Analysis of Active Watch data after stroke

Xi Chen

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics & Physics
Newcastle University
Newcastle upon Tyne
United Kingdom
March 2021

To my family members, Chuanzhong, Donghui and Siting.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Jian Qing Shi who leads me, guides me, inspires me and tries his best to help me during my PHD years. I am also very greatful to Prof. Janet Eyre who has given me considerable helpful advice when dealing with the problems encountered by stroke patients.

I must also extend my gratitude to all my family members, my parents and my wife; they support me, inspire me and try their best to help me as well, expecially my wife, who takes great care of me during the PHD period.

I also take this opportunnity to thank Yafeng who gave me considerable help in the first year. And I thank all of my friends who helped me during my PHD years.

Finally, I am very grateful to the School of Mathematics, Statistics and Physics for providing the necessary facilities and resources to conduct this research.

Abstract

Due to the rapid development of modern technology and the low cost of wearable devices, a very large amount of accelerometer data has been recorded and used in many different areas, particularly in medical research. However, the analysis of accelerometer data is very challenging due to its complex structure, i.e., the large noise-signal ratio, the very large amount of data and the heterogeneity in different data sets. In this thesis, we use wavelet in a functional data analysis (**FDA**) framework to analyse the data and apply this method to evaluate upper limb function after stroke, a difficult task in medical research. In addition to the commonly used features (Preece et al., 2009; Sekine et al., 1998) based on the wavelet energy preserving condition for accelerometer data under the discrete wavelet transform (**DWT**), we propose two new types of scalar features. They extract different types of information from the accelerometer data and use to predict upper limb function for stroke patients. To further investigate the 'details' based on wavelet-domain, under a Bayesian hierarchical model (**NIG-MT**), the wavelet coefficients with small values can be eliminated properly and efficiently with negligible loss from the total information. We will use the slide window approach and multivariate functional principal component analysis (**fPCA**) based on the small **DWT** tree structure. This further reduces the size of the data set and extracts the useful information from the pattern of small **DWT** tree structures in wavelet-domains. Classification and regression models are developed based on the small **DWT** tree structure. The models have been applied to distinguish between the different activities in the designed data and refine the new features in free-living data to assess the patients' upper limb function respectively.

Contents

List of Figures

List of Tables

List of Abbreviations

FDA	Functional data analysis
VM	Signal vector magnitude
CWT	Continuous wavelet transform
DWT	Discrete wavelet transform
DWPT	Discrete wavelet packet transform
NIG-MT	Normal Inverse-Gamma Markov tree model
fPCA	Functional principal component analysis
GPC	Gaussian process classifier
GMM	Gaussian Mixture Models

Chapter 1

Introduction

If a set of observed objects can be described by curves, surfaces or any mathematical structure varying over a continuum, we describe it as functional data. Functional data analysis (**FDA**), as a branch of statistics, has been widely used in different areas, e.g., medicine, meteorology and biology. Nowadays, stroke is a common worldwide serious disease causing disability (Donnan et al., 2008). Recently, in order to analyze this disease, a low cost wearable device called an AX3 has been used in this area. The accelerometer data collected from the wrist-worn sensor, AX3, is a type of functional data which can be used in the **FDA** framework. However, the analysis of accelerometer data from a stroke patient is very challenging due to its complex structure. A very large amount of the data includes noise, and there is a heterogeneity among different patients. In this thesis, an efficient functional data analysis, through wavelet-domain, will be applied to the analysis of accelerometer data.

1.1 FDA based on Wavelet-domain

In our study, the accelerometer data collected at a frequency of 100Hz. This covers the frequencies at which individual movement occurs and is observed in a one dimensional time-domain. However, different individual movements relate to different frequency information, e.g., running and writing respond to high and low frequency information in the frequency-domain. Instead of giving an analysis of the raw accelerometer data directly, a more proper way is to consider the different frequency information in the accelerometer data. How to capture the certain frequency information from the accelerometer data becomes our challenge. Regarding this problem, a powerful tool called wavelet transform (Donald B. Percival, 2000), which has been very popular in recent decades, will be used. Through the wavelet transform, the accelerometer data will be transformed to two types of **DWT** coefficients which includes the time-domain and frequency-domain respectively.

The movements can be described by accelerometer data only in the time-domain. If we want to analyze the movements at a deeper level, it is better to consider the frequency domain as well. Thus, after applying **DWT** from accelerometer data to the **DWT** coefficients, the integral characteristic in the time-domain will be separated into the particular property in both the time-domain and the frequency-domain. In other words, an analysis of the wavelet coefficients will present more valued information since it contains the information from both the time and frequency-domains.

1.2 DWT tree structure

A significant structure called the **DWT** tree structure can be found within the **DWT** coefficients. To be more specific: **(i)** The number of **DWT** coefficients at higher decomposition levels is twice the number of those at lower decomposition levels; each **DWT** coefficient (parent node) at the previous level can correspond to two **DWT** coefficients (children nodes) at the next level. In other words, the **DWT** coefficients form a tree structure which looks like a pyramid. **(ii)** Among adjacent decomposition levels, if the parent node at the previous level is far away from 0, then two children nodes tend to be far away from 0 at the next level; otherwise, two children nodes at the next level tend to be close or equal to 0. This implies that there is a strong correlation between parent node and children nodes in the whole **DWT** tree structure. This enables us to carry out an analysis of the **DWT** tree structure directly through the time and frequency-domains.

1.3 Data reduction system based on the wavelet-domain

Patients were asked to wear two wrist-worn sensors, AX3 for 3 full days (including night time); the accelerometer data are very large and contain too much "noise" and "still" (see **Section** 4.1.2, **Chapter** 4). After data are transformed from the **DWT**, the corresponding **DWT** coefficients will contain no or little useful information at different decomposition scales. Thus, how to find a proper way to carry out these **DWT** coefficients becomes our first challenge. Under a Bayesian strategy, based on spike-and-slab mixture prior (Chipman et al. (1997), CLYDE et al. (1998) and Clyde & George (2000)), the small values of wavelet coefficients can be shrunk to zero through the wavelet-domain. Ma & Soriano (2017) used a Bayesian hierarchical model, called Normal Inverse-Gamma Markov Tree Model (**NIG-MT**), for the first step of data reduction. Wavelet coefficients are connected to the posterior marginal probability in a hidden state which indicates the zero or non-zero state of each coefficient. Under the posterior marginal probability, a special threshold, which is evaluated from the energy loss rate (see **Section** 4.4.2, **Chapter** 4) in accelerometer data, can be defined. Based on that threshold, the wavelet coefficients

with small probability will be forced to zero. Hence the "noise" and "still" parts among wavelet coefficients can be eliminated. Moreover, the data reduction process gives good performance with negligible loss of the total information.

Although about three quarters of **DWT** coefficients can be forced to zero and removed through the **NIG-MT** model, the number of **DWT** coefficients is still very large. As we discussed previously, the **DWT** coefficients consist of both the time-domain and frequency-domain data. Now we have a clear picture: the horizontal and vertical direction of the **DWT** coefficients present the time-domain and frequency-domain data respectively in the **DWT** tree structure. For the vertical direction, **DWT** coefficients at different decomposition levels can be treated as the multi-dimensions corresponding to the different frequencies. It also shows a strong correlation between parent node and children nodes at different decomposition levels within the special tree structure. Hence, it is hard to give an analysis to the large scale of **DWT** coefficients directly.

To further analyse the 'details' of the **DWT** coefficients from the accelerometer data, we used the slide window approach, which divides the whole **DWT** tree structure into different small **DWT** tree structures in the time-domain. In other words, the **DWT** coefficients in the whole tree structure can be analysed by one small **DWT** tree structure to another small **DWT** tree structure in the time-domain. However, each small **DWT** tree structure is still difficult to analyse since it contains a large sample size of **DWT** coefficients in both the time and frequency-domains. To address this problem, We further use the multivariate functional principal component analysis and then find patterns in each slide window.

Ramsay & Silverman (2005) provide the multivariate functional principal component analysis (**fPCA**) which is used to further reduce the dimensionality of the small **DWT** tree structure in both the time and frequency-domains. Through multivariate **fPCA**, the small **DWT** tree structure at several decomposition levels (which contain massive **DWT** coefficients) can be transformed into a lower dimensional space with minimal loss of information. Moreover, Principal Components (PCs) are obtained in order to retain most of the variation presented at transformed levels from the small **DWT** tree structure.

There are two advantages to our data reduction system (**NIG-MT**, slide window and multivariate **fPCA**) based on the wavelet-domain: **(i)** the large scale of **DWT** coefficients can be reduced efficiently and then based on the small **DWT** tree structure. **(ii)** the useful information can be obtained from the pattern of the small **DWT** tree structure among the accelerometer data.

1.4 Classification based on the DWT tree structure

A subject's activities are described and recorded by the accelerometer data. We aim to investigate the **DWT** tree structure from the accelerometer data to further classify the subject's activities. The whole **DWT** tree structure can be transformed from the entire data and, to some extent, the small **DWT** tree structure will be more valued. To be more specific, a great quantity of activities (e.g., walking and running) correspond to patterns (see **Section** 5.2.1, **Chapter** 5) in our daily life. Instead of carrying out an analysis on the whole **DWT** tree structure, we prefer to apply the classification based on the small **DWT** tree structure. Through the slide window and multivariate **fPCA** approaches, the large sample size of raw **DWT** coefficients will be reduced efficiently. Moreover, useful information can be extracted from the pattern of small **DWT** tree structure from the designed data. Then the classification of three activities will be conducted based on this infromation.

1.5 Evaluating the upper limb function for stroke patient

The CAHAI-9 assessment is a fully validated measure (Barreca et al., 2006) of assessing upper limb functional ability. It has 9 items, and each is scored by using a 7-point quantitative scale. However, this method is subjective and also it is quite time consuming, leading to high costs. We aim to find the relationship between clinical assessed CAHAI scores and accelerometer data; then the predictive models will be built to measure the recovery level of the stroke patients by using free-living data.

After **DWT**, the wavelet coefficients present the particular frequency information at certain decomposition scales. Preece et al. (2009) provide some commonly used wavelet features based on wavelet coefficients through the wavelet energy preserving condition. However, patients after stroke have difficulties in moving one hand because of the brain injury caused by the stroke. The data is collected from stroke patients for each hand, but usually the data from paralysed side describes the most upper limb functional ability. In contrast, the data from non-paralysed side will provide less information because the patients can move their non-paralysed side normally. In addition, there is severe heterogeneity among patients. The direct use of those commonly used features failed to provide good predictive results. Given this problem, our challenge is to combine the information from both hands for each patient. Two new types of wavelet features (see **Chapter** 3), which combine the information for both hands, are proposed and they show very good performance when they are used to predict upper limb function for stroke patients. Furthermore, Gaussian Mixture Models (**GMM**) (Bishop, 2007) can be used to cluster the useful information from different small **DWT** tree structures through the **NIG-MT**, slide window and multivariate **fPCA**. New types of clustering features will also be defined based

on the information in different clusters (see **Chapter** 6).

Based on the wavelet and clustering features, the longitudinal mixed-effects model will be investigated to address heterogeneity between patients with a Bayesian Gaussian prior to define nonlinear random-effects. (Shi et al., 2012).

1.6 Contribution of the thesis

Two main contributions in this thesis are summarised as follows:

Stroke-rehab-driven Features We proposed two new types of compact wavelet-based features based on wavelet-domain from the accelerometer data: (i) Scalar features ($\boldsymbol{PNP}^1$ and $\boldsymbol{PNP}^2$; see **Section** 3.2.1); (ii)The clustering features $\boldsymbol{NPP}$ (from small **DWT** tree structures by using **NIG-MT**, slide window, multivariate **fPCA** and **GMM**; see **Section** 6.4.2). The above wavelet-based features can encode information from both paralysed and non-paralysed sides to represent upper limb functional abilities for stroke rehabilitation assessment. They are well defined the influences of personal behaviours or irrelevant daily activities for data collected in the noisy free-living environment.

Data Reduction System A data reduction system has been developed which combines the **NIG-MT**, slide window and multivariate **fPCA** approach in **Chapter** 6. Through the data reduction system based on the wavelet-domain, the sample size of original data can be reduced significantly and efficiently. Moreover, this also refines the valuable information during the procedure and gives a further analysis in the complex free-living environment. The data reduction system proposed in this thesis has shown good potential for solving similar problems involving bigger data.

1.7 Structure of the thesis

This thesis is organized as follows. Chapter 2 is the literature review and the background of the accelerometer data. Several wavelet transforms will be introduced: continue wavelet transform (**CWT**), discrete wavelet transform (**DWT**) and discrete wavelet packet transform (**DWPT**). We present the background of the accelerometer data including the clinical assessed score (CAHAI), participants (stroke patients), data collection and wrist-worn sensor AX3.

The next content evaluates the upper limb function by using the **DWT** and the **DWPT** in Chapter 3. Two important wavelet features will be defined and the predictive model will be included.

In Chapter 4, the data reduction through the Normal Inverse-Gamma Markov Tree (**NIG-MT**) model will be carried out. After data reduction, the large scale of no or little useful **DWT** coefficients will be shrunk with negligible loss from the total information.

Chapter 5 discusses the problem of classification based on slide window and multivariate **fPCA** from the design data. We investigate the problem with specific **DWT** tree structures by using slide window approach. Dealing with the covariance structure through the extension of multivariate **fPCA**, we then classify the different activities based on the designed accelerometer data.

Chapter 6 continues to focus on the specific **DWT** tree structure. Instead of the designed data, a branch of new clustering features is defined through **GMM** model from the stroke patients' free-living data. Afterwards, the predictive model will be used based on the new features.

In Chapter 7, we highlight the main contribution, give a conclusion and close the thesis by suggesting ideas for further research.

Chapter 2

Literature review and the background of the data

The overview of related and existing work in the area of accelerometer data is given in this chapter. Also, the background of the accelerometer data will be introduced. Since the following chapters will be mainly based on the wavelet transforms, we will introduce how wavelet transforms work properly. Based on the definition of a wavelet, we introduce the continuous wavelet transform (**CWT**), and extend this concept to the discrete wavelet transform (**DWT**) and the discrete wavelet packet transform (**DWPT**). We begin with the accelerometer data from wrist-worn sensor AX3 and the related and existing work will be introduced in **Section** 2.1. **Section** 2.2 provides the background of the accelerometer data, which includes the clinical assessed score (CAHAI), participants (stroke patients), data collection and wrist-worn sensor AX3. Finally, **Section** 2.3.1 introduces the basic idea of the wavelet and its properties. We will discuss the **CWT**'s in **Section** 2.3.2. Based on **CWT**, we move on to **DWT**, which is the specific tool used in this thesis, in **Section** 2.3.3. Instead of defining the **DWT** by using the filter notation, we mainly use matrix notation which is a much easier and more direct way. After that, we introduce **DWPT** which is a more complicated method based on **DWT** in **Section** 2.3.4.

2.1 Overview on evaluating upper limb function after stroke

It is widely known that stroke is a worldwide health problem causing disability and death (Donnan et al., 2008). It occurs when a blood clot cuts off oxygen supplied to a region of the brain. Hemiparesis is a very common symptom of post-stroke and is the fractional or intact paralysis of one side of the body, i.e., the opposite side to where the blood clot occurred, and it results in difficulties in performing activities, e.g., reduced arm movement. Patients can recover some of their capabilities with intense therapeutic input,

so it is important to assess their recovery levels over time.

Brain imaging techniques, which are considered the most reliable approach, can provide information regarding brain haemodynamics (Wintermark et al., 2005). However, this approach requires special equipment and is very expensive. Questionnaire-based approaches investigate the functional ability during a period, and can be categorised into two types: patient-completed and caregiver-completed (Ferrari et al., 2007). Although it is much cheaper than brain imaging, it may contain a high-level of bias. For instance, patients may not remember their daily activities (i.e.,recall bias) and the caregivers may not be able to observe the patients all the time. These biases make questionnaire-based approaches less precise. Lab-based clinical assessment approaches (Barreca et al., 2005b)(Barreca et al., 2005a), on the other hand, provide an alternative solution. The patients' upper limb functionality will be assessed by clinicians, e.g., by observing patients' capabilities of finishing certain pre-defined activities (Barreca et al., 2005b). Compared with brain imaging or questionnaire-based approaches, the cost of lab-based clinical assessment approaches is reasonable and has high accuracy. However, this assessment is normally taken in clinics/hospitals, which is not convenient for the patients, making continuous monitoring less feasible.

2.1.1 Automated Behaviour Assessment using Wearables

Recently, wearable sensing and machine learning (ML) techniques were comprehensively studied for automated health assessment. Compared with the traditional assessment approaches (e.g., via self-reporting, clinical assessment, etc.) which are normally subjective and expensive, the automated systems may provide an objective, low-cost alternative, which can also be used for continuous monitoring/assessment. Some automated systems were developed to assess the behaviours of diseases such as Parkinson's disease (Rehman et al., 2019) (Hammerla et al., 2015), autism (Plötz et al., 2012), depression (Little et al., 2020); or to monitor the health status such as sleep (Zhai et al., 2020) (Supratak et al., 2017), fatigue (Bai et al., 2020), (Ibrahim et al., 2020) or recovery-level from surgery (Ratcliffe et al., 2020) (Gurchiek et al., 2019), etc.

After collecting behavioural or physiological signals (e.g., accelerometers, ECG, audio, etc.), assessment/monitoring models can be developed. For applications with a high interpretability requirement, feature engineering can be a crucial step. For example, with gait parameters extracted from IMU sensors (such as stride, velocity, etc.), one can build simple ML models (e.g., random forest) for Parkinson's disease classification (Rehman et al., 2019) or fatigue score regression (Ibrahim et al., 2020). Compared with the redundant IMU data, gait parameters are more compact and interpretable, making it suitable for clinical applications. However, designing interpretable/clinically-relevant features can be a time-consuming process, which may also require domain knowledge (Zhai et al.,

2020)(Ibrahim et al., 2020) (Rehman et al., 2019)(Ratcliffe et al., 2020) (Gurchiek et al., 2019).

On the other hand, when interpretability is less important, deep learning can be an alternative approach, and can be directly applied to the raw signal (Supratak et al., 2017) or engineered features (Hammerla et al., 2015) (Zhai et al., 2020) (Bai et al., 2020) (Little et al., 2020) for (high-level) representation learning and classification/regression tasks. However, it normally requires adequate data annotation for better model generalisation.

2.1.2 Sensing Techniques for Automated Stroke Rehabilitation Monitoring

With the rapid development of sensing/ML techniques, researchers also started to use various sensors for stroke rehabilitation monitoring. In Dolatabadi et al. (2017), a Kinect sensor was used in home-like environments to monitor the key joints such that stroke patients' behaviour can be assessed. In Ganesh et al. (2018), a wireless surface Electromyography (sEMG) device was used to monitor the muscle recruitment of the post-stroke patients to see the effect of orthotic intervention. In clinical environments, five wearable sensors were placed on the trunk, upper and forearm of the two upper limbs to measure the reaching behaviours of stroke survivors (Jung et al., 2018). To monitor motor functions of stroke patients during rehabilitation sessions at clinics, an ecosystem including a jack and a cube for hand grasping monitoring, as well as a smart watch for arm dynamic monitoring was designed (Bobin et al., 2019). In this way, it is possible to objectively assess/measure the behaviours of the stroke patients, yet they are either limited to clinical environments (Bobin et al., 2019)(Jung et al., 2018) (Ganesh et al., 2018) or constrained environments (e.g., in front of a camera (Dolatabadi et al., 2017)).

2.2 Background of the accelerometer data

In this section, we begin by providing information about the clinical assessment known as CAHAI. Then some details of the stroke patients and a description of the data collection method will be given. Finally, we introduce the procedure used for data pre-processing in this thesis.

2.2.1 Chedoke Arm and Hand Activity Inventory (CAHAI)

As described in **Section** 2.1, lab-based clinical assessment was one of the most effective stroke rehabilitation assessment methods. We introduce the lab-based approach named Chedoke Arm and Hand Activity Inventory (CAHAI) scoring, which corresponds to the accelerometer data from AX3 device.

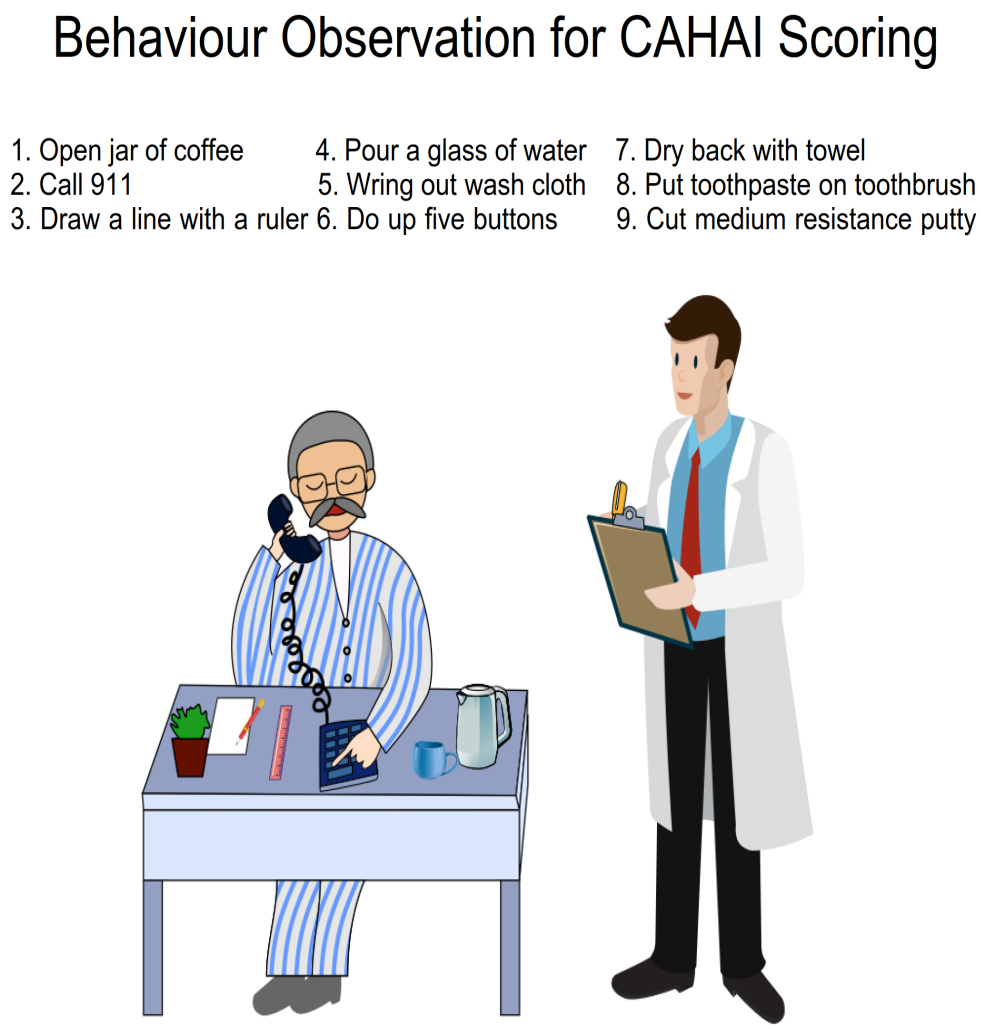


Figure 2.1: The clinical behaviour assessment for CAHAI scoring.

