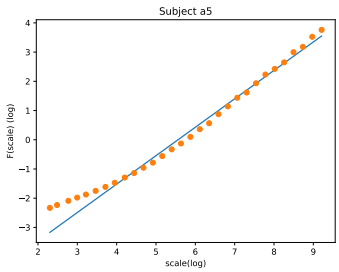




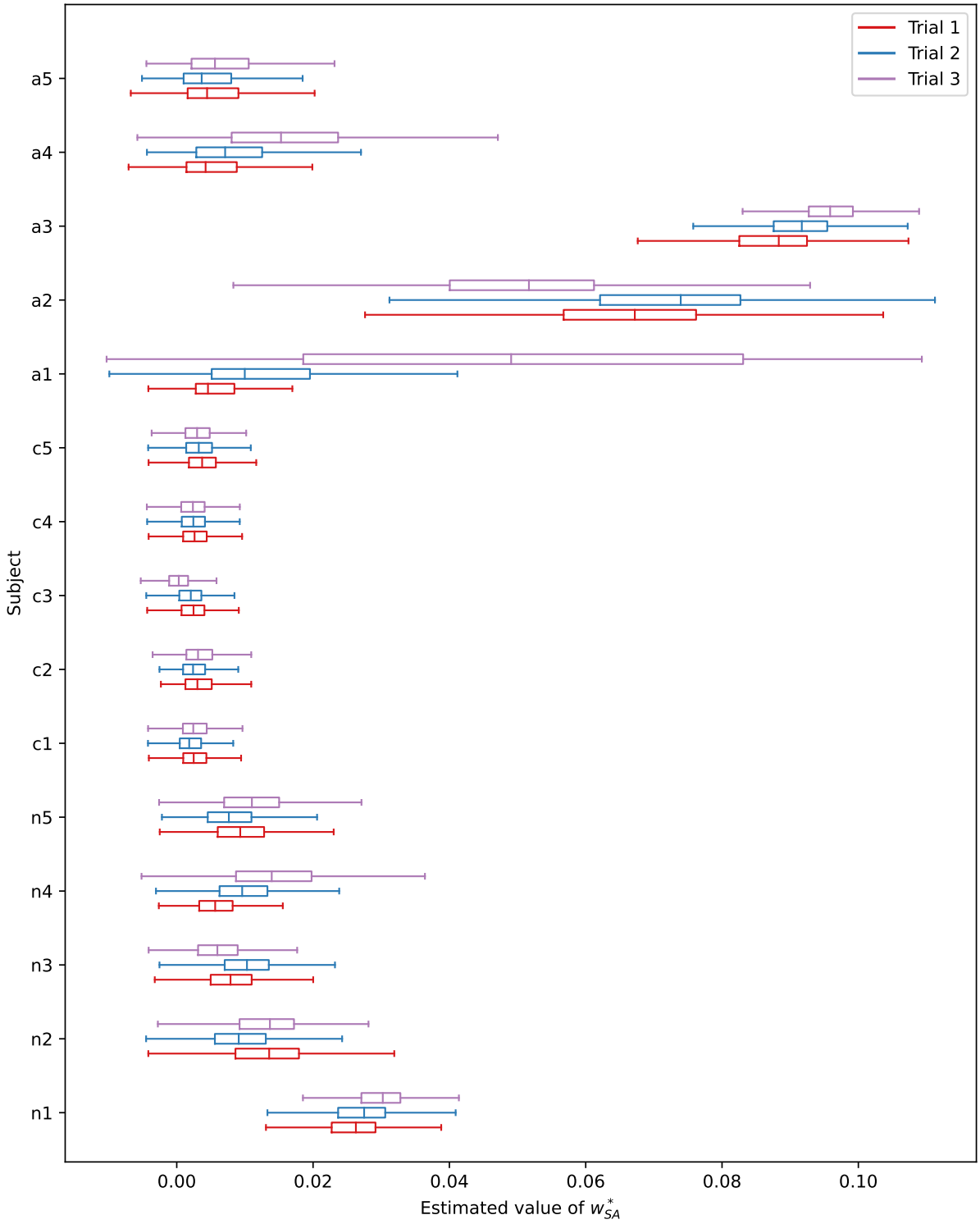