CAHAI scoring is a clinical assessment method for stroke rehabilitation, and it is a fully validated measure (Barreca et al., 2006) of upper limb functional ability with 9 tasks which are scored by using a 7-point quantitative scale. In the assessment, the patient will be asked to perform 9 tasks, including opening a jar of coffee, drawing a line with a ruler, calling 911, etc. Then the clinician will score these behaviours based on patient's performance at a scale from 1 (total assist weak) to 7 (complete independence i.e., timely, safely) (Barreca et al., 2006). A task example "call 911" is shown in **Figure** 2.1. Thus the minimum and maximum summation scores are 7 and 63 respectively. A CAHAI score form can be found in **Figure** 2.2 as follows:

Chedoke Arm and Hand Activity Inventory: Score Form

CAHAI-9 Version

Name: **Date:**

Activity Scale			
1. total assist (weak U/L < 25%) 2. maximal assist (weak U/L = 25-49%) 3. moderate assist (weak U/L = 50-74%) 4. minimal assist (weak U/L > 75%)	5. supervision 6. modified independence (device) 7. complete independence (timely, safely)		
	Affected Limb:		Score
1. Open jar of coffee	☐ holds jar	☐ holds lid	☐
2. Call 911	☐ holds receiver	☐ dials phone	☐
3. Draw a line with a ruler	☐ holds ruler	☐ holds pen	☐
4. Pour a glass of water	☐ holds glass	☐ holds pitcher	☐
5. Wring out washcloth			☐
6. Do up five buttons			☐
7. Dry back with towel	☐ reachs for towel	☐ grasps towel end	☐
8. Put toothpaste on toothbrush	☐ holds toothpaste	☐ holds brush	☐
9. Cut medium resistance putty	☐ holds knife	☐ holds fork	☐
Total Score			☐ /63
Comments			

Figure 2.2: The CAHAI score form Barreca et al. (2006).

2.2.2 Participants and data collection

Data was collected as part of a larger research study, the aim of which was to use a bespoke, professionally-written video game as a therapeutic tool for stroke rehabilitation (Shi et al., 2013). Ethical approval was obtained from the National Research Ethics Committee and all work undertaken was in accordance with the Declaration of Helsinki. Written, informed consent from all the subjects was obtained. A cohort of 59 stroke survivors, without significant cognitive or visual impairment, were recruited for the study. Patients were divided into two groups, i.e.,

- **Group 1**: the acute patient group, consisting of 26 participants who enrolled into the study within 6 months after stroke;
- **Group 2**: the chronic patient group, was formed by 33 participants who were 6 months or more post onset of stroke.

The distributions of acute/chronic condition, gender, dominant/non-dominant hand, paralysed/non-paralysed side with respect to age are shown in **Figure** 2.3.

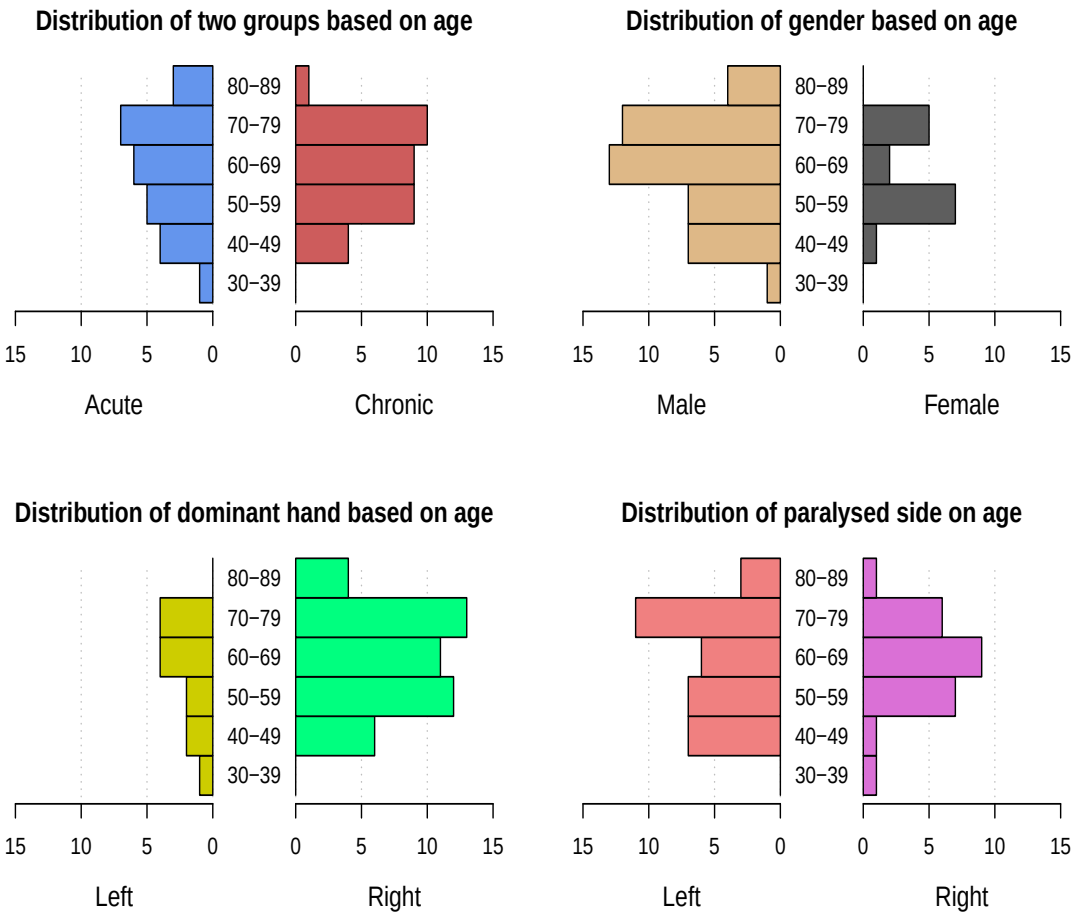


Figure 2.3: The distributions of acute/chronic condition, gender, dominant/non-dominant hand, paralysed/non-paralysed side with respect to age among the 59 subjects

These 59 patients visited the clinic for the CAHAI scoring every week (a random weekday) for a duration of 8 weeks. Over the course of the eight weeks, they were asked to wear two wrist-worn sensors for 3 full days (including night time) a week. They were also advised to remove the device during shower or swimming. Since some patients needed time to get used to this data collection procedure, we did not use the first week's accelerometer data. This was done in order to obtain better data. The first week's CAHAI scores were used as medical history information.

In contrast to other afore-mentioned sensing techniques (Jung et al., 2018)(Bobin et al., 2019)(Ganesh et al., 2018)(Dolatabadi et al., 2017), in this study we collected the accelerometer data from wrist-worn sensors in free-living environments. The sensor used

for this study, i.e., AX3 (Axivity Ltd, 2013), is a triaxial accelerometer logger that was designed for physical activity/behaviour monitoring, and it has been widely used in the medical community (e.g., for the UK Biobank physical activity study (Doherty et al., 2017)). The wrist bands were also designed such that the users can comfortably wear them without affecting their behaviours. The accelerometers used to collect data are wrist-worn sensor AX3 (Axivity Ltd, 2013); the devices are small (39 mm $\times$ 36 mm $\times$ 12.5 mm), lightweight (19 g) and waterproof. Their dynamic range is $\pm$2 g to $\pm$ 16 g and sampling rates up to 100 Hz are possible. The data was collected at 100Hz sampling rate, which can well preserve the daily activities of a human being (Bouten et al., 1997). The devices are tri-axial and hence sensitive to movement along the three coordinate axes. Different from human activity recognition, which requires sample-wise or frame-wise annotation (Guan & Plötz, 2017) (Plotz & Guan, 2018), the data collection in this study is relatively straight-forward. The patients put on both wrist-worn sensors 3 full days a week, before visiting clinicians for CAHAI scoring (i.e., week-wise annotation).

In other words, we aim to use accelerometer data captured in free-living environments to represent the stroke survivors' upper limb activities to measure the degree of paresis (Hen, 1999) (i.e., CAHAI score).

One problem with most commercial sensors is that only summary data (e.g., step count from, say, a fitbit), instead of raw data, are available. The algorithms of producing summary data are normally non-open source, and may vary from vendor to vendor – making the data collection and analysis device-dependent, and thus less practical in terms of generalisation and scalability.

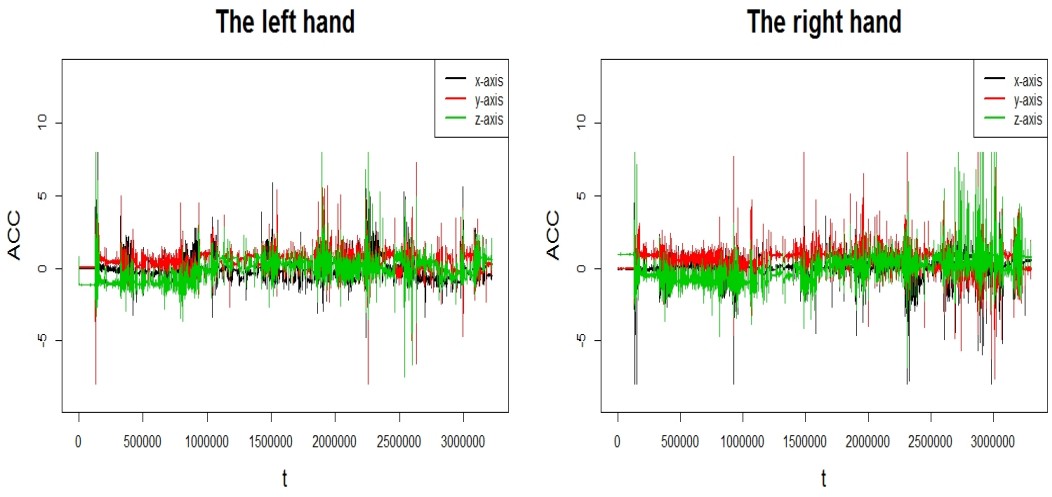


Figure 2.4: The accelerometer data (ACC) collected from the AX3 devices, including the acceleration along the x-, y- and z-axis over time. The data is collected at 100Hz from both wrists of a subject from 8am to 5pm.

The AX3 device used in this study, on the other hand, outputs the raw acceleration information in the x, y, z directions. In **Figure** 2.4, we present the raw acceleration data (measured in g units 1g = $9.8m/s^2$) collected from both wrists. It is simple and transparent, making the collected data re-usable, which is crucial for research communities.

2.2.3 Data pre-processing

For accelerometer data, signal vector magnitude (**VM**) (Karantonis et al., 2006) is a popular representation, which is simply the magnitude of the triaxial acceleration data defined as followed:

$$a(t) = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)},$$

where $a_x(t)$,$a_y(t)$,$a_z(t)$ are the acceleration along the x, y, z axes at timestamp t. The gravity effect can be removed by:

$$VM(t) = |a(t) - 1|.$$

Because of its simplicity and effectiveness, **VM** has been widely used in health monitoring tasks, such as fall detection (Karantonis et al., 2006), physical activity monitoring (Doherty et al., 2017), perinatal stroke assessment (Gao et al., 2019), etc.

Moreover, after we combine the 3 coordinate axes (x,y,z) and transform it to **VM**, the accelerometer data's sampling frequency is 100HZ (100 points in each second). Since the patients wear the wrist-worn sensor AX3 for 3 full days, which means the size of the data is large and we need to reduce the sample size at first, the data for each second consists of 100 observations. To further reduce the data volume, we take the mean of these 100 observations at each second and transform the data as 1 second-wise **VM** data. The 1 second-wise **VM** will be used with notation "**VM** data" in the **Chapters** 3, 4 and 6. Some second-wise **VM** examples can be found in **Figure** 3.5, **Chapter** 3.

2.3 Wavelet theory

2.3.1 The essence of wavelet

A wavelet is a mathematical tool which deals with time series and images powerfully, especially in signal processing applications. So what is wavelet? A wavelet is a very "small wave", which grows and decays dramatically in a very short time period. Comparing with the "big wave", for example, the sine function, it has a swinging up and down in the plot of $\sin(\mu)$ or $\cos(\mu)$ where $\mu \in (-\infty, \infty)$. **Figure** 2.5 gives examples of four types of wavelet.

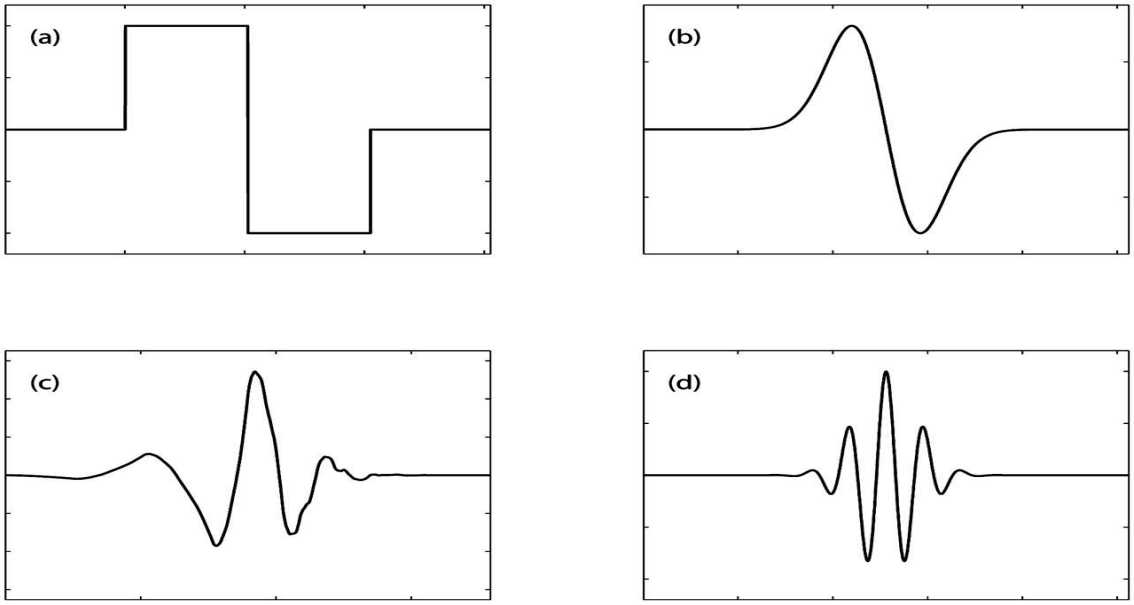


Figure 2.5: Different types of "small wave" (see Baker (2007)). (a) Haar wavelet, (b) Gaussian wavelet of order 1, (c) Daubechies wavelet of order 4, and (d) Morlet wavelet.

A wavelet function $\psi(\cdot)$ satisfies two basic properties:

(i): The integral of $\psi(\cdot)$ is zero:

$$\int_{-\infty}^{+\infty} \psi(\mu) d\mu = 0. \tag{2.1}$$

(ii): The integral of squared $\psi(\cdot)$ to unity:

$$\int_{-\infty}^{+\infty} \psi^2(\mu) d\mu = 1. \tag{2.2}$$

Any function can be defined as wavelet if it satisfies **Equations** (2.1) and (2.2).

There is another important and common additional condition, which is admissibility condition. A wavelet $\psi(\cdot)$ is admissible if $\psi(t) \in \mathbf{L^2}$ and satisfies:

$$C_\psi = \int_{-\infty}^{+\infty} \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|} d\omega < \infty, \tag{2.3}$$

where $\psi(\mu)$ is the wavelet function, the $\widehat{\psi}(\omega)$ is the Fourier transform of $\psi(\mu)$. For example, the first, (a) in **Figure** 2.5, is called the Haar wavelet function:

$$\psi^{(\text{Haar})}(u) \equiv \begin{cases} -1/\sqrt{2}, & -1 < u \leq 0 \\ 1/\sqrt{2}, & 0 < u \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

The second, (b) in **Figure** 2.5, is a Gaussian wavelet function, with the Gaussian (normal) probability density function for a random variable with zero mean and variance σ^2 as follows:

$$\phi^{(\text{Gaussian})}(u) \equiv \frac{e^{-u^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}}, \quad -\infty < u < \infty.$$

the details can be found in Donald B. Percival (2000).

Why are wavelets so useful? It is because they measure the weighted average of a time series (signal) varying from one average period to the next. This will be discussed in the next subsection. In wavelet analysis, there are typically two types of wavelet transform, **CWT** and **DWT**. In this project, we mainly focus on **DWT**.

2.3.2 Continuous wavelet transform (CWT)

From **Figure** 2.5, we see that the wavelet function is a wave-like oscillation with an amplitude that begins from zero, increases and then decreases back to zero. In **CWT**, we mainly focus on the wavelet function.

The main purpose of the wavelet function is to provide a source function to generate the daughter wavelet which is simply a translated and scaled versions of the wavelet function:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}}\, \psi(\frac{t-b}{a}),$$

where $\psi(t)$ is a continuous function called the wavelet function in both the time domain and the frequency domain. The wavelet function is scaled by a factor of a and translated by a factor of b. We do the scaling and shifting in the wavelet function by changing the parameters a and b. If we change the parameter a (scaling), we will get the functions shown in **Figure** 2.6:

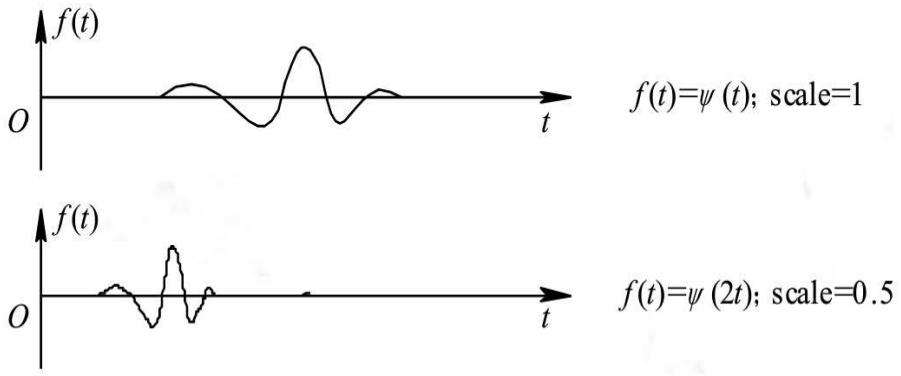


Figure 2.6: Scaling the wavelet function

Similarly, we can also shift the wavelet function through the parameter b (translating); see examples in **Figure** 2.7.

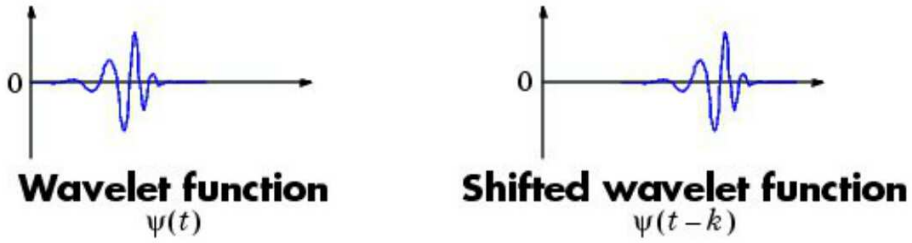


Figure 2.7: Shifting the wavelet function

CWT is used to divide a continuous-time function into wavelets. The continuous wavelet transform of a function (or signal) $x(t)$ at a scale (a>0, $a \in \mathbf{R}^+$) and translational value b ($b \in \mathbf{R}$) is expressed by the following integral:

$$\mathrm{W}_c(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} x(t)\overline{\psi(\frac{t-b}{a})}dt. \tag{2.4}$$

where $\psi(t)$ is the wavelet function and $\overline{\psi(t)}$ is the conjugate calculator of $\psi(t)$, based on the **Figures** 2.6 and 2.7, the wavelet function is like a 'rule' which can map (through the scaling and shifting) to different part of the signal. **CWT** is the sum over all the time of the signal multiplied by scaled and shifted versions of the wavelet which can be found in **Figure** 2.8. This process produces wavelet coefficients, which are a function of scale and position. We calculate the coefficients using different scales in different sections of the signal. The coefficients constitute the results of a regression of the original signal performed on the wavelets.

For example: consider a piece of signal by applying **CWT** from a scale of 1 to 64, ψ is the wavelet function (choosing Haar wavelet function). The coefficient is obtained using scale a and position b from **Equation** (2.4). Since the signal $x(t)$ is a discrete signal, we use a function $x(k)$ to represent it, where $k = 1, 2, ..., length(x)$. For each scale parameter a, the value of b is from $b_0 = 1$ to $b = length(x)$, calculating the corresponding coefficients with a from 1 to 64. The output of the **CWT** is a image shown in **Figure** 2.8.

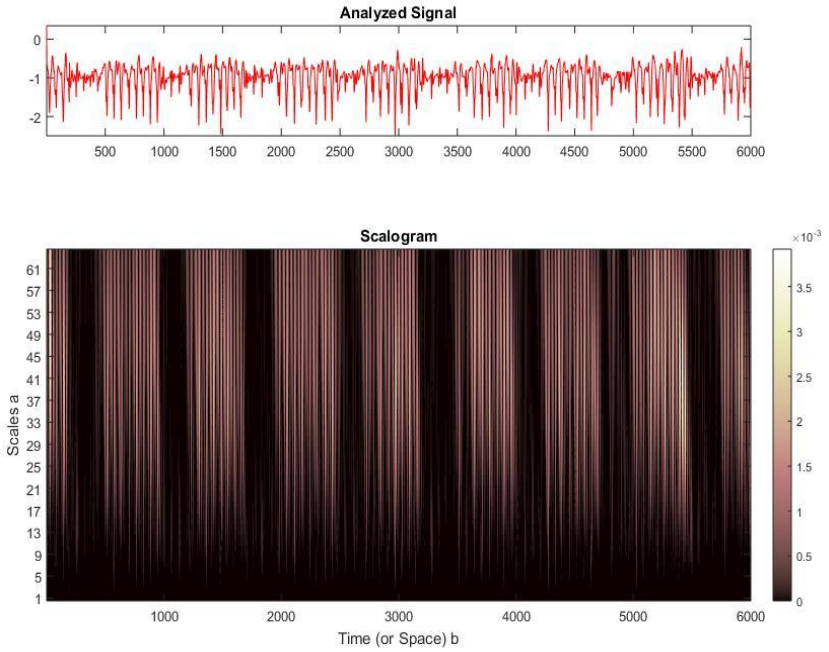


Figure 2.8: The scalogram of **CWT** for each wavelet coefficient by using Haar wavelet.

It is clear that from the above **CWT** plot, that the large magnitudes correspond to the bright spots, which means that the signal changes or "jumps" dramatically in these time periods. In contrast, the dark area represents the small magnitudes at which the signal does not change fast but keeps smooth status.

2.3.3 Discrete wavelet transform (DWT)

Following on from **CWT**, the discrete wavelet transform (**DWT**) is a more efficient way of analysing the signal. From **CWT**, we can find that we transform the 1 dimensional signal into the 2 dimensional **CWT** image plot. **CWT** processes the magnitudes of coefficients in each scale, and it is easy to find that there is usually a small variation in the adjacent scale. For example, the images in scales 61 and 62 are very similar and there are only slight differences between them (see **Figure** 2.8). It is obvious that the result of the **CWT** process contains a lot of redundancy. The idea of **DWT** is to deal with the "dyadic" scale, in other words, to give a discrete scale of 2^j based on the continuous scale of **CWT**. It is easy to find that the **DWT** coefficients reflect long term variations, which are "proportional to the differences of weighted averages" (see **Figure** 2.9).

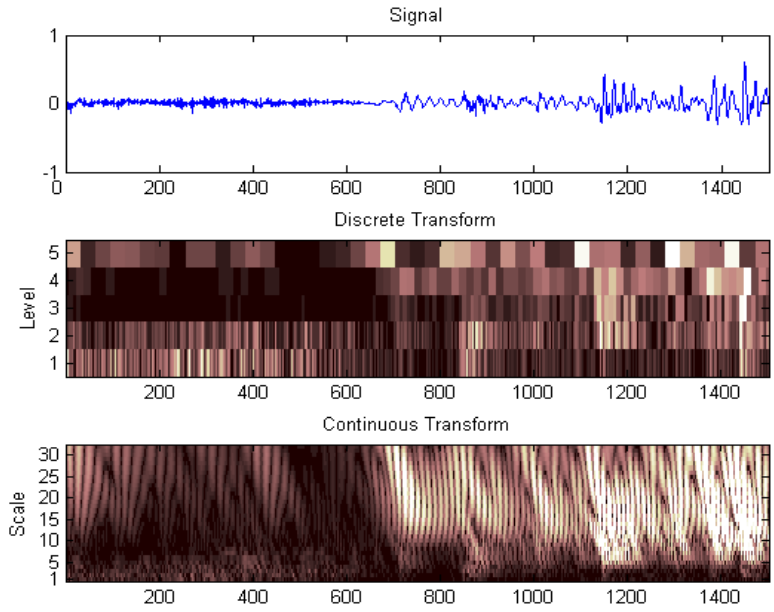


Figure 2.9: Comparing with the scalogram of DWT and CWT.

To be more specific, the process of **CWT** will compute the wavelet coefficients at different scales in different sections of the signal, so it will be computationally demanding. In such cases, discrete analysis is sufficient and continuous analysis is redundant. So we need the discrete wavelet transform, based on the wavelet function of the continuous wavelet transform:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}}\,\psi(\frac{t-b}{a}).$$

If we choose a part of the scaled factor of a and the translated factor of b, it will minimize the process of computing. Assuming $a = a_0^j$, $b = kb_0a_0^j$, $j, k \in \mathbf{Z}$, the wavelet function becomes:

$$\psi_{j,k}(t) = |a_0|^{-\frac{j}{2}}\,\psi(a_0^{-j}t - kb_0), \qquad j, k \in \mathbf{Z}.$$

We then have the discrete wavelet transform, which is expressed by the following integral of a function, $x(t)$, at a scale value a and translational value b,

$$\mathrm{W}_d(j,k) = |a_0|^{-\frac{j}{2}} \int_{-\infty}^{+\infty} x(t)\overline{\psi(a_0^{-j}t - kb_0)}dt. \tag{2.5}$$

Specifically, if we assume $a_0 = 2$, $b_0 = 1$, we get the dyadic wavelet:

$$\psi_{j,k}(t) = 2^{-\frac{j}{2}}\psi(2^{-j}t - k), \qquad j, k \in \mathbf{Z}.$$

Here, we usually see the discrete wavelet transforms the stretch and shift parameters by powers of 2; stretching or shifting by powers of 2 is often referred to as "dyadic" (see Donald B. Percival (2000)). For example, dyadic dilation means stretching (or shrinking) by factors of 2 (e.g., $2, 4, 8, 16$).

The essence of the DWT procedure and the pyramid Algorithm

The **DWT** procedure includes two parts: decomposition and reconstruction, this project mainly focuses on the decomposition part. When $\mathbf{X}$ is the raw signal with length $N = 2^J$, we can describe details of the **DWT** using matrix algebra:

$$\mathbf{W} = \mathcal{W}\mathbf{X}, \tag{2.6}$$

where $\mathbf{W}$ is the output of matrix of **DWT** coefficients in different scales. It consists of all the $\mathrm{W}_d(j,k)$ from a different scale j and location k in **Equation** (2.5). The orthonormal matrix $\mathcal{W}$ is designed by the **DWT** pyramid algorithm, and more discussion of this algorithm will be given later in this section. $\mathcal{W}$ is the orthonormal matrix containing different orthonormal wavelet bases ($\mathcal{W}$ depends on which wavelet function is used, the formulas can be found in Daubechies (2006) and Donald B. Percival (2000)). It satisfies $\mathcal{W}^T\mathcal{W} = \mathbf{I}_N$. The orthonormal matrix $\mathcal{W}$ with dimension $N \times N$ can be separated into $J+1$ submatrices $\mathcal{W} = [\mathcal{W}_1, \mathcal{W}_2, ..., \mathcal{W}_J, \mathcal{V}_J]^T$, each of $\mathcal{W}_j$ can produce a partitioning of the vector $\mathbf{W}$ of **DWT** coefficients in each scale $j, j = 1, 2, ..., J$.

If we rewrite **Equation** (2.6) in the following form, the raw signal $\mathbf{X}$ can be expressed at the decomposition scale J based on Mallat's pyramid algorithm (Mallat, 1989) as follows:

$$\mathcal{W}\mathbf{X} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \vdots \\ \mathcal{W}_J \\ \mathcal{V}_J \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{W}_1\mathbf{X} \\ \mathcal{W}_2\mathbf{X} \\ \vdots \\ \mathcal{W}_J\mathbf{X} \\ \mathcal{V}_J\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_J \\ \mathbf{V}_J \end{bmatrix} = \mathbf{W}, \tag{2.7}$$

where $\mathbf{W}_j$ is a column vector of length $N/2^j$ representing the differences in adjacent weighted averages from scale 1 to scale J, $\mathbf{V}_J$ is the last column contained in $\mathbf{W}$ which has the same length with $\mathbf{W}_J$. $\mathbf{W}_j$ is defined as detailed (wavelet) coefficients at scale j. $\mathbf{V}_J$ contains the approximated (scaling) coefficients at the J-th scale. Note that the approximated coefficients at j scale are used to generate the detailed coefficients at $j+1$ scale; we will discuss this in the pyramid algorithm later. $\mathcal{W}_j$ has dimension $N/2^j \times N$, where $j = 1, 2, ..., J$ and $\mathbf{V}_J$ has the same dimension with $\mathbf{W}_J$. Note that the rows of design orthonormal matrix $\mathcal{W}$ depend on the decomposition level j-th. In other words, the

value of J depends on the **DWT** decomposition scale of the raw signal. The maximum decomposition level j equals J since our signal $\mathbf{X}$ has length $N = 2^J$.

The pyramid algorithm in DWT

We now discuss the pyramid algorithm which describes how to calculate **DWT** recursively for $j = 1, 2, ..., J$. Previously, we introduced the overall picture of the **DWT** process; we continue to explore it in depth. As we mentioned before, the rows of design orthonormal matrix $\mathcal{W}$ depend on the number and scale of J when dealing with a raw signal. Given the decomposition level $J = 1$, the **Equation** (2.7) becomes:

$$\mathcal{W}\mathbf{X} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{V}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{W}_1\mathbf{X} \\ \mathcal{V}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{V}_1 \end{bmatrix}, \tag{2.8}$$

where $\mathcal{W}_1$ is a matrix of dimension $N/2 \times N$ which satisfies $\mathcal{W}_1{\mathcal{W}_1}^T = \mathbf{I}_{N/2}$, and contains the first $N/2$ rows in our design orthonormal matrix $\mathcal{W}$, the length of detailed coefficients in vector $\mathbf{W}_1 = \mathcal{W}_1\mathbf{X}$ is $N/2$. Similarly, the matrix $\mathcal{V}_1$ of dimension $N/2 \times N$ satisfies $\mathcal{V}_1{\mathcal{V}_1}^T = \mathbf{I}_{N/2}$ and contains the last $N/2$ rows in our design orthonormal matrix $\mathcal{W}$, the length of the approximated coefficients in vector $\mathbf{V}_1 = \mathcal{V}_1\mathbf{X}$ is $N/2$. Note that $\mathcal{W}_1{\mathcal{V}_1}^T = \mathcal{V}_1{\mathcal{W}_1}^T = \mathbf{0}_{N/2}$.

Now, we consider the decomposition level $J = 2$, the **Equation** (2.7) becomes:

$$\mathcal{W}\mathbf{X} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \mathcal{V}_2 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{W}_1\mathbf{X} \\ \mathcal{W}_2\mathbf{X} \\ \mathcal{V}_2\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{V}_2 \end{bmatrix}, \tag{2.9}$$

where $\mathcal{W}_1$ and $\mathbf{W}_1$ are the same as the first step of pyramid Algorithm **DWT**. $\mathcal{W}_2$ is a matrix of dimension $N/4 \times N$ which satisfies $\mathcal{W}_2{\mathcal{W}_2}^T = \mathbf{I}_{N/4}$, and contains the $N/4$ rows in our design orthonormal matrix $\mathcal{W}$, the length of detailed coefficients in vector $\mathbf{W}_2 = \mathcal{W}_2\mathbf{X}$ is $N/4$. Similarly, the matrix $\mathcal{V}_2$ of dimension $N/4 \times N$ satisfies $\mathcal{V}_2{\mathcal{V}_2}^T = \mathbf{I}_{N/4}$ and contains the last $N/4$ rows in our design orthonormal matrix $\mathcal{W}$, the length of the approximated coefficients in vector $\mathbf{V}_2 = \mathcal{V}_2\mathbf{X}$ is $N/4$. Note that, $\mathcal{W}_2{\mathcal{V}_2}^T = \mathcal{V}_2{\mathcal{W}_2}^T = \mathbf{0}_{N/4}$.

Let $\mathcal{A}_2$ and $\mathcal{B}_2$ to be the matrices with dimension $\frac{N}{4} \times \frac{N}{2}$, which satisfies $\mathcal{W}_2 = \mathcal{B}_2\mathcal{V}_1$ and $\mathcal{V}_2 = \mathcal{A}_2\mathcal{V}_1$. Note that, $\mathcal{A}_2{\mathcal{A}_2}^T = \mathcal{B}_2{\mathcal{B}_2}^T = \mathbf{I}_{N/4}$ and $\mathcal{B}_2{\mathcal{A}_2}^T = \mathcal{A}_2{\mathcal{B}_2}^T = \mathbf{0}_{N/4}$. Now **Equation** (2.9) can be re-written to:

$$\mathcal{W}\mathbf{X} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{B}_2\mathcal{V}_1 \\ \mathcal{A}_2\mathcal{V}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{W}_1\mathbf{X} \\ \mathcal{B}_2\mathcal{V}_1\mathbf{X} \\ \mathcal{A}_2\mathcal{V}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{V}_2 \end{bmatrix}. \tag{2.10}$$

From **Equation** (2.8) in the first step of the pyramid Algorithm in **DWT**, we have $\mathbf{V}_1 =$

$\mathcal{V}_1\mathbf{X}$, so the **Equation** (2.10) can be re-written to:

$$\mathcal{W}\mathbf{X} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{B}_2\mathcal{V}_1 \\ \mathcal{A}_2\mathcal{V}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{W}_1\mathbf{X} \\ \mathcal{B}_2\mathbf{V}_1 \\ \mathcal{A}_2\mathbf{V}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{V}_2 \end{bmatrix}. \tag{2.11}$$

Let $\mathcal{P}_2 = \begin{bmatrix} \mathcal{B}_2 \\ \mathcal{A}_2 \end{bmatrix}$. After simplifying **Equation** (2.11), we get the following in the second step of pyramid Algorithm in **DWT**:

$$\mathcal{P}_2\mathbf{V}_1 = \begin{bmatrix} \mathcal{B}_2 \\ \mathcal{A}_2 \end{bmatrix} \mathbf{V}_1 = \begin{bmatrix} \mathcal{B}_2\mathbf{V}_1 \\ \mathcal{A}_2\mathbf{V}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_2 \\ \mathbf{V}_2 \end{bmatrix}. \tag{2.12}$$

Finally, considering the decomposition level J, recalling the design matrix in Jth level of decomposition:

$$\mathcal{W} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \mathcal{W}_3 \\ \vdots \\ \mathcal{W}_j \\ \vdots \\ \mathcal{W}_J \\ \mathcal{V}_J \end{bmatrix},$$

where $\mathcal{W}_j$ is a matrix of dimension $N/2^j \times N$ which satisfies $\mathcal{W}_j{\mathcal{W}_j}^T = \mathbf{I}_{N/2^j}$, and contains the $N/2^j$ rows in our design orthonormal matrix $\mathcal{W}$. Similarly, the matrix $\mathcal{V}_J$ of dimension $N/2^J \times N$ satisfies $\mathcal{V}_J{\mathcal{V}_J}^T = \mathbf{I}_{N/2^J}$ and contains the $N/2^J$ rows in our design orthonormal matrix $\mathcal{W}$.

For each $j \geq 2$, let $\mathcal{W}_j = \mathcal{B}_j\mathcal{V}_{j-1}$ and $\mathcal{V}_j = \mathcal{A}_j\mathcal{V}_{j-1}$, $\mathcal{A}_j$ and $\mathcal{B}_j$ are the matrices with dimension $\frac{N}{2^j} \times \frac{N}{2^{j-1}}$. Note that $\mathcal{A}_j{\mathcal{A}_j}^T = \mathcal{B}_j{\mathcal{B}_j}^T = \mathbf{I}_{N/2^j}$ and $\mathcal{B}_j{\mathcal{A}_j}^T = \mathcal{A}_j{\mathcal{B}_j}^T = \mathbf{0}_{N/2^j}$. We now extend our design orthonormal matrix $\mathcal{W}$ to the j-th level:

$$\mathcal{W} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \mathcal{W}_3 \\ \vdots \\ \mathcal{W}_j \\ \vdots \\ \mathcal{W}_J \\ \mathcal{V}_J \end{bmatrix} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{B}_2\mathcal{V}_1 \\ \mathcal{B}_3\mathcal{A}_2\mathcal{V}_1 \\ \vdots \\ \mathcal{B}_j\mathcal{A}_{j-1}\ldots\mathcal{A}_2\mathcal{V}_1 \\ \vdots \\ \mathcal{B}_J\mathcal{A}_{J-1}\ldots\mathcal{A}_2\mathcal{V}_1 \\ \mathcal{A}_J\mathcal{A}_{J-1}\ldots\mathcal{A}_2\mathcal{V}_1 \end{bmatrix}. \tag{2.13}$$

Now rewriting $\mathcal{B}_1 = \mathcal{W}_1$ and $\mathcal{A}_1 = \mathcal{V}_1$. The **Equation** (2.13) becomes:

$$\mathcal{W} = \begin{bmatrix} \mathcal{W}_1 \\ \mathcal{W}_2 \\ \mathcal{W}_3 \\ \vdots \\ \mathcal{W}_j \\ \vdots \\ \mathcal{W}_J \\ \mathcal{V}_J \end{bmatrix} = \begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2\mathcal{A}_1 \\ \mathcal{B}_3\mathcal{A}_2\mathcal{A}_1 \\ \vdots \\ \mathcal{B}_j\mathcal{A}_{j-1}\dots\mathcal{A}_2\mathcal{A}_1 \\ \vdots \\ \mathcal{B}_J\mathcal{A}_{J-1}\dots\mathcal{A}_2\mathcal{A}_1 \\ \mathcal{A}_J\mathcal{A}_{J-1}\dots\mathcal{A}_2\mathcal{A}_1 \end{bmatrix}. \tag{2.14}$$

If we recall $\mathbf{W}_1 = \mathcal{W}_1\mathbf{X}$ and $\mathbf{V}_1 = \mathcal{V}_1\mathbf{X}$, then the j-th level is followed:

$$\mathbf{W}_j = \mathcal{B}_j\mathcal{A}_{j-1}\dots\mathcal{A}_1\mathbf{X} = \mathcal{W}_j\mathbf{X},$$

$$\mathbf{V}_j = \mathcal{A}_j\mathcal{A}_{j-1}\dots\mathcal{A}_1\mathbf{X} = \mathcal{V}_j\mathbf{X}.$$

Let $\mathcal{P}_j = \begin{bmatrix} \mathcal{B}_j \\ \mathcal{A}_j \end{bmatrix}$, we get the formation of the j-th step of pyramid Algorithm in **DWT**:

$$\mathcal{P}_j\mathbf{V}_{j-1} = \begin{bmatrix} \mathcal{B}_j \\ \mathcal{A}_j \end{bmatrix}\mathbf{V}_{j-1} = \begin{bmatrix} \mathcal{B}_j\mathbf{V}_{j-1} \\ \mathcal{A}_j\mathbf{V}_{j-1} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_j\mathcal{A}_{j-1}\dots\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_j\mathcal{A}_{j-1}\dots\mathcal{A}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_j \\ \mathbf{V}_j \end{bmatrix}. \tag{2.15}$$

In practice, the terms $\mathcal{A}_j$ and $\mathcal{B}_j$, contained in $\mathcal{W}$, are low-pass and high-pass filters which relate to the different frequency domains. The discrete wavelet transform produces two types of coefficients from **DWT**: detailed coefficients $\mathbf{W}_j$ in the j-th scale, $j = 1, 2, ..., J$ and approximated coefficients $\mathbf{V}_J$ in scale J. The two types of coefficients represent the differences in adjacent weighted averages from scale 1 to scale J.

2.3.4 The discrete wavelet packet transform (DWPT)

We now further consider **DWPT**. The discrete wavelet packet transform is an expansion of the discrete wavelet transform.

Recalling the pyramid Algorithm in **DWT** at the jth decomposition, $j = 1, 2, ..., J$, the length of signal is $N = 2^J$, where J is an integer. $\mathcal{A}_j$ and $\mathcal{B}_j$ are the matrices with dimension $\frac{N}{2^j} \times \frac{N}{2^{j-1}}$, which satisfies $\mathcal{A}_j{\mathcal{A}_j}^T = \mathcal{B}_j{\mathcal{B}_j}^T = \mathbf{I}_{N/2^j}$ and $\mathcal{B}_j{\mathcal{A}_j}^T = \mathcal{A}_j{\mathcal{B}_j}^T = \mathbf{0}_{N/2^j}$. Let $\mathbf{W}_1 = \mathbf{W}_{1.1}$, $\mathbf{V}_1 = \mathbf{W}_{1.0}$. The first stage of pyramid Algorithm in **DWT** is followed:

$$\begin{bmatrix} \mathcal{B}_1 \\ \mathcal{A}_1 \end{bmatrix}\mathbf{X} = \begin{bmatrix} \mathcal{B}_1\mathbf{X} \\ \mathcal{A}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{V}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{1.1} \\ \mathbf{W}_{1.0} \end{bmatrix} = \mathbf{W}^{1*}. \tag{2.16}$$

Let $\mathbf{W}_2 = \mathbf{W}_{2.1}$, $\mathbf{V}_2 = \mathbf{W}_{2.0}$, then, for the second stage of pyramid Algorithm in **DWT**, the transform can be rewritten as followed:

$$\begin{bmatrix} \mathcal{B}_1 \\ \mathcal{B}_2\mathcal{A}_1 \\ \mathcal{A}_2\mathcal{A}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{B}_1\mathbf{X} \\ \mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_2\mathcal{A}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \mathbf{V}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{1.1} \\ \mathbf{W}_{2.1} \\ \mathbf{W}_{2.0} \end{bmatrix}. \tag{2.17}$$

From the **Equations** (2.16) and (2.17), it is easy to find the previous transform for $\mathcal{A}_1$ by using $\mathcal{B}_2$ and $\mathcal{A}_2$. At the same time, we set the term $\mathcal{B}_1$ aside, which means that we don't use $\mathcal{B}_1$ in the second step of pyramid Algorithm in **DWT**. Now we do the transform again by leaving $\mathcal{A}_1$ alone and dealing with $\mathcal{B}_1$ by using $\mathcal{B}_2$ and $\mathcal{A}_2$ as followed:

$$\begin{bmatrix} \mathcal{A}_1 \\ \mathcal{B}_2\mathcal{B}_1 \\ \mathcal{A}_2\mathcal{B}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{A}_1\mathbf{X} \\ \mathcal{B}_2\mathcal{B}_1\mathbf{X} \\ \mathcal{A}_2\mathcal{B}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{W}_2^* \\ \mathbf{V}_2^* \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{1.0} \\ \mathbf{W}_{2.3} \\ \mathbf{W}_{2.2} \end{bmatrix}. \tag{2.18}$$

Now, by combining the **Equations** (2.17) and (2.18), the whole second level transform deals with the terms $\mathcal{A}_1$ and $\mathcal{B}_1$ together by using $\mathcal{B}_2$ and $\mathcal{A}_2$ as follows:

$$\begin{bmatrix} \mathcal{A}_2\mathcal{A}_1 \\ \mathcal{B}_2\mathcal{A}_1 \\ \mathcal{A}_2\mathcal{B}_1 \\ \mathcal{B}_2\mathcal{B}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{A}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_2\mathcal{B}_1\mathbf{X} \\ \mathcal{B}_2\mathcal{B}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_2 \\ \mathbf{W}_2 \\ \mathbf{V}_2^* \\ \mathbf{W}_2^* \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{2.0} \\ \mathbf{W}_{2.1} \\ \mathbf{W}_{2.2} \\ \mathbf{W}_{2.3} \end{bmatrix} = \mathbf{W}^{2*}, \tag{2.19}$$

where $\mathbf{W}^{2*}$ is the result of the second stage of **DWPT**. Each coefficient vector $\mathbf{W}_{2.i}$ with length $N/2^2$, $i = 0, 1, 2, 3$, has the same dimension as the coefficients at the second level of **DWT** decomposition. Similarly, the third level discrete wavelet packet transform as follows:

$$\begin{bmatrix} \mathcal{A}_3\mathcal{A}_2\mathcal{A}_1 \\ \mathcal{B}_3\mathcal{A}_2\mathcal{A}_1 \\ \mathcal{A}_3\mathcal{B}_2\mathcal{A}_1 \\ \mathcal{B}_3\mathcal{B}_2\mathcal{A}_1 \\ \mathcal{A}_3\mathcal{B}_2\mathcal{A}_1 \\ \mathcal{B}_3\mathcal{B}_2\mathcal{A}_1 \\ \mathcal{A}_3\mathcal{A}_2\mathcal{B}_1 \\ \mathcal{B}_3\mathcal{A}_2\mathcal{B}_1 \end{bmatrix} \mathbf{X} = \begin{bmatrix} \mathcal{A}_3\mathcal{A}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{B}_3\mathcal{A}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_3\mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{B}_3\mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_3\mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{B}_3\mathcal{B}_2\mathcal{A}_1\mathbf{X} \\ \mathcal{A}_3\mathcal{A}_2\mathcal{B}_1\mathbf{X} \\ \mathcal{B}_3\mathcal{A}_2\mathcal{B}_1\mathbf{X} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{3.0} \\ \mathbf{W}_{3.1} \\ \mathbf{W}_{3.2} \\ \mathbf{W}_{3.3} \\ \mathbf{W}_{3.4} \\ \mathbf{W}_{3.5} \\ \mathbf{W}_{3.6} \\ \mathbf{W}_{3.7} \end{bmatrix} = \mathbf{W}^{3*}, \tag{2.20}$$

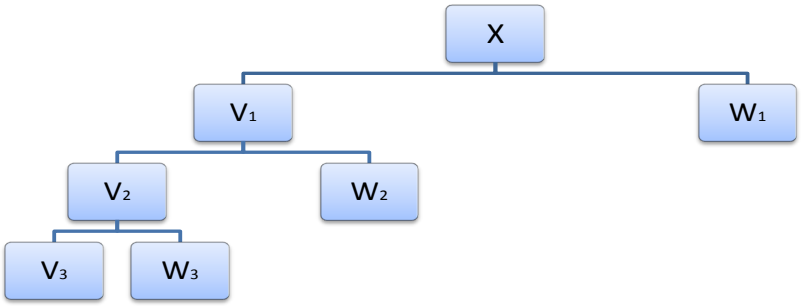
where $\mathbf{W}^{3*}$ is the result of the second stage of discrete wavelet packet transform. Each coefficient vector $\mathbf{W}_{3.i}$, with length $N/2^3$, $i = 0, 1, 2, ..., 7$, has the same dimension as the coefficients at the third level of **DWT** decomposition.