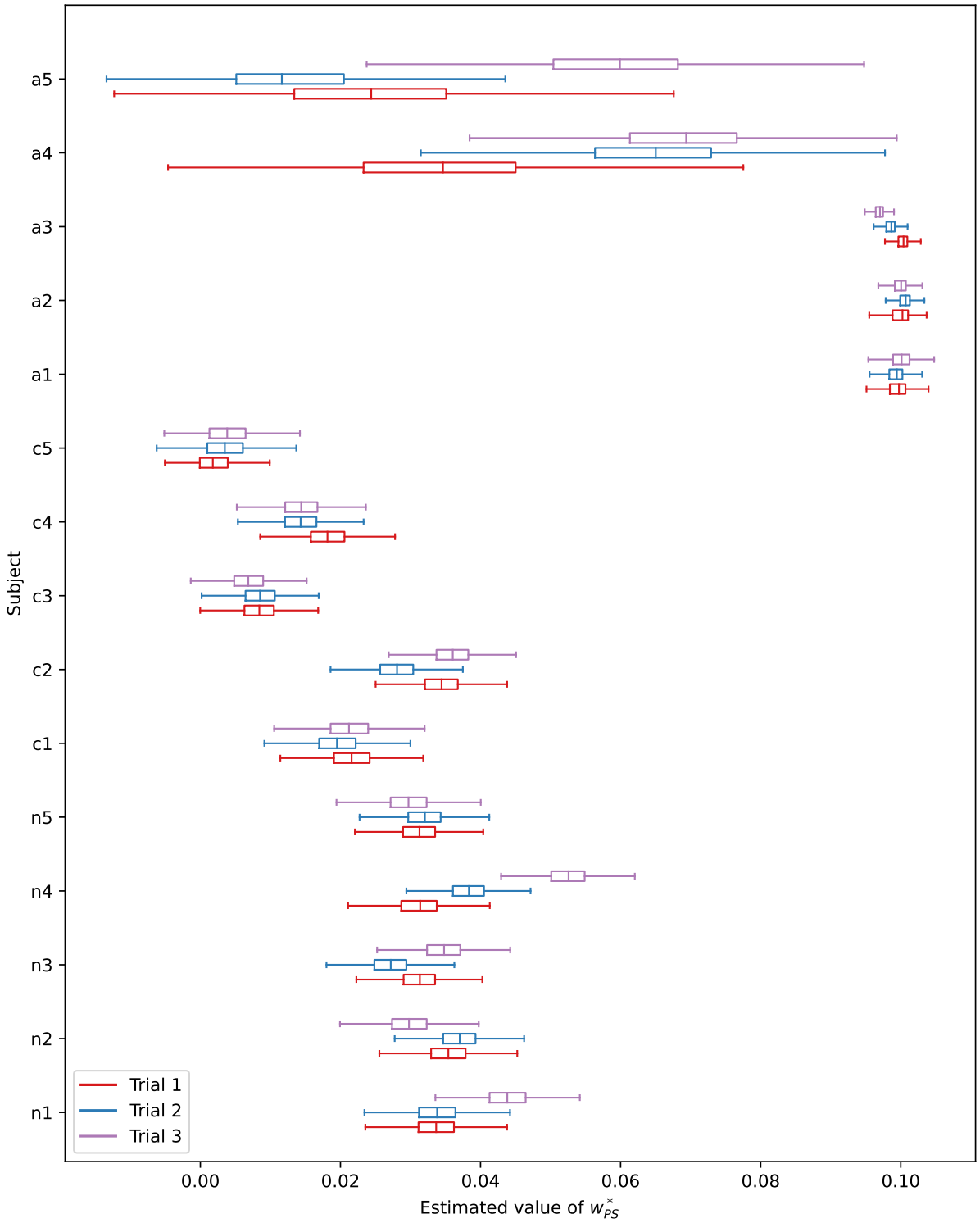


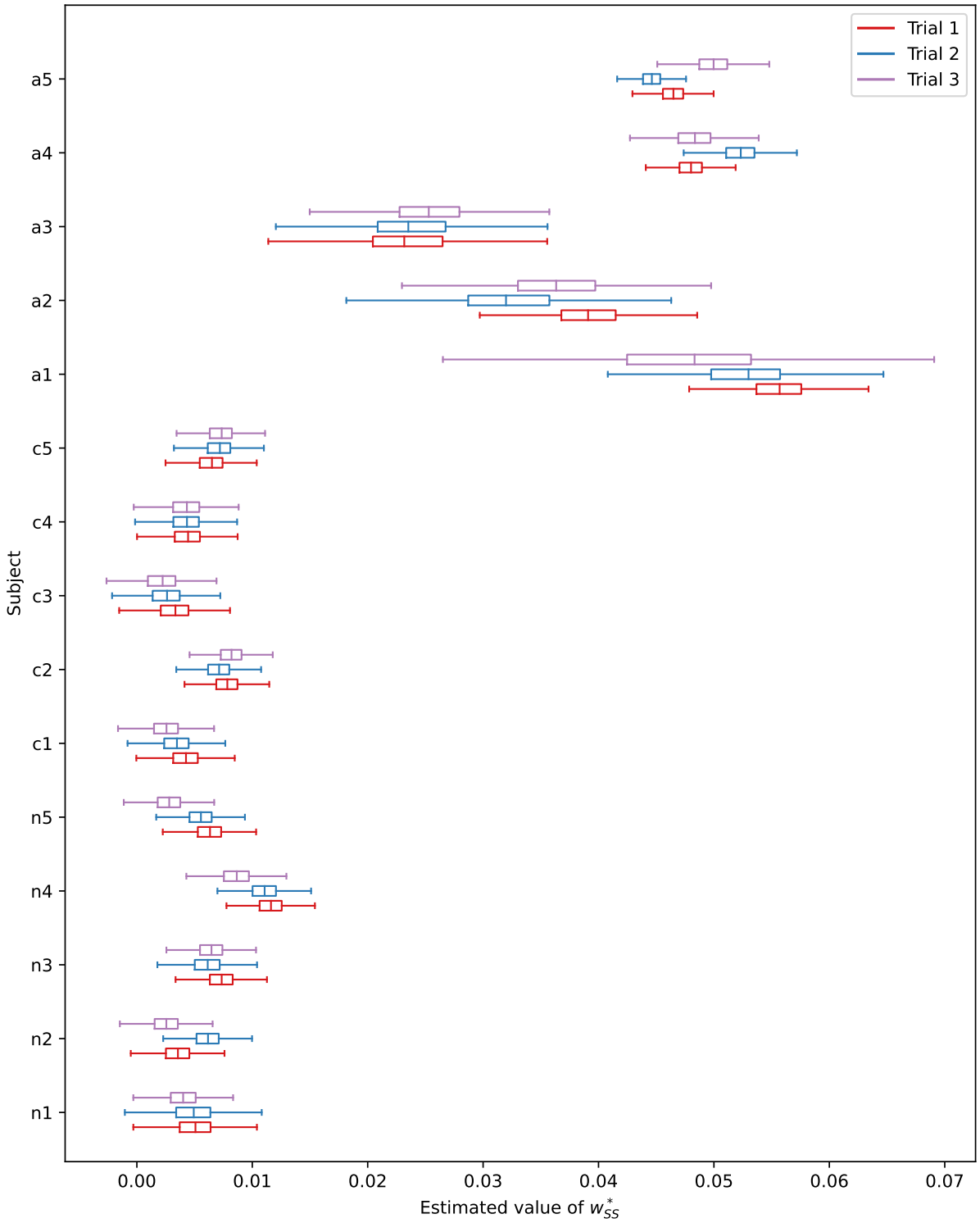




C Parameters Estimation of the Ivanov