Generally, the discrete wavelet packet transform has the following formation:

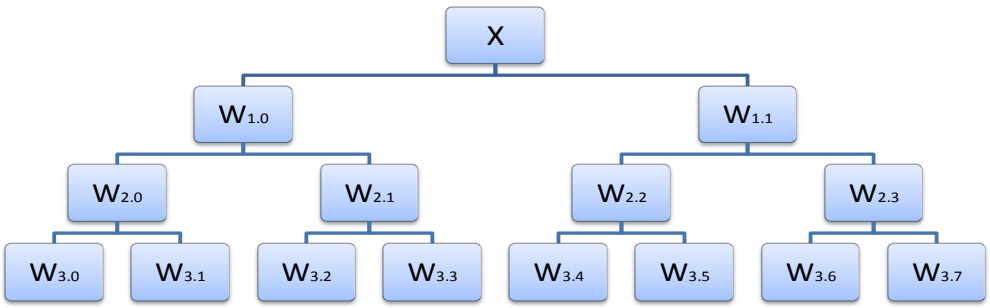
$$\begin{bmatrix} \mathbf{W}_{j.0} \\ \mathbf{W}_{j.1} \\ \vdots \\ \mathbf{W}_{j.2^j-2} \\ \mathbf{W}_{j.2^j-1} \end{bmatrix} = \mathbf{W}^{j*}, \tag{2.21}$$

where $\mathbf{W}^{j*}$ is the result of the jth stage of discrete wavelet packet transform. Each coefficient vector in $\mathbf{W}^{j*}$ with length $N/2^j$, has the same dimension with the coefficients in the third level of **DWT** decomposition. More details can be found in Donald B. Percival (2000).

Finally, at 3 scales of decomposition as example, a very striking plot will be given in **Figure** 2.10, where the difference between **DWT** and **DWPT** will be seen very clearly. The **DWT** divides the signal with the approximated coefficients and the detailed coefficients at scale 1. Then, we keep the detailed coefficients at scale 1 and the approximated coefficients are divided into the detailed coefficients and approximated coefficients at scale 2, and so on. However, through the **DWPT**, the signal is divided into the detailed coefficients and approximated coefficients at scale 1 firstly. Instead of dividing only the approximated coefficients into 2 parts at scale 2 in **DWT**, we also divide the detailed coefficients into two parts as well. In summary, the **DWT** just decomposes the approximated coefficients but keeps the detailed coefficients at each decomposition level; the **DWPT** decomposes both detailed coefficients and approximated coefficients at each decomposition level and it also gives more frequency information in the coefficients of **DWT** at scale 1.



Discrete Wavelet Transform (DWT) at 3 scales of decomposition.



Discrete Wavelet Packet Transform (DWT) at 3 scales of decomposition.

Figure 2.10: Comparing with the **DWT** and **DWPT** at three scales decomposition. The **DWT** just decomposes the approximated coefficients but keeps the detailed coefficients at each decomposition level, the **DWPT** decomposes both detailed coefficients and approximated coefficients at each decomposition level. Moreover, applying the **DWPT** will get more frequency information at the coefficients of DWT at scale 1 (see **Section** 3.2.1, **Chapter** 3).

Chapter 3

Evaluating the upper limb function based on DWT

In this work, we aim to build an automated stroke rehabilitation assessment system using wearable sensing and machine learning techniques. Our system is different from previously mentioned approaches that it can measure the patients objectively and continuously in free-living environments.

We collected accelerometer data using wrist-worn accelerometer sensors, and designed compact features that can capture rehabilitation-related movements, before mapping these features to clinical assessment scores (i.e., the model training process). The trained model can be used to infer recovery-level for other unknown patients. In free living environments, there are different types of movements which may be related to different frequencies. For example, activities such as running or jumping may correspond to high-frequency signal, while sedentary or eating may be low-frequency signal. In this study, instead of recognising the daily activities, which is hard to achieve given limited annotation (i.e., no frame/sample-wise annotation), we transformed the raw accelerometer data to the frequency domain, where we design features that can encode the rehabilitation-related movements. Specifically, wavelet transform (Donald B. Percival, 2000) was used, and the wavelet coefficients can represent the particular frequency information at certain decomposition scales. Preece et al. (2009) provided some commonly used wavelet features extracted from accelerometer data. However, to capture stroke rehabilitation-related activities, some domain knowledge should be taken into account to design better features. After stroke, patients have difficulties in moving one side (i.e., paralysed side) due to the brain injury, and data from paralysed side tends to describe more about the upper limb functional ability, than the non-paralysed side (i.e., normal side). However, such signals can be significantly affected by personal behaviours or irrelevant daily activities, and such "noises" should be dealt with before developing the predictive models. Various

wavelet features were studied, and we proposed two new types of feature that can encode information from both paralysed and non-paralysed sides, before developing predictive models for stroke rehabilitation assessment. We further propose to use the longitudinal mixed-effects model with Gaussian process prior (LMGP), which can model the random effects caused by different subjects and time slots (during the 8 weeks). Comprehensive experiments were conducted to evaluate our system on both acute and chronic patients, and the results suggested its effectiveness.

3.1 Background

The basic idea of wavelet transform is given in **Section** 2.3, **Chapter** 2. The background of clinical assessed CAHAI score, participants, data collection and data pre-processing were introduced in **Section** 2.2, **Chapter** 2. In this chapter, based on the wavelet approaches, a predictive model between the accelerometer data and CAHAI score will be built using accelerometer data, collected over a 3 day period, (the physical movements are unknown behind the data) to predict the clinical summation CAHAI score (from 7 to 63). Instead of doing the 9-tasks to evaluate the clinical CAHAI score in hospital, the ultimate aim is that the stroke patients just need to wear the wrist-worn sensor AX3 and upload the accelerometer data to the server. Once the server finishes analyzing the data, the stroke patients will get their current summation CAHAI scores conveniently at home. Finally, once the CAHAI scores can be predicted from accelerometer data without the 9-tasks, the doctor's resources can be saved in hospital which means that government spending in this area will be reduced.

3.2 Methodology

Most recently, wrist-worn sensors have been used for stroke rehabilitation monitoring of patients in free-living environment (Tang et al., 2020). In the trial, 3-day accelerometer data were collected from both wrists (with a trial-wise annotation, i.e., CAHAI score), and for case (Tang et al., 2020) data analysis was performed using the sliding window approach. To reduce the data redundancy of the raw data, PCs score were extracted from each window (Tang et al., 2020). Then, a Gaussian Mixture Models (GMM) clustering approach was employed to learn the holistic trial-wise representation before developing the regression model. The method (Tang et al., 2020) suffered from the lack of annotation. The application of GMM clustering made it less feasible to use large data, and in (Tang et al., 2020) only 1% of training data were used due to high computational cost.

In our work, by analysing the nature of the paralysed/non-paralysed sides, we design stroke-rehab-driven features which can directly encode the long accelerometer sequence

(e.g., a trial with 3-day accelerometer data) into a very compact representation. The features are expected to emphasis the stroke-related behaviours while dealing with the irrelevant activities. Based on the proposed features, a predictive model that is adaptive to different subjects/time-slots can be developed using LMGP (Shi et al., 2012) for CAHAI score prediction. The **Figure** 3.1 shows a graphical representation of process in this chapter. It shows progress in stroke rehabilitation by using wearable sensor AX3 to predict the clinical CAHAI score.

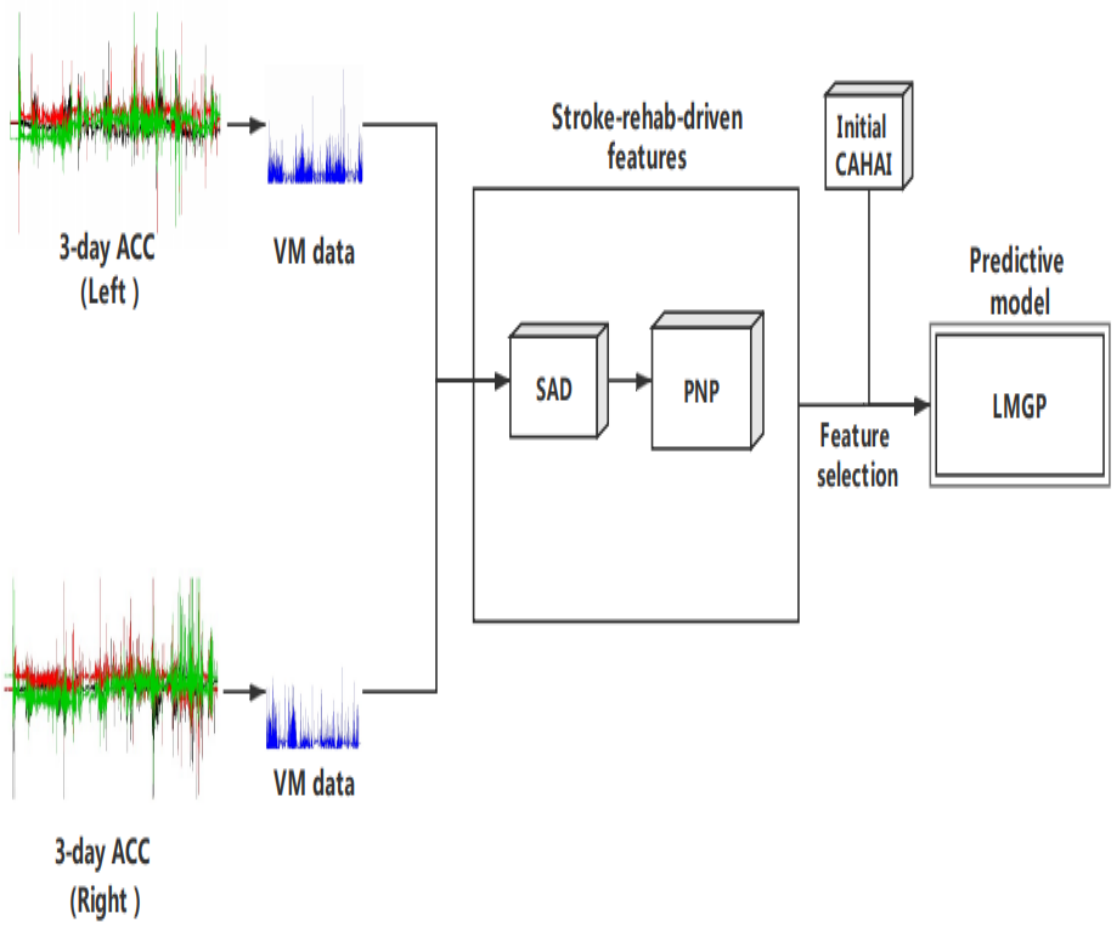


Figure 3.1: The pipeline of our automated stroke rehabilitation assessment system

3.2.1 The Proposed Stroke-Rehab-Driven Features

We aim to build a model that can map the 3-day time-series data to the CAHAI score. Different from other wearable-based behaviour analysis tasks (e.g., (Plötz et al., 2012)(Guan & Plötz, 2017)), the annotation here is inadequate. Even if we used the second-wise VM data, each trial still included roughly 3 days × 24h/day × 3600s/h

= 259200 samples (a.k.a. timestamps) with one annotation (i.e., CAHAI score). In contrast to the popular deep learning based human activity recognition approaches, which can be trained when with rich annotations (in frame-wise or sample-wise level), the lack of annotation makes it hard to learn effective representation directly (using machine/deep learning) from the raw data. Moreover, since the data were collected in free-living environments, and the 3 full days (per week) can be taken in weekdays or weekends, which may increase the intra-subject variability significantly, making it hard to model. To address these issues, domain knowledge driven feature engineering may play a major role in extracting compact and discriminant signatures.

For time-series analysis, wavelet analysis is a powerful tool to represent various aspects of non-stationary signals such as trends, discontinuities, and repeated patterns (Ayachi et al., 2016) (Donald B. Percival, 2000) (Preece et al., 2009), which is especially useful in signal compression or noise reduction. Given their properties, wavelet features were widely used in accelerometer-based daily living activity analytics (Ayachi et al., 2016). In this work, we used discrete wavelet transform (**DWT**) and discrete wavelet packet transform (**DWPT**) as feature extractors, based on which new features were designed to preserve the stroke rehabilitation-related information.

Commonly used wavelet features

In the discrete wavelet transform, as we discussed in **Section** 2.3.3, $\mathbf{W}_j$ represents **DWT** coefficients in the j-th decomposition scale. **DWT** can be written as $\mathbf{W} = \mathcal{W}\mathbf{X}$, where $\mathbf{W}$ is a column vector with length 2^j and $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_J, \mathbf{V}_J]^{\mathrm{T}}$, $\mathcal{W}$ is the orthonormal matrix which satisfies $\mathcal{W}^T\mathcal{W} = \mathbf{I}_N$ and contains different filters. Due to the orthonormality of **DWT**, which means that $\mathbf{X} = \mathcal{W}^{\mathrm{T}}\mathbf{W}$ and $\|\mathbf{X}\|^2 = \|\mathbf{W}\|^2$, $\|\mathbf{W}_j\|^2$ shows energy in the **DWT** coefficients with decomposition level j.

Now the energy preserving condition can be written as:

$$\|\mathbf{X}\|^2 = \|\mathbf{W}\|^2 = \sum_{j=1}^{J} \|\mathbf{W}_j\|^2 + \|\mathbf{V}_J\|^2, \tag{3.1}$$

where $\mathbf{X}$ is our **VM** data (the signal vector magnitude of accelerometer data; see **Section** 2.2.3) with length N, $j = 1, 2, ..., J$ is the discrete wavelet transform decomposition level. $\mathbf{W}_j$ denotes the detailed coefficient in scale j, and is a vector of length $N/2^j$ representing the differences in adjacent weighted averages from scale 1 to scale J. $\mathbf{V}_J$ denotes the approximated coefficients in the Jth level and has the same length as $\mathbf{W}_J$. Based on the decomposition, each $\|\mathbf{W}_j\|^2$ represents a special part of the energy in our **VM** data which relates to the certain frequency domain (Preece et al., 2009) (Donald B. Percival, 2000).

In Donald B. Percival (2000), the sample variance through the **DWT** is defined, the

sample variance can be decomposed as:

$$\widehat{\sigma}^2_{\mathbf{X}} = \frac{1}{N} \|\mathbf{W}\|^2 - \overline{X}^2 = \sum_{j=1}^{J} \frac{\|\mathbf{W}_j\|^2}{N}. \tag{3.2}$$

Where $\overline{X}$ is the mean of X, the term $\frac{\|\mathbf{W}_j\|^2}{N}$ represents the sample variance in our **VM** data **X** at different levels of **DWT** decomposition.

There are many wavelet features (e.g., Preece et al. (2009)) for the classification of dynamic activities from accelerometer data using **DWT**. On this basis, we extract the features from the energy preserving condition and sample variance mentioned previously.

We aim to look for the features which imply the recovery level among the stroke patients (see **Section** 3.2.1). Now, we define the features in the j-th level discrete wavelet transform and discrete wavelet packet transform :

$$\mathbf{SSD}_j = \frac{\|\mathbf{W}_j\|^2}{N/2^j} = 2^j \frac{\|\mathbf{W}_j\|^2}{N}.$$

For the detailed coefficients $\mathbf{W}_j$ at decomposition level j, $\|\mathbf{W}_j\|^2$ presents its energy and the raw data with length N. Hence the physical explanation of $\mathbf{SSD}_j$ is that it stands for the point energy at the decomposition level j. Moreover, from the **Equation** (3.2), $\frac{\|\mathbf{W}_j\|^2}{N}$ represents the sample variance at the decomposition level j, $\mathbf{SSD}_j$ also has properties of both the energy preserving condition and the sample variance in wavelet analysis with constant 2^j .

Comparing with $\mathbf{SSD}_j$ (sums of the square values divided by the constant $N/2^j$ at scales j), we define other features called $\mathbf{SAD}_j$, which is calculated by evaluating the sums of the absolute values divided by the constant $N/2^j$ at scale j:

$$\mathbf{SAD}_j = \frac{\|\mathbf{W}_j\|_1}{N/2^j} = 2^j \frac{\|\mathbf{W}_j\|_1}{N}.$$

After we check the correlation between the important wavelet feature **PNP** (**Section** 3.2.1) and CAHAI score, the branch of features **PNP** using **SAD** based perform better than those using **SSD** based in **Table** 3.1. Hence we consider the commonly used feature $\mathbf{SAD}_j$ in this paper.

-	Acute Patients				Chronic Patients			
Scale (k)	PNP_k^1 (SSD)	PNP_k^2 (SSD)	PNP_k^1 (SAD)	PNP_k^2 (SAD)	PNP_k^1 (SSD)	PNP_k^2 (SSD)	PNP_k^1 (SAD)	PNP_k^1 (SAD)
k=1.1	0.60	-0.65	0.68	-0.70	0.45	-0.45	0.56	-0.56
k=1.2	0.60	-0.66	0.69	-0.71	0.46	-0.45	0.57	-0.56
k=1.3	0.63	-0.69	0.70	-0.72	0.49	-0.48	0.58	-0.57
k=1.4	0.62	-0.68	0.69	-0.71	0.47	-0.47	0.57	-0.57
k=2	0.65	-0.69	0.69	-0.71	0.45	-0.45	0.56	-0.55
k=3	0.63	-0.67	0.67	-0.68	0.39	-0.38	0.53	-0.52
k=4	0.59	-0.63	0.60	-0.63	0.31	-0.30	0.48	-0.47
k=5	0.46	-0.50	0.49	-0.52	0.29	-0.27	0.43	-0.42
k=6	0.32	-0.38	0.35	-0.38	0.20	-0.16	0.35	-0.34
k=7	0.16	-0.19	0.19	-0.20	0.13	-0.10	0.25	-0.24

Table 3.1: The correlation between SAD and SSD based wavelet features and CAHAI score for acute and chronic patients .

In our analysis, we assume the discrete wavelet decomposition level $J = 7$ which is the same level as in Sekine et al. (1998) and contains enough low-frequency component as the stroke patients' movement. The frequency domain with seven scales is shown in **Table** 3.2:

	Scale 7	Scale 6	Scale 5
Frequency	0.0078hz-0.0156hz	0.0156hz - 0.0312hz	0.0312hz - 0.0625hz
	Scale 4	Scale 3	Scale 2
Frequency	0.0625hz - 0.125hz	0.125hz - 0.25hz	0.25hz - 0.50h
	Scale 1		
Frequency	0.50hz - 1hz		

Table 3.2: The frequency domain from scale 1 to scale 7 by using DWT.

So far, we have decomposed the **VM** data $\mathbf{X}$ to get $\mathbf{W}_1$, $\mathbf{W}_2$, ... , $\mathbf{W}_7$ using **DWT**. Since the frequency domain at scale 1 is so wide (0.50hz - 1hz), it is better to divide it into smaller one, then using **DWPT** in **Section** 2.3.4, we can further decompose $\mathbf{W}_1$ into $\mathbf{W}_{3.4}$, $\mathbf{W}_{3.5}$, $\mathbf{W}_{3.6}$ and $\mathbf{W}_{3.7}$ which are the results of the 3-rd stage of **DWPT**, each coefficient vector with length $N/2^3$ has the same dimension as the coefficients in the third level of **DWT** decomposition, that is

$$\|\mathbf{X}\|^2 = \|\mathbf{W}\|^2 = \|\mathbf{W}_{3.4}\|^2 + \|\mathbf{W}_{3.5}\|^2 + \|\mathbf{W}_{3.6}\|^2 + \|\mathbf{W}_{3.7}\|^2 + \sum_{j=2}^{J} \|\mathbf{W}_j\|^2 + \|\mathbf{V}_J\|^2.$$

Now we have coefficients at 10 decomposition scales by using **DWT** and **DWPT**: $\mathbf{W}_{3.4}$, $\mathbf{W}_{3.5}$, $\mathbf{W}_{3.6}$, $\mathbf{W}_{3.7}$, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{W}_4$, $\mathbf{W}_5$, $\mathbf{W}_6$ and $\mathbf{W}_7$. Based on these detailed coefficients, we define the commonly used wavelet features again:

$$\begin{aligned}
\textbf{Scale 1.1} &: SAD_{1.1} = \frac{\|\mathbf{W}_{3.4}\|_1}{N/2^3} = 2^3\frac{\|\mathbf{W}_{3.4}\|_1}{N}, \\
\textbf{Scale 1.2} &: SAD_{1.2} = \frac{\|\mathbf{W}_{3.5}\|_1}{N/2^3} = 2^3\frac{\|\mathbf{W}_{3.5}\|_1}{N}, \\
\textbf{Scale 1.3} &: SAD_{1.3} = \frac{\|\mathbf{W}_{3.6}\|_1}{N/2^3} = 2^3\frac{\|\mathbf{W}_{3.6}\|_1}{N}, \\
\textbf{Scale 1.4} &: SAD_{1.4} = \frac{\|\mathbf{W}_{3.7}\|_1}{N/2^3} = 2^3\frac{\|\mathbf{W}_{3.7}\|_1}{N}, \\
\textbf{Scale j} &: SAD_j = \frac{\|\mathbf{W}_j\|_1}{N/2^j} = 2^j\frac{\|\mathbf{W}_j\|_1}{N}, \qquad j = 2,3,4,5,6,7.
\end{aligned}$$

There are 10 features which provide reliable and valid information (corresponding to more frequency domains) from different frequency domains. The frequency domain of these features, among 10 scales, is listed in **Table** 3.3:

	Scale 1.1	Scale 1.2	Scale 1.3
Frequency	0.5hz - 0.625hz	0.625hz - 0.75hz	0.75hz - 0.875hz
	Scale 1.4	Scale 2	Scale 3
Frequency	0.875hz - 1hz	0.25-0.50hz	0.125hz - 0.25hz
	Scale 4	Scale 5	Scale 6
Frequency	0.0625hz - 0.125hz	0.0312hz - 0.0625hz	0.0156hz - 0.0312hz
	Scale 7		
Frequency	0.0078hz - 0.0156hz		

Table 3.3: The frequency domain from scale 1.1 to scale 7 by using **DWPT** and **DWT**.

We now have 10 SAD_j from **DWT** and **DWPT** at 10 scales, where each scale correlates to different frequency information (See **Table** 3.3). Comparing with SAD_j, the features in Preece et al. (2009) are extracted from accelerometer data by using **DWT**, defined as data power measurements and calculated as the sum of the squared (absolute) detailed coefficients at different **DWT** scales. Those features calculate the total energy of the accelerometer data in specific **DWT** decomposition scales. SAD_j follow the same principle with data power measurements and calculate the sum of the absolute detailed coefficients at 7 scales by using **DWT** at first. Moreover, **DWPT** will be used in scale 1 to separate it into 4 scales. On this basis, a constraint is already added to it. To be more specific, the sum of the absolute detailed coefficients in certain scales will be divided

by the number of detailed coefficients. Instead of calculating the total energy of detailed coefficients in certain **DWT** decomposition scales, the point energy will be calculated with the constraint.

New proposed Features

From commonly used wavelet features in **Section** 3.2.1, according to the accelerometer data in stroke patients, SAD_j is a type of point energy in decomposition scale j. In our first naive hypothesis, the larger value of SAD_j indicates more activities, meaning those stroke patients tend to have a better recovery level. However, when we investigate those features against clinical assessed CAHAI scores, we find that this is not true. This is mainly due to the large variation in patients' life style.

Patients have different habits in their daily life, so the first problem is that some patients have a good recovery level, but tend to be inactive and have a somewhat sedentary lifestyle. We call this type of patients as "can do but doesn't want to". There is another type of patients called "couldn't do but wants to" who are very active but have a low recovery level. For example, Patients la027 and la038 visit the hospital 8 times totally, Patient la027 has a significant recovery during the treatment, the clinical assessed CAHAI score rises from 13 to 57 within 8 visits. However, Patient la038 shows a low recovery level, who visits the hospital 8 times as well, with clinical assessed CAHAI score only from 11 to 22. It can be seen that from **Figure** 3.2 (the **VM** data between Patient la012 and Patient la038 of paralysed side in last visit), Patient la027 (CAHAI 57) spends the half time staying still even with good recovery level, while Patient la038 (CAHAI 22) keeps moving frequently but has a low recovery level. Our objective is to eliminate the influence of "can do but doesn't want to" (e.g. Patient la027) and "couldn't do but wants to" (e.g. Patient la038) among patients.

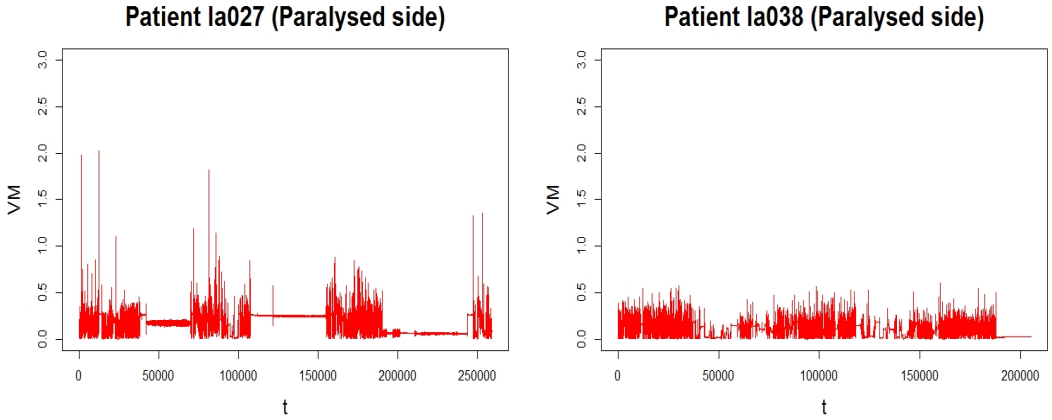


Figure 3.2: The **VM** data between two types of patients in the paralysed side. Left panel: "can do but doesn't want to"(la027), Right panel: "couldn't do but wants to"(la038).

We now check the value of SAD_j from 10 scales between Patient la027 and Patient la038 of paralysed side in the 8-th visit (from **Figure** 3.3,also see **Table** 3.12 in **Appendix** 3.5). The value of SAD_j in different scales is very similar for these two patients. In other words, features SAD_j are hard to remove from the effect of "can do but doesn't want to" and "couldn't do but wants to". Therefore, it is not suitable to apply our previous hypothesis that patients with good recovery usually have big values of SAD_j and patients with bad recovery have small value of SAD_j.

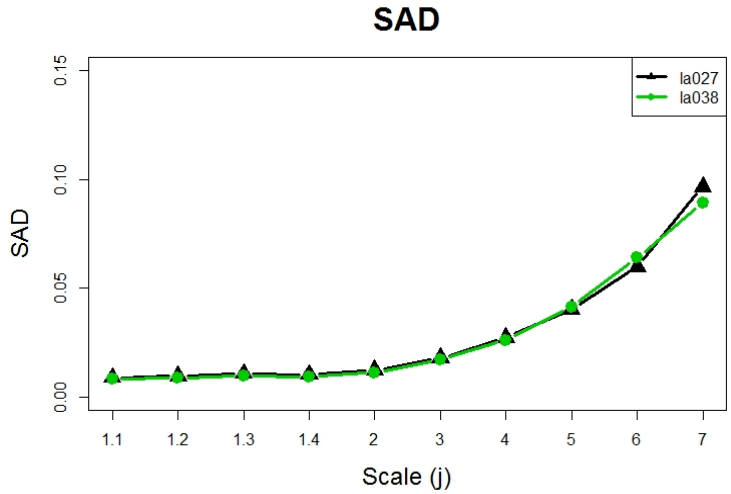


Figure 3.3: The SAD_j between two types of patients in the paralysed side. 10-dimensional **SAD** features were extracted from the paralysed side between two types of patients (with different CAHAI scores); They exhibit similar patterns, indicating the necessity of developing more informative stroke-related features.

It is known that a very important property of stroke patients is that their hands can be allocated to either the paralysed side or the non-paralysed side. The second problem we encounter when using SAD_j is that, on the paralysed side, it only provides the information about the degree of disability. This means that SAD_j when used only on the non-paralysed side, gives us no information about disability level and is, threrfore, not useful when used in this way.

So far, due to the complexity of all patients' data, the commonly used feature SAD_j has great limitations, which are reflected mainly the above two problems. In summary, it is hard to build a good connection with the recovery levels (clinical assessed CAHAI score) in paralysed side since it is very difficult to distinguish. Also SAD_j is of very limited use in the non-paralysed side. We now have a problem of finding a way to remove the effect from the two kinds of patients above and to use SAD_j properly in the non-paralysed side.

The key point is to find new features. **Figure** 3.4 presents the plots of CAHAI scores against SAD_2 (high frequency domain) and SAD_6 (low frequency domain) for both

paralysed and non-paralysed hands (the plots for other scales are rather similar). See also **Figures** 3.16, 3.17, 3.18 and 3.19 in **Appendix** 3.5. They show poor correlation between CAHAI scores and SAD_j, meaning the latter provides little information in terms of predicting CAHAI scores, i.e., the recovery level.

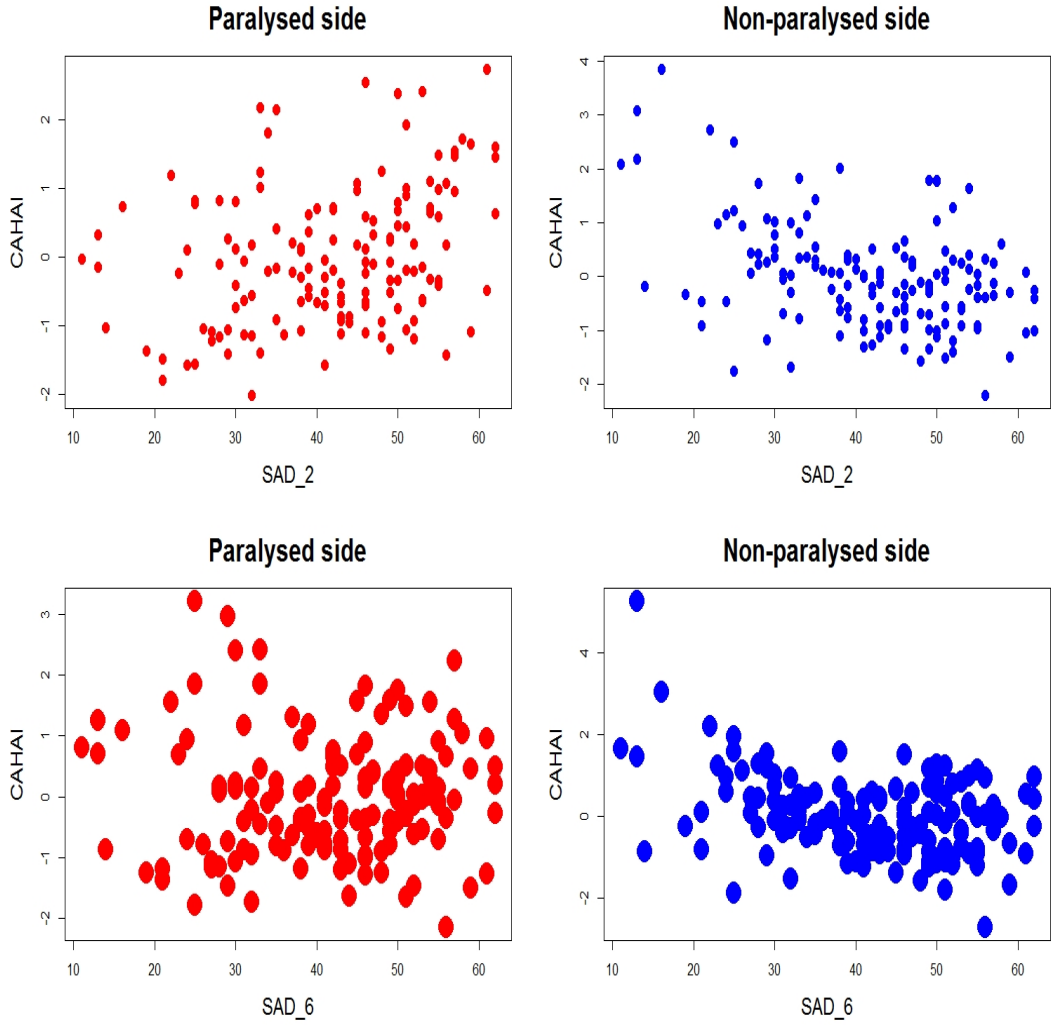


Figure 3.4: The typical SAD_j at scale 2 and scale 6 against CAHAI in both sides. Left panel: paralysed side. Right panel: non-paralysed side.

We continue to use Patient la027 and Patient la038 as examples to illustrate the different usage rate on the paralysed side and the non-paralysed side. As we discussed before, Patient la027 has a good recovery level and Patient la038 has a low recovery level. **Figures** 3.5 (**VM** data between Patient la027 and Patient la038 in last visit) show that Patient la027 uses both hands equally and Patient la038 tends to use the non-paralysed side more.

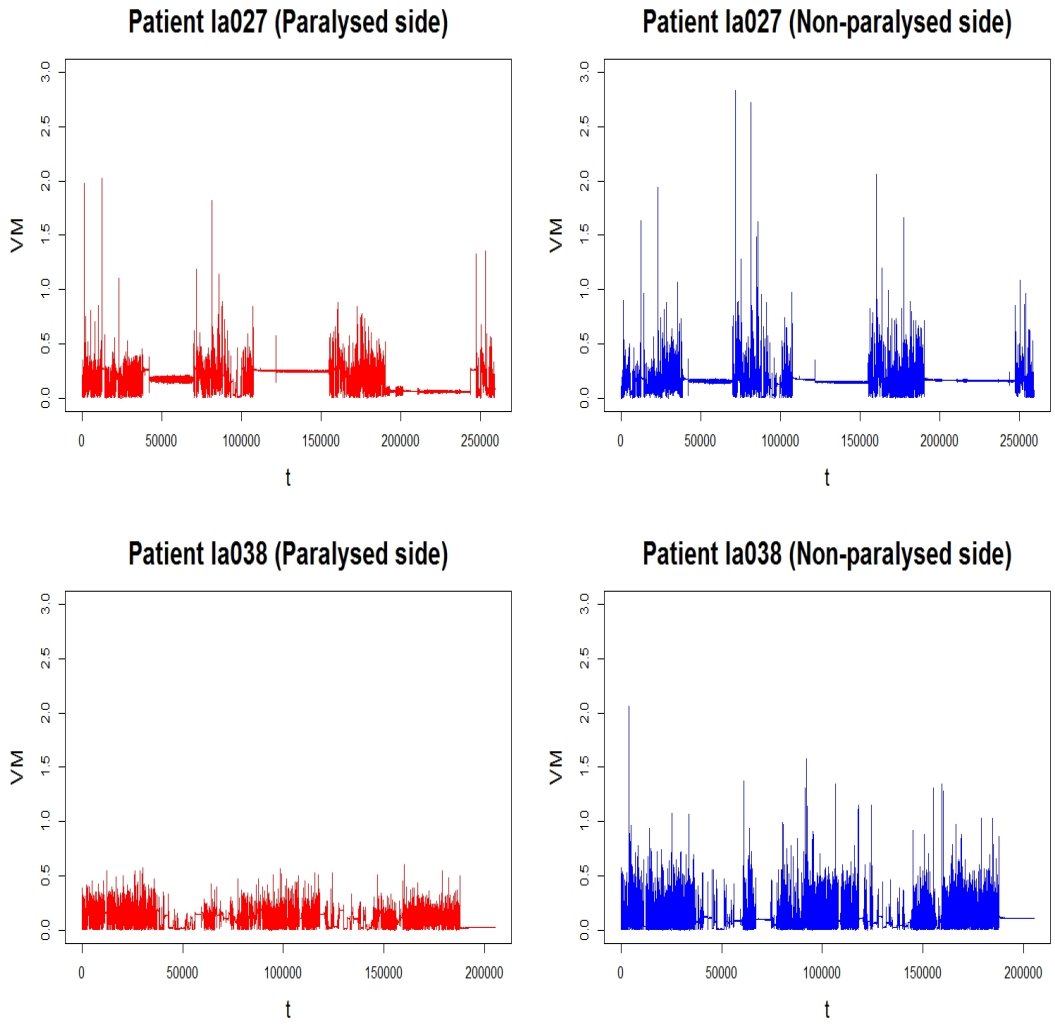


Figure 3.5: The **VM** data between both hands in Patient la027 (Top) and Patient la038 (Bottom). Left panel: paralysed side, Right panel: non-paralysed side.

We now check the values of SAD_j from 10 scales of these two patients in the 8-th visit from **Figure** 3.6, (see also **Table** 3.13 in **Appendix** 3.5). On the one hand, the values of SAD_j between the paralysed side and the non-paralysed side are extremely similar for Patient la027. But, on the other hand, the values of SAD_j with the non-paralysed side are obviously bigger than those with the paralysed side for Patient la038. This indicates that patients with low recovery level (eg. Patient la038) may be more likely to use their non-paralysed side in daily life, but patients who have good recovery will use their both hands equally. This motivates us to investigate the connection between recovery level (clinical assessed CAHAI score) and the ratio of SAD_j's between two hands. Consequently, we find some new features which are well associated with the recovery levels.

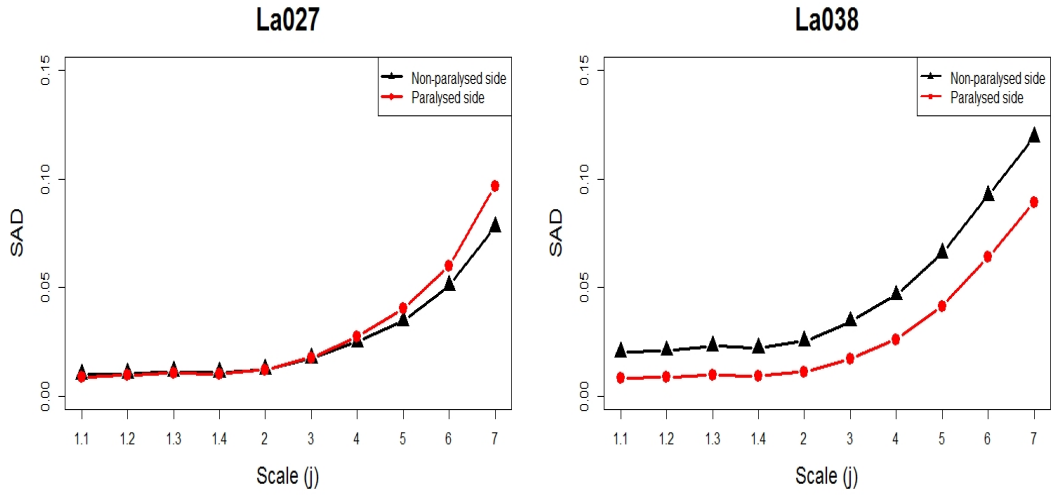


Figure 3.6: The SAD_j between two types of patients in both sides. Left panel: Patient la027, Right panel: Patient la038. **SAD** features from the non-paralysed side may contain discriminant information for stroke-rehab modelling.

Specifically, we find two types of new features, which are defined by

$$PNP_k^1 = \frac{SAD_k^p}{SAD_k^{np}}$$
$$PNP_k^2 = \frac{SAD_k^{np} - SAD_k^p}{SAD_k^{np} + SAD_k^p},$$

where $k = 1.1, 1.2, 1.3, 1.4, 2, 3, 4, 5, 6, 7$ corresponds to the 10 scales, while p and np refer to the paralysed side and non-paralysed side respectively. where $k = 1.1, 1.2, 1.3, 1.4, 2, 3, 4, 5, 6, 7$ corresponds to the scale 1.1, scale 1.2, scale 1.3, scale 1.4, scale 2, scale 3, scale 4, scale 5, scale 6 and scale 7 in our previous features. The term SAD_k^p and SAD_k^{np} refer to the features at the k-th scale of decomposition in the paralysed side and non-paralysed side, respectively.

Both $\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$ are the energy rates between paralysed side and non-paralysed side. If $\mathbf{PNP_k^1}$ gets smaller and $\mathbf{PNP_k^2}$ becomes bigger, the patient will use the paralysed side less frequently. In other words, comparing with the usage rate of non-paralysed side, if patients use more paralysed side, they will have a good recovery level. From SAD_j of paralysed side in **Figure** 3.3, it is hard to distinguish Patient la038 to la027 although their recovery levels are very different. However, from **Figure** 3.7 (see also **Table** 3.14 in **Appendix** 3.5) $\mathbf{PNP_k^1}$ of Patient la027 with a good recovery level, who uses both paralysed side and non-paralysed side equally, is bigger than that of Patient la038 who has a bad recovery level, and tends to use the non-paralysed side because the paralysed side is hard to move due to the bad recovery level. In contrast, $\mathbf{PNP_k^2}$ of Patient la038

is bigger than that of Patient la027 since Patient la027 has a good recovery level than Patient la038. Hence, these two new features, $\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$, will provide sufficient information to predict the recovery levels after stroke and will make up the shortage of $\mathbf{SAD_j}$ which is commonly used in practice.

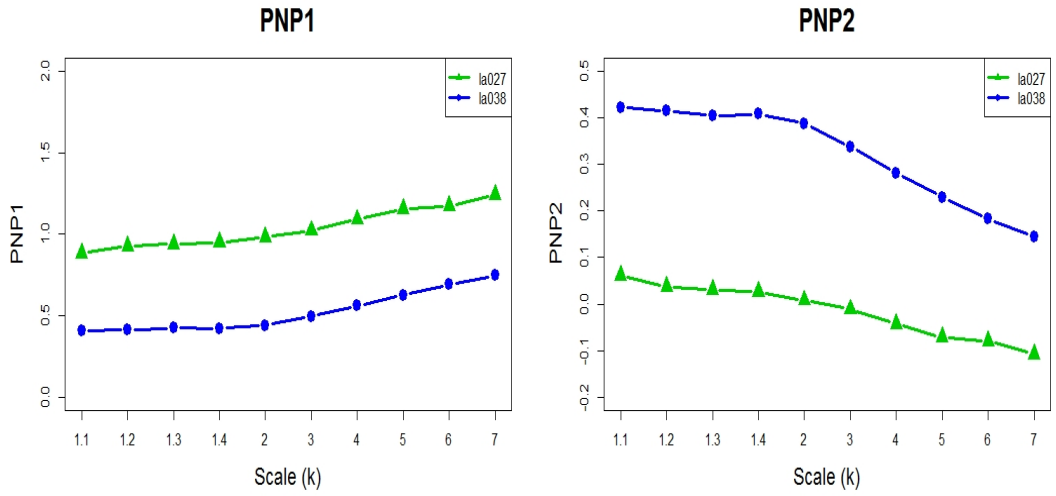


Figure 3.7: Two proposed **PNP** representations for two types of patients in both sides. This can provide discriminant information in distinguishing the patients with different recovery levels (clinical CAHAI score).

Note that, because the two new features ($\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$) combine the previous commonly used features $\mathbf{SAD_j}$ in both the paralysed and non-paralysed sides, where the decomposition scale $j = k$, they share the same frequency domain at 10 scales with $\mathbf{SAD_j}$ in **Table** 3.3. Therefore, based on previous features $\mathbf{SAD_j}$, we have 10 SAD_j for each paralysed side and non-paralysed side, i.e., 20 SAD_j. After obtaining these two kinds of new features $\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$, 20 new features have been created. These will be used to build our predictive model. Although $\mathbf{SAD_j}$ are better at describing a single hand in different frequency domains, they do not correlate well with recovery level as measured in the clinical assessed CAHAI score. However, they do have some merit and we will use them as candidate covariates which will be discussed in the next step.

Now we have totally 40 different features which are listed in **Table** 3.4:

Feature type	Dimension	feature entries for each type
$\mathbf{SAD^p}$	10	$SAD^p_{1.1}$, $SAD^p_{1.2}$, $SAD^p_{1.3}$, $SAD^p_{1.4}$, SAD^p_2, SAD^p_3, ... , SAD^p_7
$\mathbf{SAD^{np}}$	10	$SAD^{np}_{1.1}$, $SAD^{np}_{1.2}$, $SAD^{np}_{1.3}$, $SAD^{np}_{1.4}$, SAD^{np}_2, SAD^{np}_3, ... , SAD^{np}_7
$\mathbf{PNP^1}$	10	$PNP^1_{1.1}$, $PNP^1_{1.2}$, $PNP^1_{1.3}$,$PNP^1_{1.4}$, PNP^1_2, PNP^1_3, ... , PNP^1_7
$\mathbf{PNP^2}$	10	$PNP^2_{1.1}$, $PNP^2_{1.2}$, $PNP^2_{1.3}$,$PNP^2_{1.4}$, PNP^2_2, PNP^2_3, ... , PNP^2_7

Table 3.4: The wavelet features at 10 scales.

3.2.2 Predictive models

Based on the proposed wavelet features, we aim to develop predictive models that can map features to the CAHAI score. Although we reduce data redundancy significantly via feature design, there still exist data noises, which may encode irrelevant information. It is crucial to develop robust a mechanism to select the most relevant features, and here we use a popular feature selection linear model (LASSO) to model the nonlinear random effects in the longitudinal study. We also propose to use a Gaussian Process (GP) regression model.

It is worth noting that our model will also take advantage of the medical history information (i.e., CAHAI score during the first visit) to predict CAHAI scores for the remaining 7 weeks (i.e., week 2 - week 8). From the perspective of practical application, the CAHAI score from the initial week (referred to as *ini*) may be used as an important normalisation factor for different individuals.

The linear fixed-effects model

Since there may exist some redundant or irrelevant features for the prediction task, we first propose to use the simple linear model LASSO (i.e., Least Absolute Shrinkage and Selection Operator) with a feature selection mechanism. A brief description of LASSO can be found in **Appendix** 3.5.

Given the 41-dimensional data (40 wavelet features and 1 CAHAI score from the initial week), we first standardise the data using z-norm, and each feature entry x_k will be normalised as:

$$x_k^{new} = \frac{x_k - \overline{x}}{s_k},$$

where $\overline{x}$ and s_k are the mean, and standard deviation of the k^{th} feature. Based on the aforementioned model, namely LASSO, useful features can be selected, based on which prediction model can be developed. For simplicity, we first use linear model to predict the target CAHAI score y_i:

$$y_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{\epsilon}_i,\ \epsilon_i \sim N(0, \sigma^2), \tag{3.3}$$

where i stands for the i^{th} trial from patients (out of all patients during week 2 - week 8), $\boldsymbol{x}_i$ represents the selected feature vector, and $\boldsymbol{\beta}$ is the model parameter vector to be estimated, and ϵ_i is the random noise term.

Longitudinal mixed-effects model with Gaussian process prior(LMGP)

It is simple to use the linear model for CAHAI score prediction. However, it ignores the heterogeneity among subjects in this longitudinal study. To model this, we proposed to use a nonlinear mixed-effects model (Shi et al., 2012), which consists of fixed-effects

part and random-effects part. Specifically, the random-effects part contributes mainly to modelling the heterogeneity, making the prediction process subject/time-adaptive for longitudinal studies. The longitudinal mixed-effects model with Gaussian Process prior (LMGP) is defined as follows:

$$y_{i,j} = \boldsymbol{x}_{i,j}^{\mathrm{T}}\boldsymbol{\beta} + g(\boldsymbol{\phi}_{i,j}) + \epsilon_{i,j},\ \epsilon_{i,j} \sim N(0, \sigma^2), \tag{3.4}$$

where i,j stand for the i^{th} patient at the j^{th} visit (from week 2 to week 8); $\epsilon_{i,j}$ refers to the independent random error and σ^2 is its variance; In **Equation** (3.4), $\boldsymbol{x}_{i,j}^{T}\boldsymbol{\beta}$ is the fixed-effects part and $g(\boldsymbol{\phi}_{i,j})$ represents the nonlinear random-effects part; the latter can be modelled using a non-parametric Bayesian approach with a GP prior (Shi et al., 2012).