Model with Neural Networks

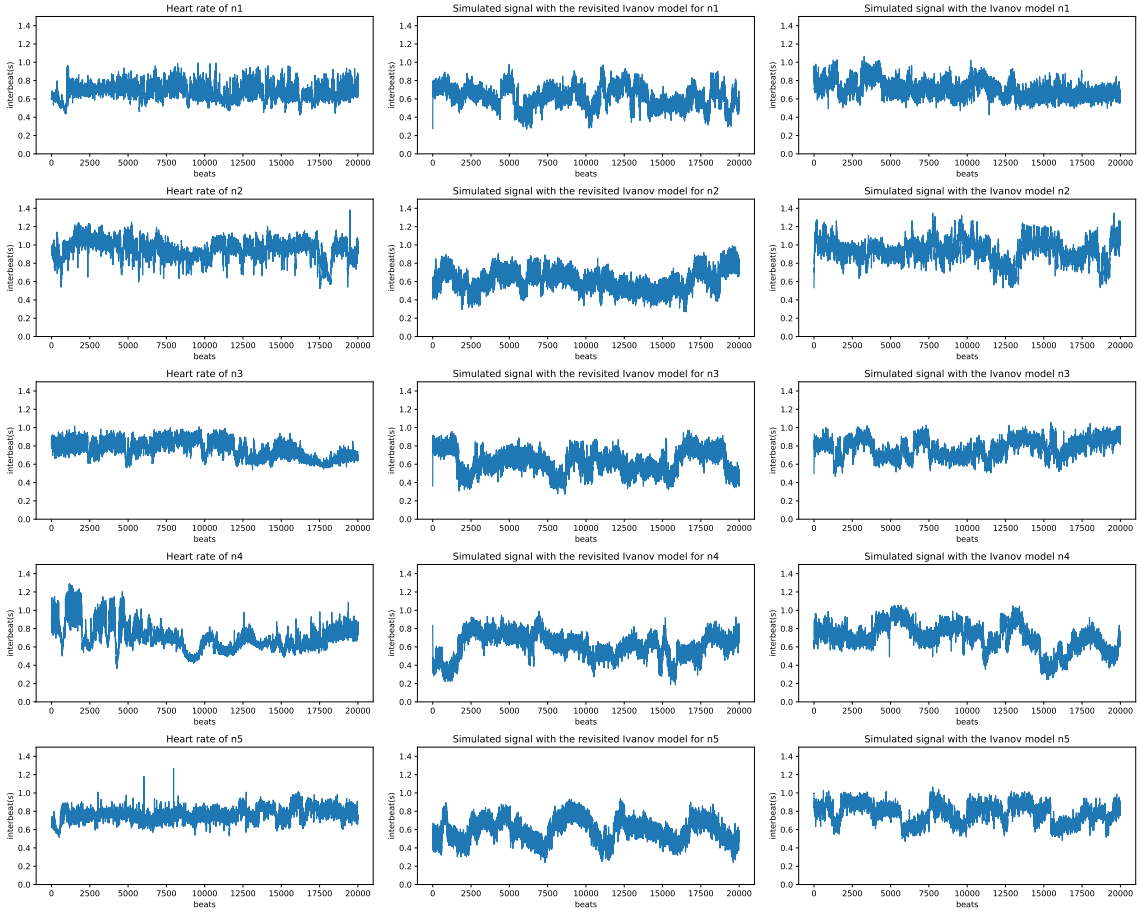



a5
a4
a3
a2
a1
c5
c4
c3
c2
c1
n5
n4
n3
n2
n1
Subject
Trial 1
Trial 2
Trial 3
0.00
0.02
0.04
0.06
0.08
0.10
Estimated value of w_{PS}^*