It is worth noting that in LMGP the fixed-effects part $\boldsymbol{x}_{i,j}^{\mathrm{T}}\boldsymbol{\beta}$ explains a linear relationship between input features and CAHAI, while the random-effects part $g(\boldsymbol{\phi}_{i,j})$ is used to explain the variability caused by differences among individuals or time slots during different weeks. By considering both parts, LMGP provides a solution of personalised modelling for this longitudinal data analysis. In LMGP, it is important to select input features to model both parts, and we refer them to as fixed-effects features and random-effects features, respectively. The effect of the fixed-effects features will be studied in **Section** 3.3.2.

For LMGP training, we first ignore the random-effects part, and only optimise the parameters $\hat{\boldsymbol{\beta}}$ of the fixed-effects part (via ordinary least squares, OLS);

With estimated parameters $\hat{\boldsymbol{\beta}}$, the residual $r_{ij} = y_{ij} - \boldsymbol{x}_{i,j}^{\mathrm{T}}\hat{\boldsymbol{\beta}} = g(\boldsymbol{\phi}_{i,j}) + \epsilon_{i,j}$ can be calculated, from which we can model the random-effects

$$g(\boldsymbol{\phi}_{i,j}) \sim GP(0, K(\cdot,\cdot;\boldsymbol{\theta})).$$

In this paper we choose $K(\cdot,\cdot;\boldsymbol{\theta})$ as the following three different kernels (linear, squared exponential and rational quadratic), and here we take the squared exponential as an example. The squared exponential (covariance) kernel function is defined as : $K\left(\boldsymbol{\phi},\boldsymbol{\phi}';\boldsymbol{\theta}\right) = v_0 \exp\left\{-d(\boldsymbol{\phi},\boldsymbol{\phi}')/2\right\}$ where $d(\boldsymbol{\phi},\boldsymbol{\phi}') = \sum_{q=1}^{Q} w_q \left(\phi_{i,j,q} - \phi'_{i,j,q}\right)^2$ is an extended distance between $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$. It involves the hyper-parameters $\boldsymbol{\theta} = (v_0, w_1, ..., w_Q)$. In a Bayesian approach, we may choose the value of those parameters based on prior knowledge. It is, however, a difficult task due to the large dimension of $\boldsymbol{\theta}$. We used, therefore, an empirical Bayesian method.

The training procedure includes two steps. (I) Estimate $\boldsymbol{\beta}$ and σ in **Equation** (3.3); (II) Estimate the values of the hyper-parameters $\boldsymbol{\theta}$ by an empirical Bayesian method, i.e. maximise the marginal likelihood from $\boldsymbol{r}_i \sim N(\boldsymbol{0}, \boldsymbol{C}_i + \sigma^2\boldsymbol{I})$ for $i = 1, \ldots, n$, where $\mathbf{C}_i \in \mathbb{R}^{J\times J}$ is the covariance matrix of $g(\cdot)$, and its element is defined by $K(\phi_{i,j}, \phi_{i,j'};\boldsymbol{\theta})$. To obtain a more accurate results, an iterative method may be used, except in the initial

step, the error item in (3.3) used in step I is replaced by

$$\boldsymbol{\epsilon}_i = (\epsilon_1, \ldots, \epsilon_J) \sim N(\mathbf{0}, \boldsymbol{C}_i + \sigma^2 \boldsymbol{I}))$$

where all the parameters are evaluated by using the values obtained in the previous iteration.

Calculation of the prediction is relatively easy. The posterior distribution of $g(\boldsymbol{\phi}_i)$ is a multivariate normal with mean $\mathbf{C}\left(\mathbf{C}+\sigma^2\mathbf{I}\right)^{-1}\boldsymbol{r}_i$ and the variance $\sigma^2\mathbf{C}\left(\mathbf{C}+\sigma^2\mathbf{I}\right)^{-1}$.

The fitted value can therefore be calculated by the sum of $\boldsymbol{x}_{i,j}^T\hat{\boldsymbol{\beta}}$ and the above posterior mean. The variance can be calculated accordingly. A detailed description can be found in Shi & Choi (2011).

3.3 Experimental Evaluation

In this section, several experiments were designed to evaluate the proposed features as well as the proposed prediction system. The patients were split into two groups according to the nature of the disease, i.e., the acute patient group (26 subjects) and the chronic patient group (33 subjects), and experiments were conducted on both groups separately.

Specifically for each group, leave-one-patient-out cross validation(LOPO-CV) was applied. That is, for a certain group (acute or chronic) with n subjects, in each iteration 1 subject was used as test set while the rest $n-1$ subjects were used for training. This procedure was repeated n times to test all the n subjects and average prediction performance (i.e., the mean predicted CAHAI) was reported.

Since CAHAI score prediction is a typical regression problem, we used the root mean square error (RMSE) from the test dataset based on the leavel-one-patient-out cross validation as the evaluation metric, and lower mean RMSE values indicate better performance.

3.3.1 Evaluation of commonly used wavelet features and new proposed features

Firstly, we treat 40 wavelet features and the initial CAHAI score as candidate covariates and check the correlation between these covariates and the CAHAI score in both acute and chronic patients. It can be seen, from the correlation heat map in **Figures** 3.9 and 3.11, that many features are highly correlated to the CAHAI score. We also calculate the numerical value of the correlation between $\mathbf{SAD_j}$ (both paralysed side and non-paralysed side), $\mathbf{PNP_k^1}$, $\mathbf{PNP_k^2}$ and clinical assessed CAHAI score at 10 decomposition scales respectively. Two partial scatter plots show the typical plots of $\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$ against CAHAI score as examples in **Figures** 3.8 and 3.10 (See also other scatter plots at different scales from **Figures** 3.20, 3.21, 3.22 and 3.23 in **Appendix** 3.5.). **Tables** 3.5 and 3.6

show that the correlation between new features $\mathbf{PNP_k^1}$, $\mathbf{PNP_k^2}$ and CAHAI score is much bigger than that between $\mathbf{SAD_j}$ in paralysed side, non-paralysed side and CAHAI score respectively. It means that new features $\mathbf{PNP_k^1}$ and $\mathbf{PNP_k^2}$ are more informative when we want to use the AX3 data to predict the CAHAI scores. The use of $\mathbf{SAD_j}$ alone is not enough. Another finding is that features from scale 1.1 to scale 5 have high correlation with the CAHAI score, but those from scale 6 and scale 7 show that they are not very important when linking to the CAHAI score.

Acute patients

Acute patients	SAD_j^p	SAD_j^{np}	PNP_j^1	PNP_j^2
Scale 1.1	-0.4130161	0.3199404	0.6806053	-0.7029202
Scale 1.2	-0.4191479	0.326648	0.6863952	-0.708192
Scale 1.3	-0.4270149	0.3221009	0.6967945	-0.7153064
Scale 1.4	-0.423402	0.3255395	0.6918816	-0.7113375
Scale 2	-0.4198529	0.3092206	0.691824	-0.7075939
Scale 3	-0.4202473	0.273319	0.6660961	-0.6816724
Scale 4	-0.4273795	0.2028392	0.6047307	-0.6252109
Scale 5	-0.4155057	0.09576458	0.4893035	-0.5159011
Scale 6	-0.3718157	-0.01252754	0.3494943	-0.3754041
Scale 7	-0.2998513	-0.1002001	0.1896564	-0.198969

Table 3.5: The correlation between wavelet features and CAHAI score for acute patients .

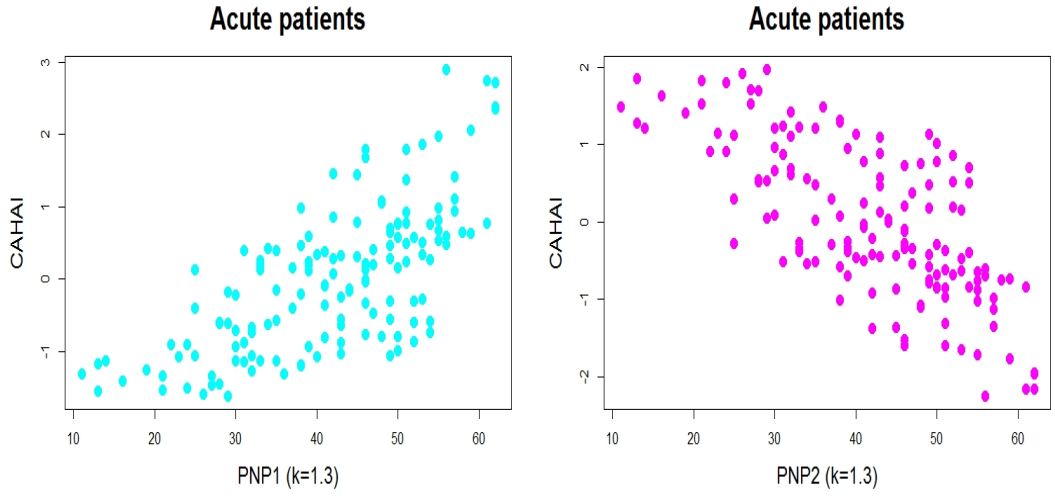


Figure 3.8: The typical proposed feature against CAHAI with acute patients' data.

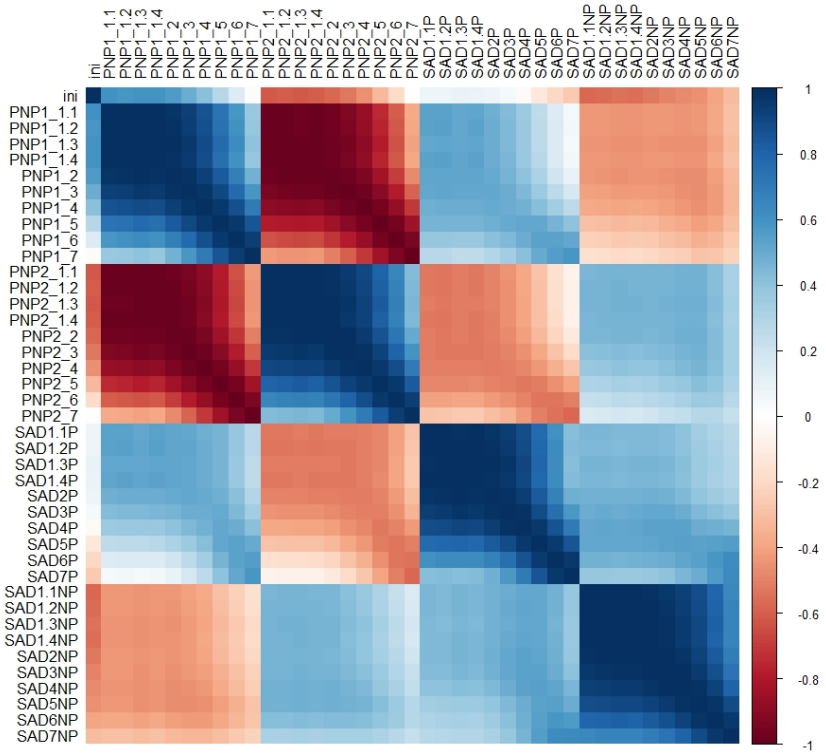


Figure 3.9: The correlation test among the variables for the acute patients.

Chronic patients

Chronic patients	SAD_j^p	SAD_j^{np}	PNP_j^1	PNP_j^1
Scale 1.1	0.22420	0.49368	0.56465	-0.55679
Scale 1.2	0.23819	0.50334	0.56783	-0.56168
Scale 1.3	0.22528	0.51023	0.57542	-0.57023
Scale 1.4	0.23637	0.51025	0.5731	-0.56856
Scale 2	0.23314	0.49950	0.55887	-0.55104
Scale 3	0.24846	0.49505	0.52619	-0.51872
Scale 4	0.26193	0.49521	0.47624	-0.46743
Scale 5	0.26867	0.50344	0.42633	-0.41690
Scale 6	0.26879	0.48327	0.34837	-0.33641
Scale 7	0.27582	0.45062	0.25380	-0.24060

Table 3.6: The correlation between wavelet features and CAHAI score for chronic patients .

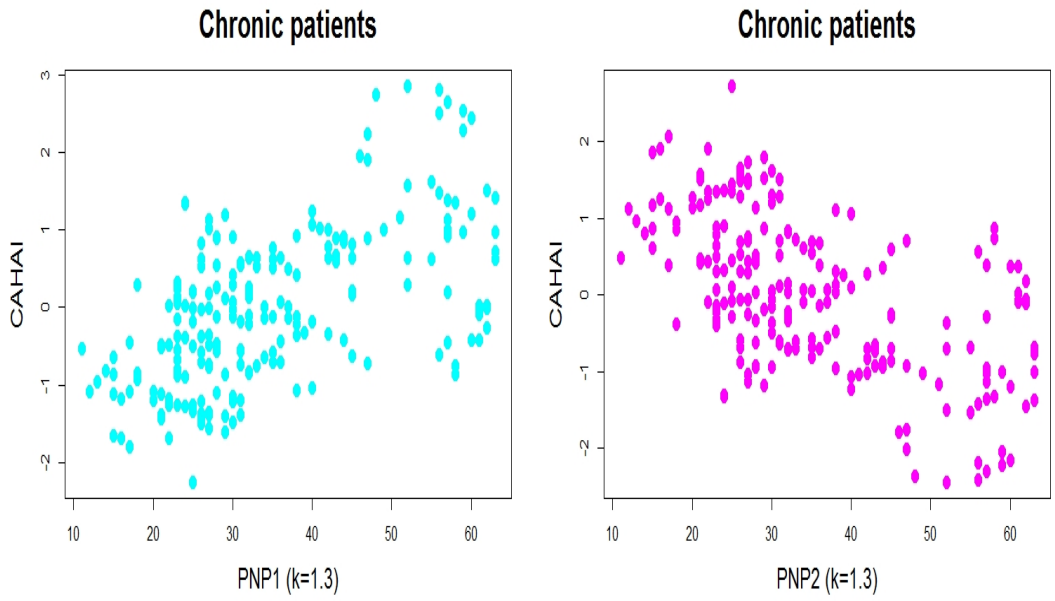


Figure 3.10: The typical proposed feature against CAHAI with chronic patients' data.

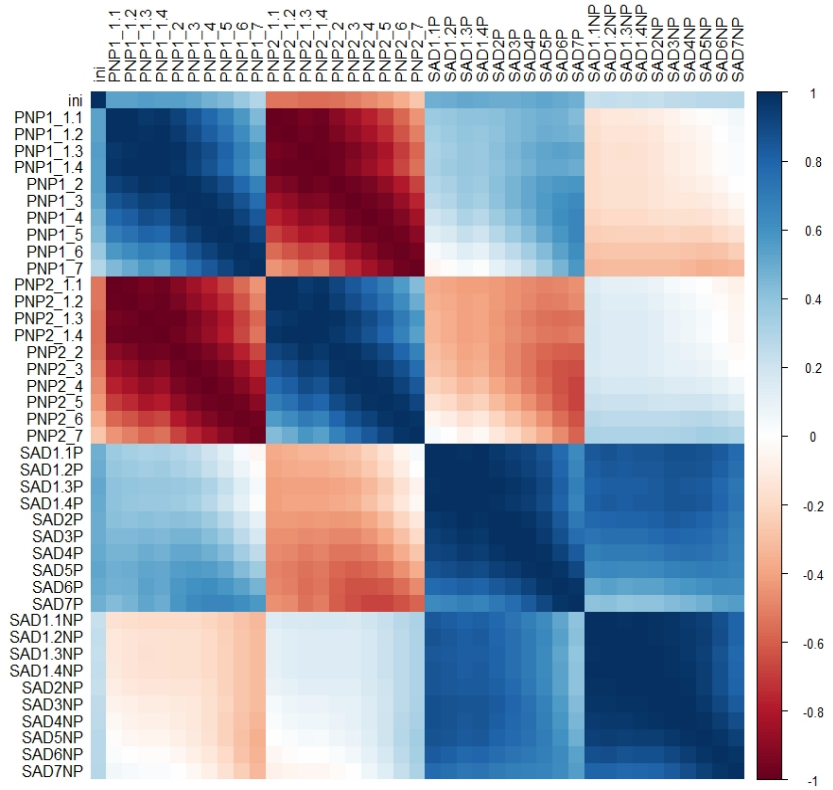


Figure 3.11: The correlation test among the variables in chronic patients.

According to the observations, it is obvious that within each feature types, there may exist a high-level of feature redundancy, and it is necessary to select the most relevant feature subsets. For acute and chronic patient groups, the optimal feature subset may vary due to the different movement patterns (e.g., on paralysed/non-paralysed sides). Although the proposed **PNP** features can alleviate this problem to some extent, it is beneficial to combine the less correlated features (i.e.,**PNP**, **SAD**, and *ini*) and, due to the feature redundancy, it is crucial to extract a compact representation for the prediction model development. Note that the *ini* is the first week's CAHAI scores which were used as medical history information.

3.3.2 Evaluation of the Predictive Models

Feature Selection

Based on the feature correlation analysis in Section 3.3.1, it is important that we select the most relevant features from various sources (i.e., **PNP**, **SAD**, and *ini*). Different from the correlation-based approach which can select each feature independently (by the correlation coefficient), LASSO can select the feature by solving a linear optimisation problem with a sparsity constraint, and it takes the relationship of the features into consideration. Various hyperparameters of the LASSO regression methods were applied in the R package, "lars" in this chapter. To be more specific, it provides the entire path, each path contains the different tuning parameter λ. Based on the criteria of minimum Mallows C_p, we find the "best path" and find out what the tuning parameter λ is. Based on LASSO we selected the most important features for both acute/chronic patients, as shown in Table 3.7.

Acute Patients	Chronic Patients
PNP_3^2, PNP_6^1, SAD_2^{np}, $SAD_{1.2}^{p}$ SAD_6^{np}, ini	$PNP_{1.4}^1$, SAD_4^p, SAD_2^{np}, $PNP_{1.3}^2$ PNP_4^1, $PNP_{1.1}^2$, ini, PNP_6^1 $SAD_{1.4}^{np}$, SAD_6^{np}

Table 3.7: Selected features using LASSO.

Wavelet-based features have a clear physical explanation. SAD_j represents the point energy in the raw signal at the decomposition level j based on the energy preserving condition (see **Section** 3.2.1 for more details). Specifically, it relates to the degree of energy among the different activity levels (in different frequency domains based on the decomposition scale j). The activities such as jumping or lifting an object may correspond to high-frequency signal, while sedentary or eating may be low-frequency signal. Based on these, we can interpret the key features in Table 3.7. For example, for acute patients key features (which is highly-related to stroke-rehab modelling) correspond to asymmetric

activities in low/medium-frequency level (i.e., with PNP_3^2, PNP_6^1), non-paralysed-based activities in low/medium-frequency level (i.e., with SAD_2^{np}, SAD_6^{np}), and paralysed-side based activities in high-frequency level (i.e.,with $SAD_{1.2}^{p}$).

Performance of linear fixed-effects model

Based on the selected features, we perform further leave-one-patient-out cross validation on these two patient groups respectively using the linear fixed-effects model. The prediction results of the chronic patient group tend to be much better than the ones of the acute group irrespective of the feature selection approach, as shown in **Figure**. 3.12. One main reason is due to the nature of the patient group. In **Figure** 3.13, we illustrated the clinical CAHAI distribution (i.e., the ground truth CAHAI) from week 2 to week 8, and we can see that the clinical CAHAI scores are very stable for chronic patients. On the other hand, for acute patients who suffered from stroke in the past 6 months, their health statuses are less stable and affected significantly by various factors, and in this case the simple linear fixed-effected model yields less promising results.

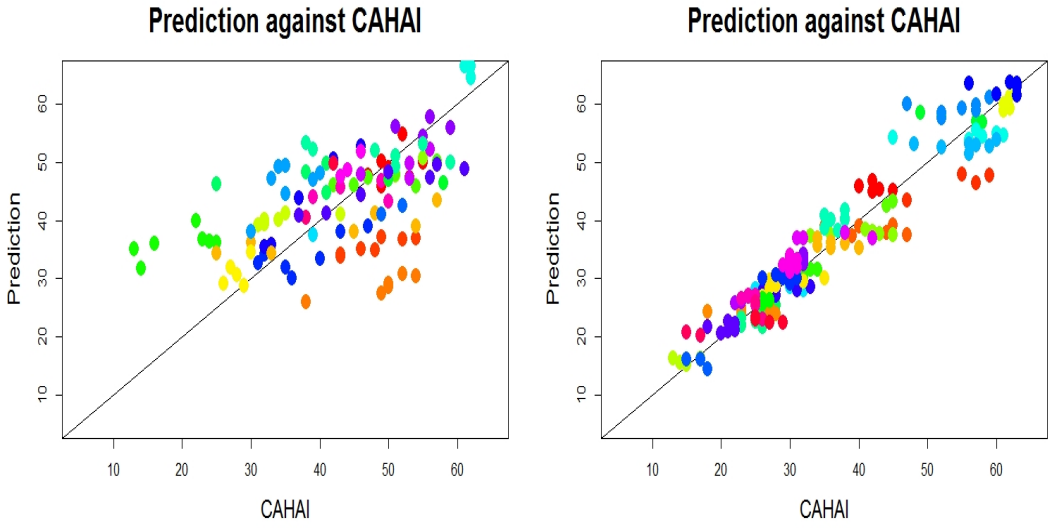


Figure 3.12: Linear model prediction vs clinical CAHAI by using leave-one-patient-out cross-validation; Left: Acute patients (RMSE 7.24); Right: Chronic patients (RMSE 3.29). Note that different colours represent different patients.

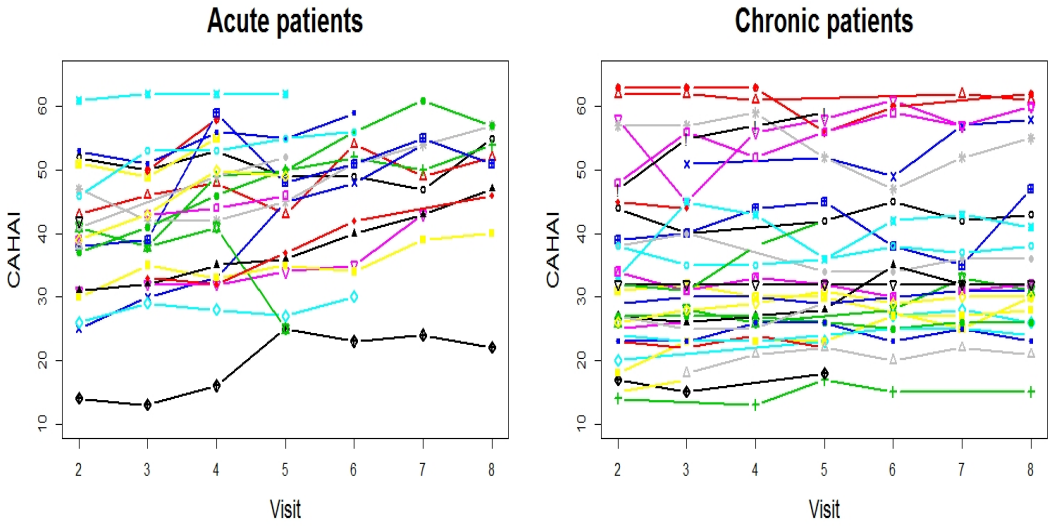


Figure 3.13: Clinical assessed CAHAI distribution with respect to visit; Stroke rehabilitation levels may be stable for chronic patients while may vary substantially for acute patients. Note that different colours represent different patients.

Performance of Longitudinal mixed-effects Model with Gaussian Process prior

Based on the selected features in **Table** 3.7, we also propose to use a Longitudinal mixed-effects Model with Gaussian Process prior (LMGP) for both patient groups. We applied different covariance kernels in LMGP models and found that the one with powered exponential kernel achieves the best results. The following discussion will therefore focus on the model which uses this kernel. Three kernels were used in LMGP, and they are linear kernel, powered exponential kernel and rational quadratic kernel. We used the selected features (from Table 3.7) as the fixed-effects features and random-effects features, and the results were reported in **Table** 3.15, **Appendix** 3.5.

Here, we set features from the fixed-effects part and random-effects part the same, such that $\boldsymbol{x}_{i,j} = \boldsymbol{\phi}_{i,j}$ in **Equation** ((3.4)). Similar to the linear fixed-effects model, we evaluated the performance based on leave-one-patient-out cross validation, and the mean RMSE values were reported in **Table** 3.8, as also shown in **Figure** 3.14.

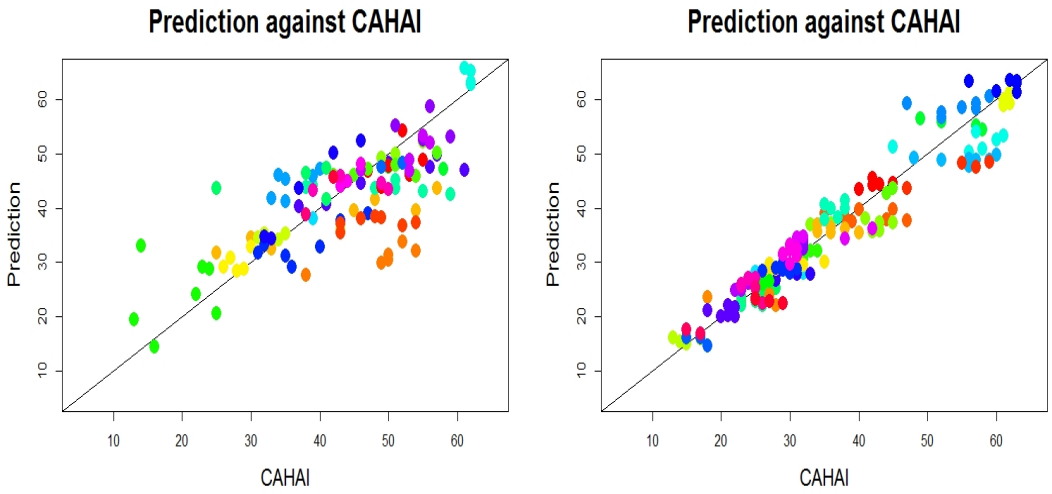


Figure 3.14: Prediction vs clinical assessed CAHAI by using leave-one-patient-out cross-validation. Left panel: RMSE is 5.75 in acute patients. Right panel: RMSE is 3.12 in chronic patients. Note that different colours represent different patients.

-	RMSE	**Selected features** ($\boldsymbol{x}_{i,j}$)	**Selected features** ($\boldsymbol{\phi}_{i,j}$)
Acute	5.75	6 features selected by LASSO	6 features selected by LASSO
Chronic	3.12	10 features selected by LASSO	10 features selected by LASSO

Table 3.8: The leave-one-patient-out corss validation using features selected by LASSO in longitudinal mixed-effects with GP prior model. Note that: the 6 and 10 selected features for acute and chronic patients by LASSO can be found in **Table** 3.7.

Figure 3.14 shows the results of leave-one-patient-out prediction against the clinical assessed CAHAI scores for both acute and chronic patients. Comparing with the results using leave one-patient-out cross validation in the fixed-effects model, **Figure** 3.12 shows poor performance for the patients with low CAHAI scores in acute patients; while the use of the nonlinear random-effects with GP prior improves the results considerably, particularly for the patients with low scores. The overall RMSE of 5.75 (7.24 for the fixed-effects model) confirms this. For comparison, we also applied the nonlinear mixed-effects model to the chronic patients' data. The difference between the mixed-effects model and the fixed-effects model is negligible as expected. Based on LMGP, we also performed "continuous monitoring" on 4 patients (two for each patient group) from week 2 to week 8 by predicting the week-wise CAHAI scores (with mean and 95% confidence interval) in **Figure** 3.15, which is extremely helpful when uncertainty measurement is required.

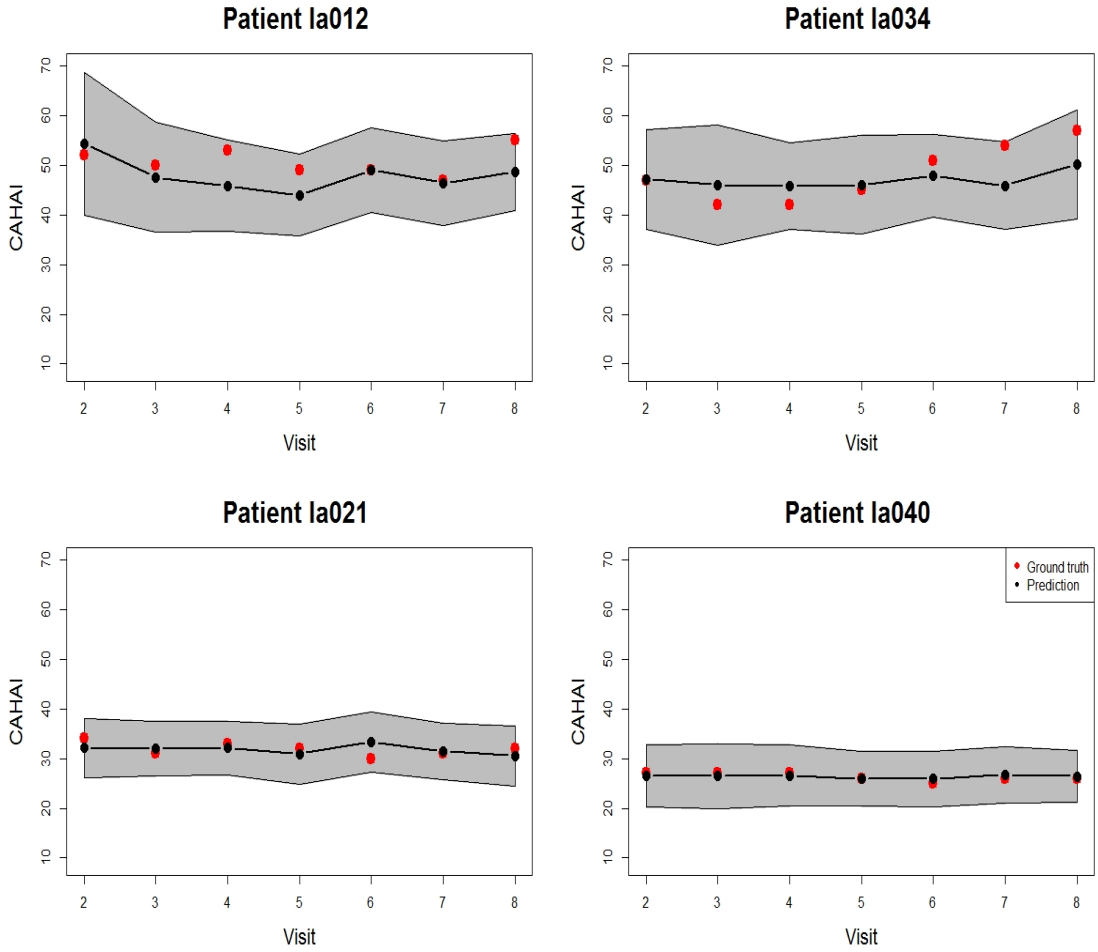


Figure 3.15: The plots of prediction for two acute patients (Top and two chronic patients (Bottom) by using longitudinal mixed-effects model with GP prior.

On the fixed-effects part of LMGP

LMGP includes two key parts, i.e., the linear fixed-effects and the non-linear random-effects part, and it is important to choose the key features for modelling. Since the fixed-effects part measures the main (linear) relationship between the input features and the predicted CAHAI score, we studied the corresponding feature subsets. For the random-effects part, we used the full LASSO features (as shown in **Table** 3.7).

To select the most important feature subset for the fixed-effects part modelling, we ranked the features (from Table 3.7) based on two criteria: LASSO coefficients, and correlation coefficients (between features and CAHAI, as described in **Section** 3.3.1). Table 3.9 demonstrates ranked features, and here only the top 50% of features were used (i.e., top 3 features for acute patients and top 5 features for chronic patients). The fixed-effects part and the settings, together with the results, were reported in Table 3.10.

Feature Ranking Criterion	Acute Patients	Chronic Patients
LASSO Coefficients (absolute value)	PNP_3^2, PNP_6^1, SAD_2^{np}, $SAD_{1.2}^p$ SAD_6^{np}, ini	$PNP_{1.4}^1$, SAD_4^p, SAD_2^{np}, $PNP_{1.3}^2$ PNP_4^1, $PNP_{1.1}^2$, ini, PNP_6^1 $SAD_{1.4}^{np}$, SAD_6^{np}
Correlation Coefficients (absolute value)	PNP_3^2, ini, $SAD_{1.2}^p$, PNP_6^1 SAD_2^{np}, SAD_6^{np}	ini, $PNP_{1.4}^1$, $PNP_{1.3}^2$, $PNP_{1.1}^2$ $SAD_{1.4}^{np}$, SAD_2^{np}, PNP_4^1, SAD_6^{np} PNP_6^1, SAD_4^p

Table 3.9: Feature importance ranking for acute/chronic patients.

Acute Patients	**Selected features** ($\boldsymbol{x}_{i,j}$)	**Selected features** ($\boldsymbol{\phi}_{i,j}$)
RMSE: 5.75	full 6 features in Table 3.7	full 6 features in Table 3.7
RMSE: 5.37	top 3 features (Corr criterion in Table 3.9): PNP_3^2, ini, $SAD_{1.2}^p$	full 6 features in Table 3.7
RMSE: 5.51	top 3 features (LASSO criterion in Table 3.9): PNP_3^2, PNP_6^1, SAD_2^{np}	full 6 features in Table 3.7
Chronic Patients	**Selected features** ($\boldsymbol{x}_{i,j}$)	**Selected features** ($\boldsymbol{\phi}_{i,j}$)
RMSE: 3.12	full 10 features in Table 3.7	full 10 features in Table 3.7
RMSE: 3.20	top 5 features (Corr criterion in Table 3.9): ini, $PNP_{1.4}^1$, $PNP_{1.3}^2$ $PNP_{1.1}^2$, $SAD_{1.4}^{np}$	full 10 features in Table 3.7
RMSE: 5.12	top 5 features (LASSO criterion in Table 3.9): $PNP_{1.4}^1$, SAD_4^p, SAD_2^{np} $PNP_{1.3}^2$, PNP_4^1	full 10 features in Table 3.7

Table 3.10: LMGP's fixed-effects part modelling results (RMSE) based on different feature subsets

It is interesting to observe that the performance may change substantially depending on different settings. Specifically, with the top feature subsets, modelling the LMGP's fixed-effects part can further reduce the errors for acute patients, in contrast to chronic patients with increased errors. The top 5 features selected via the LASSO criterion yields the worst performance for chronic patients, and one possible explanation could be that it missed the *ini* feature, which reflects the initial health condition (see **Figure** 3.13), and this may play a major role in chronic patient modelling.

Model comparison

Models	RMSE (Acute)	RMSE (Chronic)
PCA + GMM + LMGP Tang et al. (2020)	15.98	12.76
stroke-rehab-driven + DNN (3-layer MLP)	10.50	4.93
stroke-rehab-driven + SVR (linear)	7.47	3.25
stroke-rehab-driven + SVR (rbf)	9.67	4.92
stroke-rehab-driven + RF	8.19	3.93
stroke-rehab-driven + LM	7.24	3.29
stroke-rehab-driven + LMGP	5.75	3.12

Table 3.11: Model comparison, note that: DNN (3-layer MLP) stands for deep neural network regression by using 3-layer multi-layer perceptron; SVR (linear) and SVR (rbf) stand for support vector regression by using linear kernel and radial basis respectively; RF stands for the random forest regression; LM and LMGP stand for linear fixed-effects model and Longitudinal mixed-effects Model with Gaussian Process prior respectively.

For model comparison, we implemented the state-of-the-art method, found in Tang et al. (2020), which extracted features from the raw signal. Specifically, PCs score were extracted from sliding windows, before a Gaussian Mixture Model was applied for high-level representation learning (for each trial). The learned trial-wise features were fed to LMGP for CAHAI score prediction. Following our settings, we used LOPO-CV and reported the mean RMSE values (in Table 3.11) for acute group and chronic group, respectively.

Based on our proposed (41-dimensional) stroke-rehab-driven features, we also compared LMGP with a number of other ML models, such as deep neural network (DNN), support vector regression (SVR) and random forest regression(RF) for acute/chronic patient groups. It is worth noting that we cannot use the popular deep learning structures such as convolutional neural network(CNN) or recurrent neural network(RNN), due to the lack of frame-wise or sample-wise annotation. Yet with the stroke-rehab-driven features and trial-wise annotation, simple multi-layer perceptron(MLP) can be applied, and here we implemented our DNN using a 3-layer MLP for the CAHAI score regression tasks. Similarly, LOPO-CV was applied and the mean RMSE values were reported in Table 3.11.

From Table 3.11, we can see generally that systems based on the proposed stroke-rehab-driven features have a much lower RMSE than the state-of-the-art data-driven approach (Tang et al., 2020), which indicates the necessity of (domain-knowledge driven) feature engineering when adequate annotations are not available. Due to inadequate labels, unsupervised clustering (i.e., GMM (Tang et al., 2020)) was performed for trial-wise representation learning, which is computationally expensive. Due to the computational cost, only 1% of the training data was used (Tang et al., 2020), yielding unsatisfactory results in the LOPO-CV settings.

Based on the proposed stroke-rehab-driven features, we observed that linear models yielded reasonable results (linear SVR and linear fixed-effects model), while the non-linear baselines (DNN, SVR(rbf), and RF) had the worst performance. One explanation is that the over-fitting effect, where the trained non-linear models do not generalise well to the unseen patients/environments in this longitudinal study setting. RF's performance may also suffer significantly from the low-dimensionality of the selected features (6 features for acute patients and 10 features for chronic patients). Given the simplicity of the linear models and the designed low-dimensional features, linear models tend to suffer less from the over-fitting effect, giving reasonable results in these challenging environments. Compared with the baselines, our LMGP can further model the longitudinal mixed-effects (i.e., with linear fixed-effect part and non-linear random-effects part), making the system adaptive to different subjects/time-slots, with the lowest errors.

3.4 Conclusion

This chapter mainly evaluated and predicted the recovery level after stroke based on the accelerometer data from wrist-worn sensor AX3. To map the long time-series (i.e., 3-day accelerometer data) to the CAHAI score, we proposed a pipeline which performed data cleaning, feature design, to predictive model development. Specifically, we proposed two compact features which can well capture the rehabilitation characteristics while suppressing the irrelevant daily activities, which is crucial in analysing the data collected in free-living environments. We further employed LMGP, which can make the model adaptive to different subjects and different time slots (across different weeks). Comprehensive experiments were conducted on both acute/chronic patients, and very promising results were achieved, especially on the chronic patient group. We also studied different feature subsets on modelling the fixed-effects part in LMGP, and experiments suggested the errors can be further reduced for the challenging acute patient population.

Due to irrelevant daily activities and strong heterogeneity among subjects, it is very challenging for researchers in statistics, computing sciences and other areas to deal with free-living data. It is also crucial to develop models which have good mathematical properties and have physical explanation particularly in medical research. We hope that the ideas of the new features and the models discussed in this chapter can provide some hints on addressing similar problems in health research.

For further analysis in the accelerometer data from wrist-worn sensor AX3, we need to analyze the **DWT** coefficients directly rather than using the scalar features only. We will start the research by investigating **DWT** tree structure and apply it to the data reduction through the Bayesian wavelet regression in the next chapter.

3.5 Appendix

SAD_j against CAHAI score in acute patients

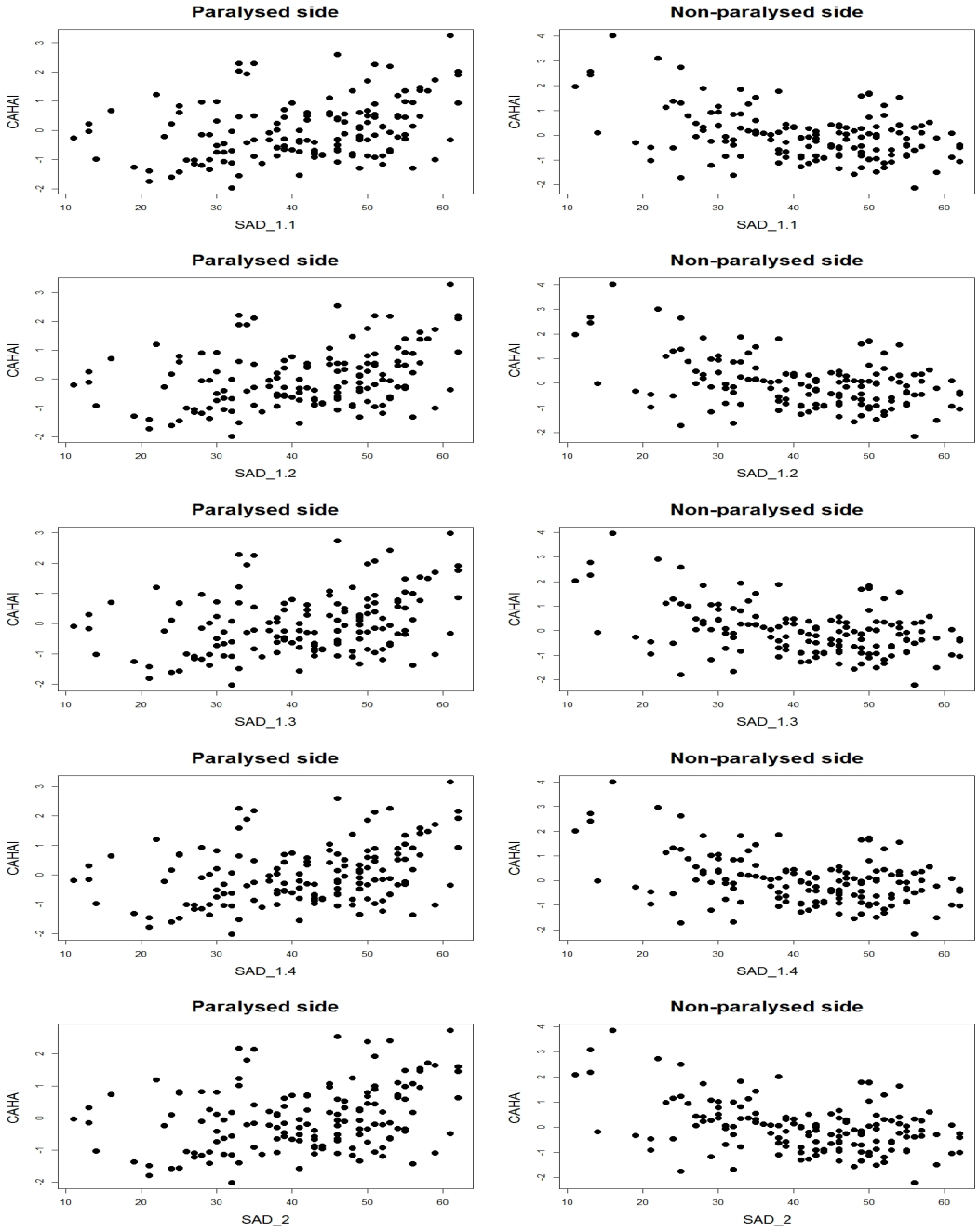


Figure 3.16: The SAD_j (acute patients) from scale 1.1 to scale 2 against CAHAI in both sides. Left panel: paralysed side. Right panel: non-paralysed side.

SAD_j against CAHAI score in acute patients

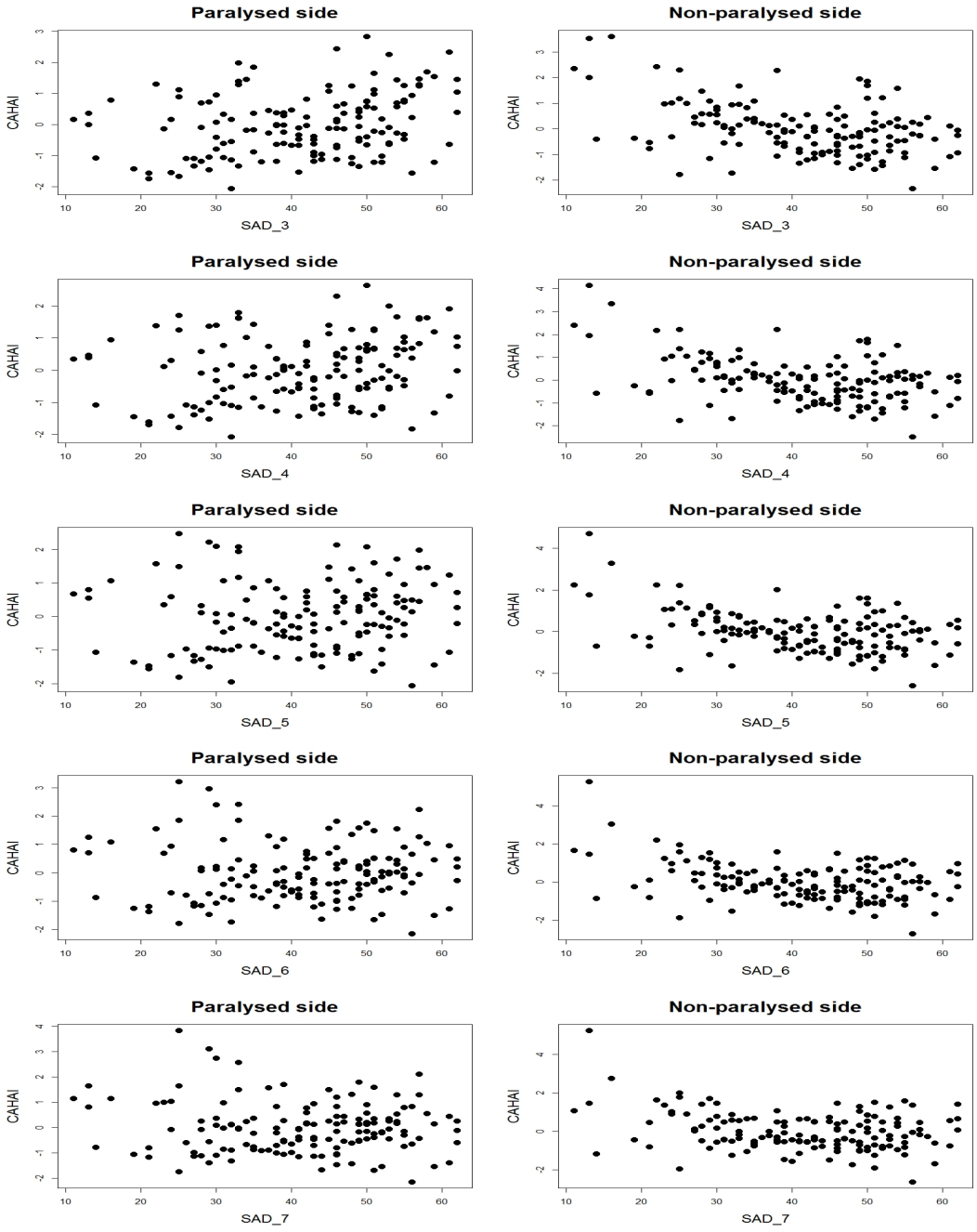


Figure 3.17: The SAD_j (acute patients) from scale 3 to scale 7 against CAHAI in both sides. Left panel: paralysed side. Right panel: non-paralysed side.