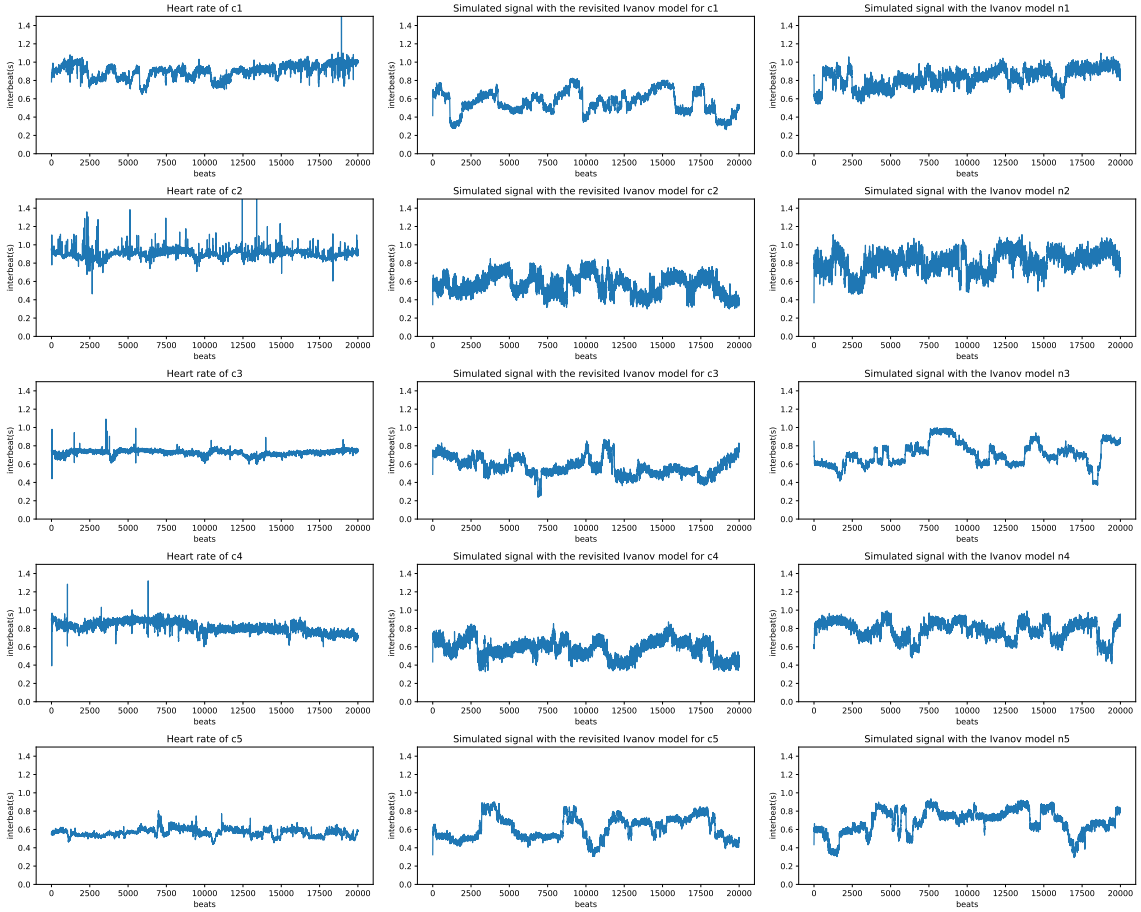

Trial 1
Trial 2
Trial 3
a5
a4
a3
a2
a1
c5
c4
c3
c2
c1
n5
n4
n3
n2
n1
Subject
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Estimated value of w^*_{SS}

D Heart Rate Signals Simulation with the (Revisited) Ivanov Model

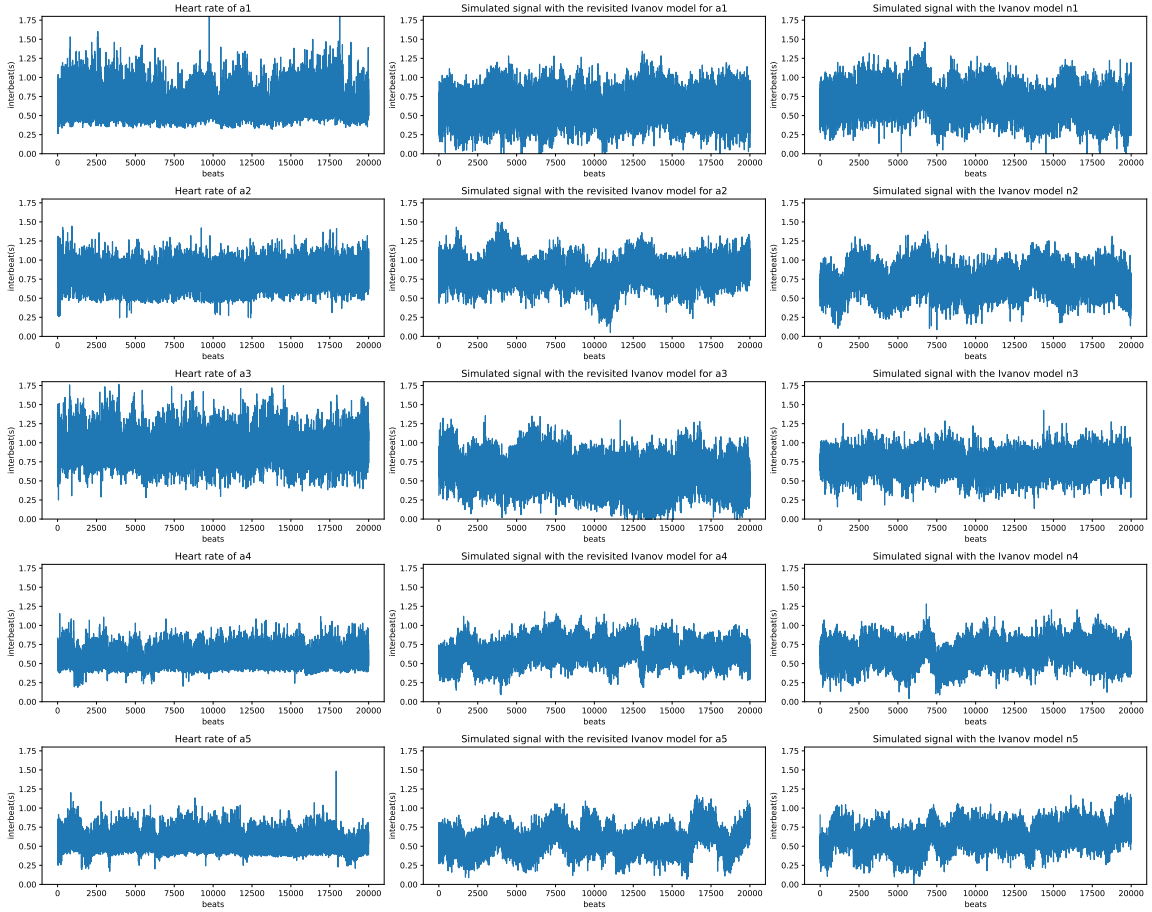
D.1 Healthy Subjects



D.2 Subjects with Congestive Heart Failure



D.3 Subjects with Atrial Fibrillation



E Exponential Family

The exponential family is the group of all distributions for which the probability (density) function is of the form:

$$f(x,\theta) = h(x)f(x)\exp[\sum_{i=1}^{k}\eta_i(\theta)T_i(x)] = h(x)\exp[\sum_{i=1}^{k}\eta_i(\theta)T_i(x) - B(\theta)].$$

Table 4.1 presents some examples of distributions from the exponential family. Although some of these distributions may not appear to be in the exponential family format, they can be transformed to the correct format.

Let's take two examples. First, let's consider the normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$ with $\theta = (\mu, \sigma^2)$

$$f(\theta, x) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-\frac{(x-\mu)^2}{2\sigma^2}) = \exp(-\frac{x^2}{2\sigma^2} - \frac{\mu^2}{2\sigma^2} + \frac{x\mu}{2\sigma^2} - \log(\sigma\sqrt{2\pi})).$$

We thus have a distribution of the exponential family with:

- $h(x) = 1$
- $T_1(x) = x^2$
- $\eta_1(\theta) = -\frac{1}{2\sigma^2}$
- $T_2(x) = x$
- $\eta_2(\theta) = -\frac{\mu}{\sigma^2}$
- $B(\theta) = \frac{\mu^2}{2\sigma^2} + \log(\sigma\sqrt{2\pi})$

Next, let's take the example of the Bernouilli distribution. Its probability mass function can be changed into

$$P(X = x) = p^x(1-p)^{1-x} = \exp(x\log(\mathrm{p}) + (1-x)\log(1-p)) = \exp(x\log(\frac{p}{1-p}) - \log(\frac{1}{1-p}))$$

with $p^x = \exp(x\log(p))$.

We thus have

- $h(x) = 1$
- $T_1(x) = x$
- $\eta_1(\theta) = \log(\frac{p}{1-p})$
- $B(\theta) = \log(\frac{1}{1-p})$

In the context of the generalized linear model, we will only consider a subset of the exponential family called the *canonical exponential family.*

$$f(y,\theta,\phi) = \exp[\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)]$$

with ϕ being the *dispersion parameter* indicating how "spread" the distribution is. θ is called the *location parameter* and indicates the position where the distribution lies.

E.1 Mean

We know for a fact that $\int f(y,\theta,\phi)dy = 1$. From there, we can derive

$$0 = \frac{\partial}{\partial\theta}\int f(y,\theta,\phi)dy = \int [\frac{\partial}{\partial\theta}\log f(y,\theta,\phi)]f(y,\theta,\phi)dy = E[\frac{\partial}{\partial\theta}l(y,\theta,\phi]$$

with $l(y,\theta,\phi) = \log f(y,\theta,\phi)$. We thus have

$$l(y,\theta,\phi) = \frac{Y\theta - b(\theta)}{a(\phi)} + c(Y,\phi).$$

Therefore:

$$\frac{\partial l}{\partial\theta} = \frac{Y - b'(\theta)}{a(\phi)}$$

and thus

$$E(\frac{\partial l}{\partial\theta}) = \frac{E(Y) - b'(\theta)}{a(\phi)} = 0$$

which leads to the final result of

$$E(Y) = \mu = b'(\theta).$$

E.2 Example of the Inverse Gaussian Distribution

We will transform the original analytical form of the inverse Gaussian distribution into an expression that resembles the canonical exponential form.

$$\begin{aligned} f(x) &= \sqrt{\tfrac{\lambda}{2\pi x^3}}\exp(-\tfrac{\lambda(x-\mu)^2}{2\mu^2 x}) \\ &= \exp(-\tfrac{\lambda(x-\mu)^2}{2\mu^2 x + \frac{1}{2}\log\frac{\lambda}{2\pi x^3}}) \\ &= \exp(-\tfrac{\lambda(x^2-2\mu x+\mu^2)}{2\mu^2 x} + \tfrac{1}{2}\log(\tfrac{\lambda}{2\pi x^3})) \\ &= \exp(-\tfrac{\lambda x}{2\mu^2} + \tfrac{\lambda}{\mu} - \tfrac{\lambda}{2x} + \tfrac{1}{2}\log(\tfrac{\lambda}{2\pi x^3})). \end{aligned}$$

We now fix $\theta = -\frac{1}{2\mu^2}$ and $\phi = \frac{1}{\lambda}$. Therefore, we have $\frac{1}{\mu} = \sqrt{-2\theta}$.

$$f(x) = \exp(\tfrac{x\theta + \sqrt{-2\theta}}{\phi} - \tfrac{1}{2x\phi} + \tfrac{1}{2}\log\tfrac{1}{2\pi x^3\phi})$$

We therefore have:

- $a(\phi) = \phi$
- $b(\theta) = -\sqrt{-2\theta}$
- $c(x,\phi) = \frac{1}{2}\log\frac{1}{2\pi x^3\phi} - \frac{1}{2x\phi}$.

For the canonical link, the constants are often removed. Therefore, the canonical link can be computed in the following way:

$$\theta = \frac{1}{\mu^2} \iff \mu = \frac{1}{\sqrt{\theta}} = b'(\theta)$$