SAD_j against CAHAI score in chronic patients

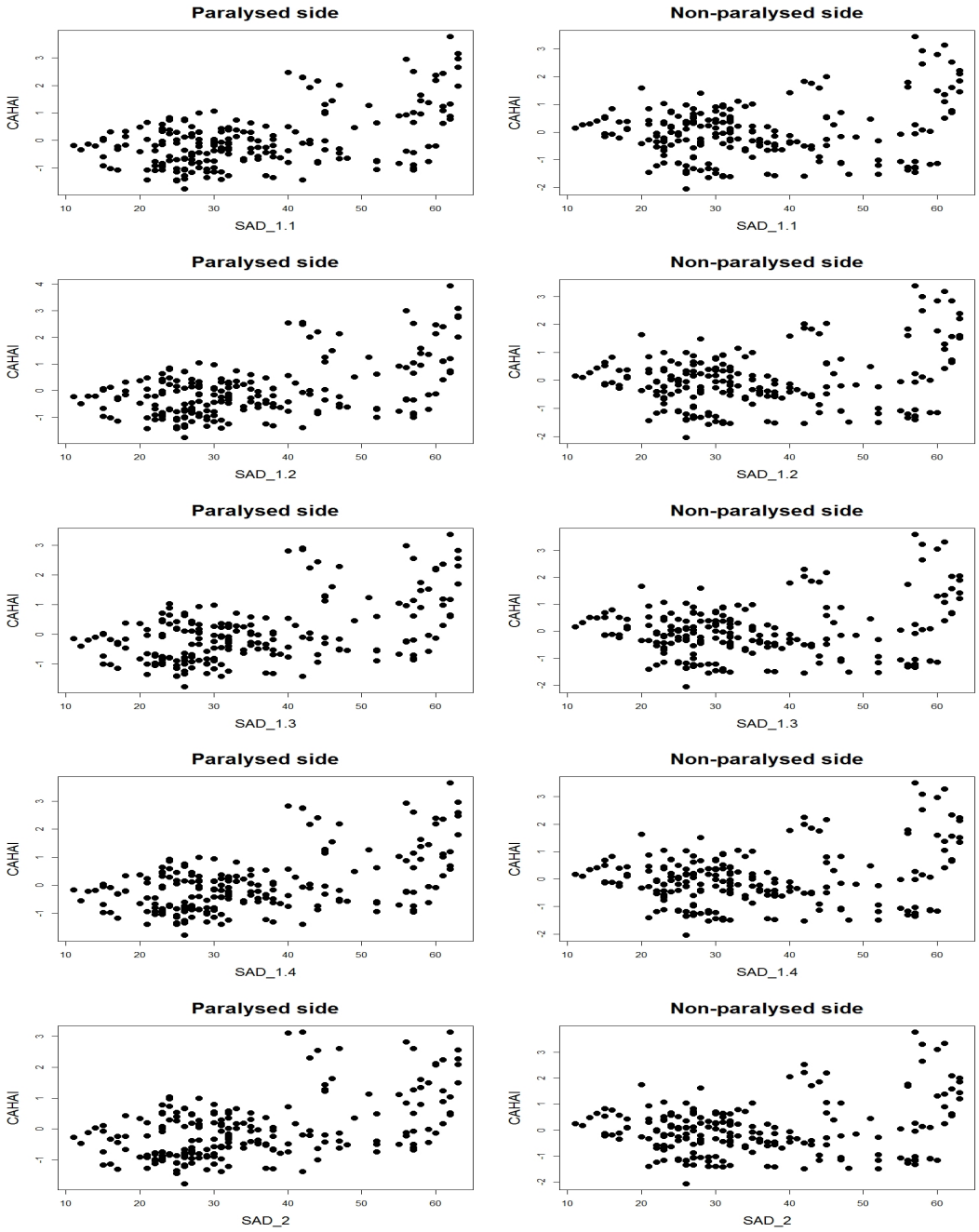


Figure 3.18: The SAD_j (chronic patients) from scale 1.1 to scale 2 against CAHAI in both sides. Left panel: paralysed side. Right panel: non-paralysed side.

SAD_j against CAHAI score in chronic patients

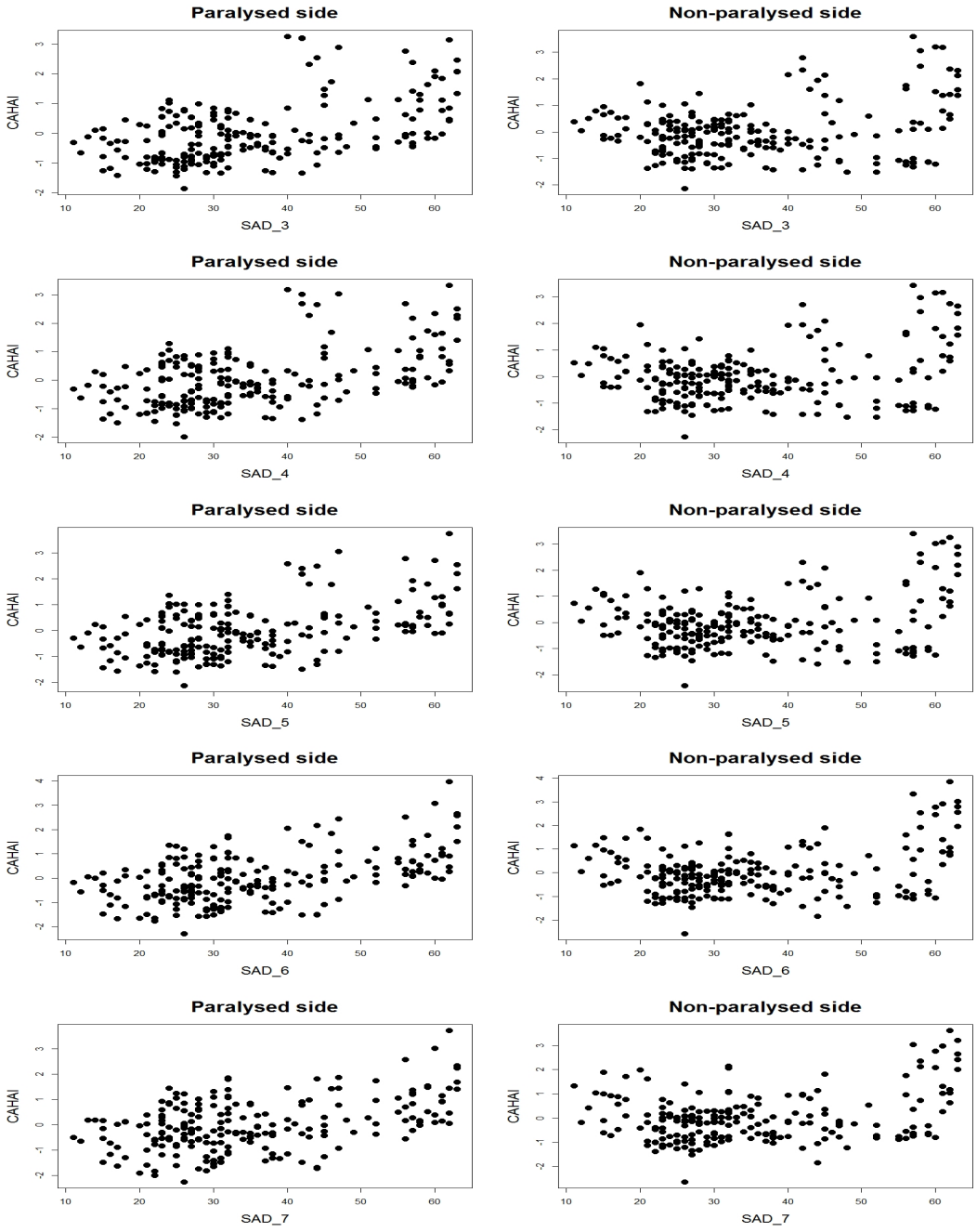


Figure 3.19: The SAD_j (chronic patients) from scale 3 to scale 7 against CAHAI in both sides. Left panel: paralysed side. Right panel: non-paralysed side.

$\mathbf{PNP_k^1}$ against CAHAI score in acute patients

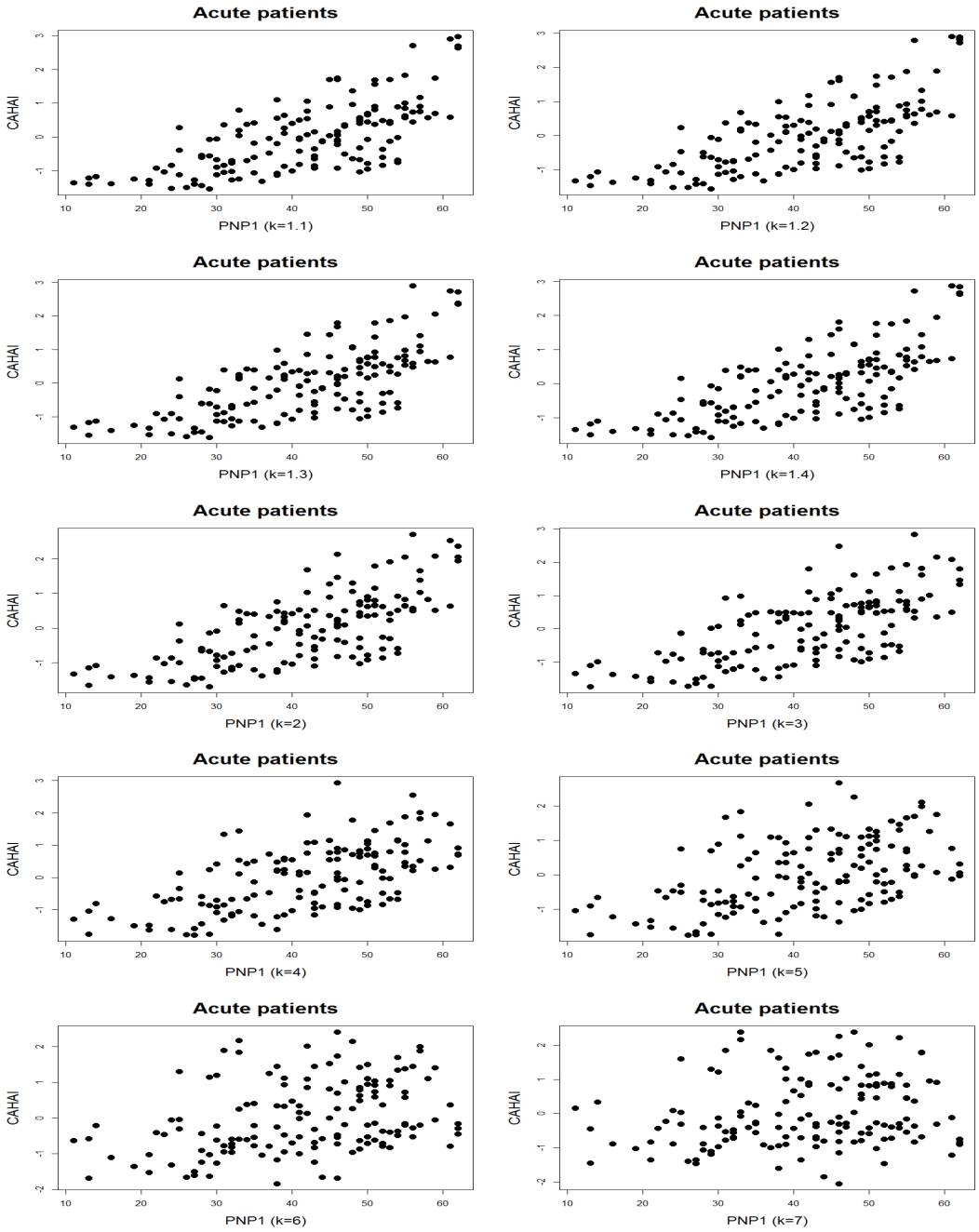


Figure 3.20: The PNP_k^1 from scale 1.1 to scale 7 against CAHAI in acute patients.

$\mathbf{PNP_k^2}$ against CAHAI score in acute patients

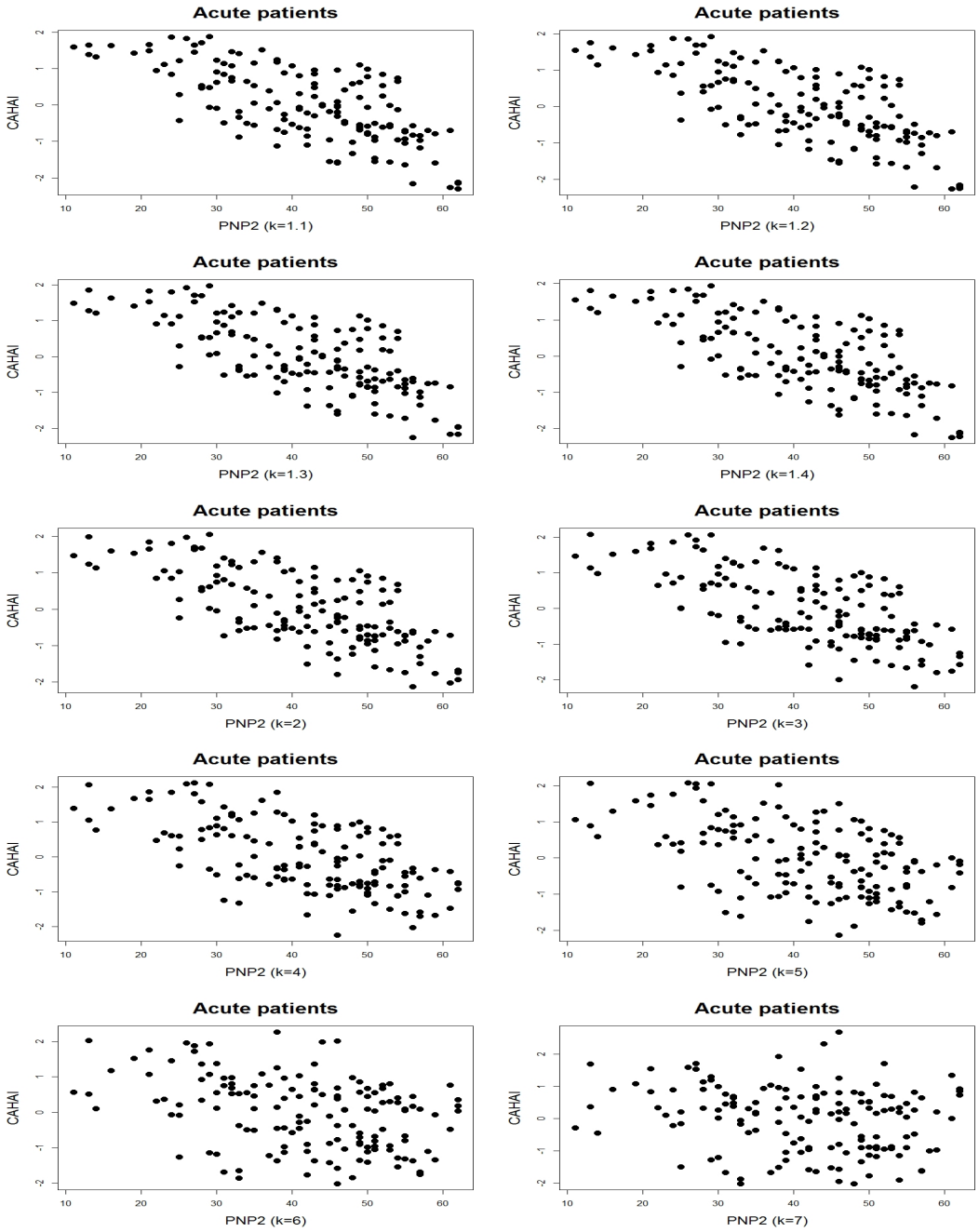


Figure 3.21: The PNP_k^2 from scale 1.1 to scale 7 against CAHAI in acute patients.

$\mathbf{PNP_k^1}$ against CAHAI score in chronic patients

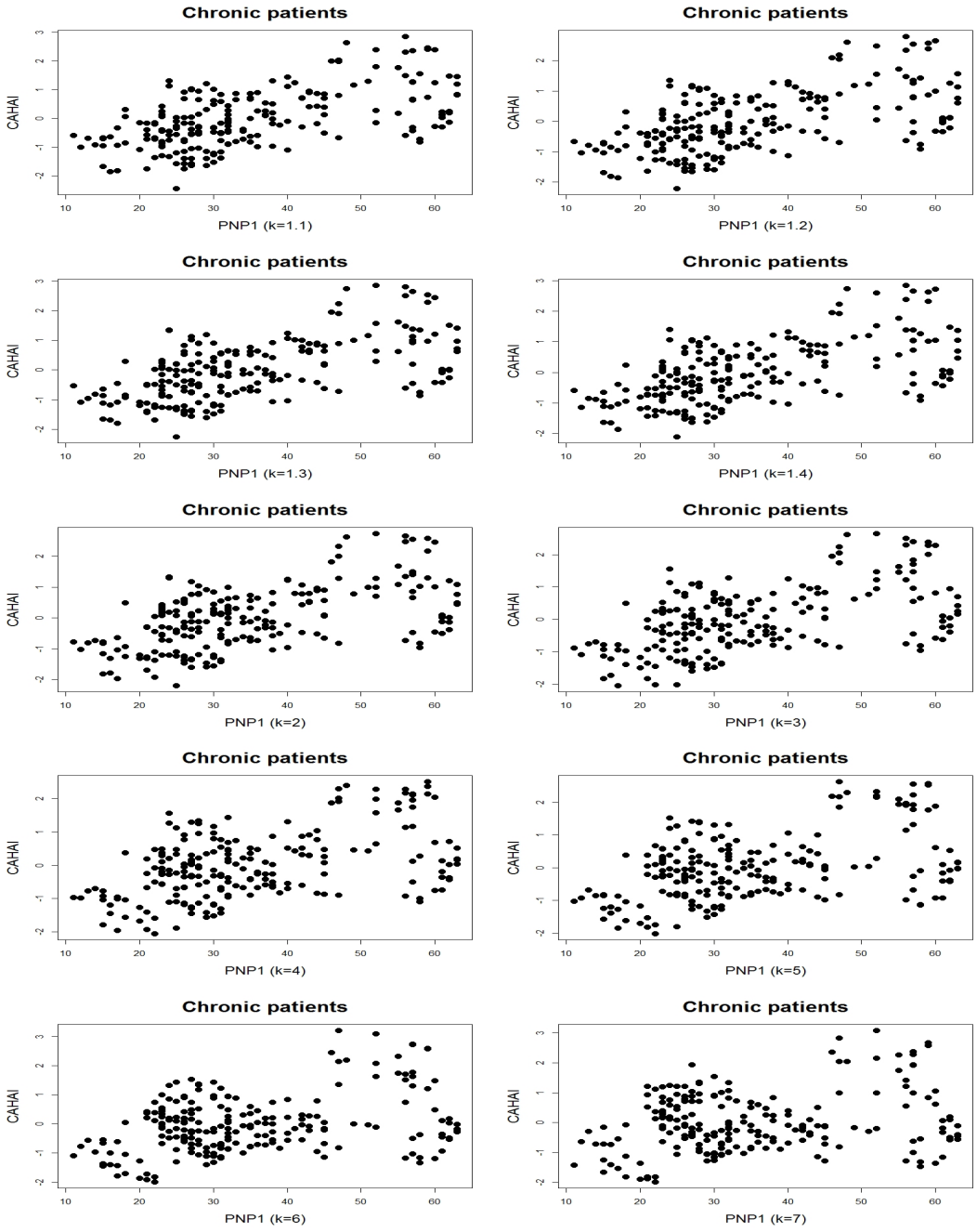


Figure 3.22: The PNP_k^1 from scale 1.1 to scale 7 against CAHAI in chronic patients.

$\mathbf{PNP_k^2}$ against CAHAI score in chronic patients

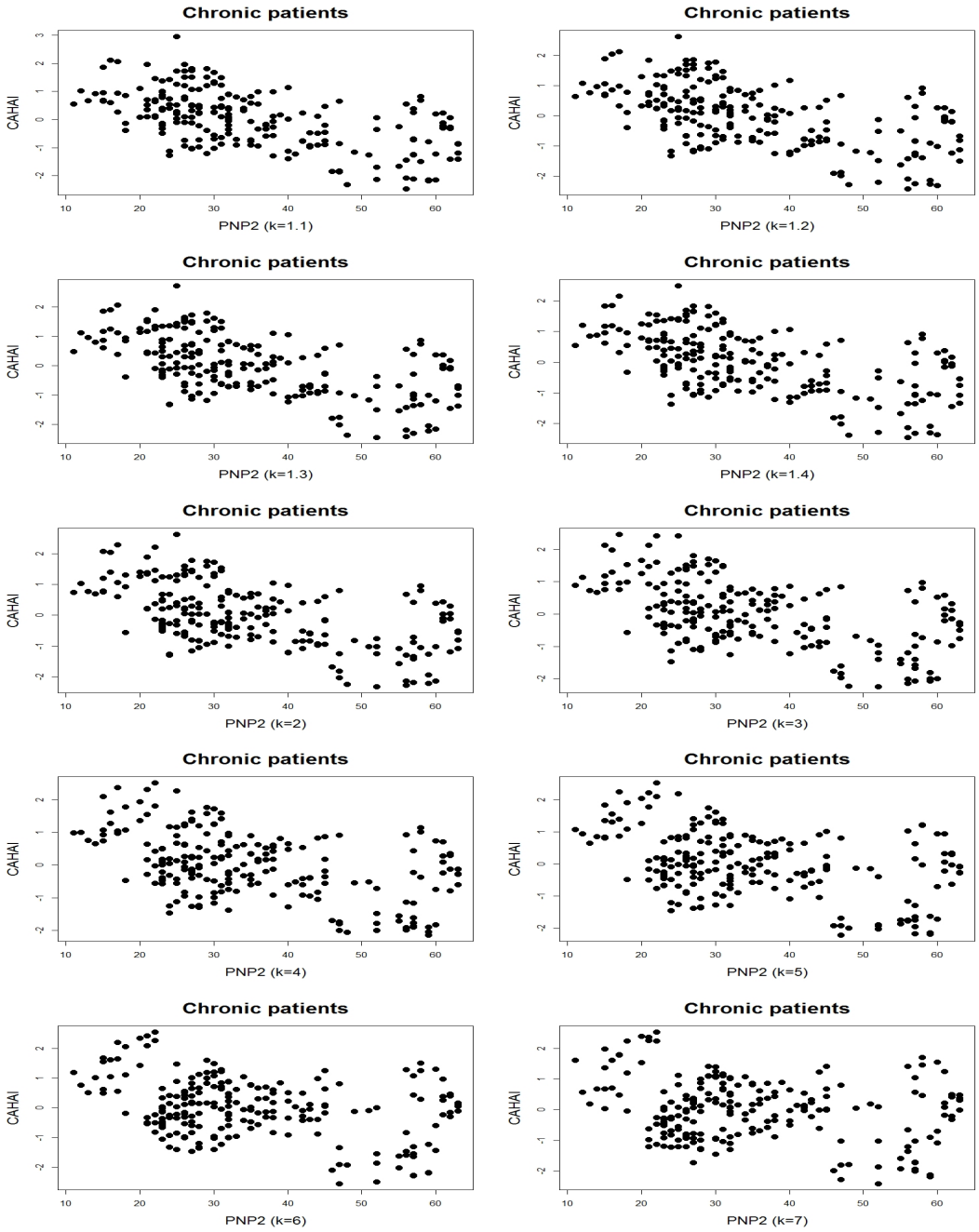


Figure 3.23: The PNP_k^2 from scale 1.1 to scale 7 against CAHAI in chronic patients.

	CAHAI	$SAD_{1.1}$	$SAD_{1.2}$	$SAD_{1.3}$	$SAD_{1.4}$	SAD_2
la027	57	0.0087442	0.0096013	0.0106240	0.0102000	0.0121560
la038	22	0.0082301	0.0086263	0.0097689	0.0092717	0.0111490
	CAHAI	SAD_3	SAD_4	SAD_5	SAD_6	SAD_7
la027	57	0.017742	0.025349	0.039887	0.061012	0.097131
la038	22	0.017008	0.026047	0.041279	0.063810	0.088992

Table 3.12: The SAD_j between Patient la012 and Patient la038 in 8-th visit (paralysed side).

	CAHAI	$SAD_{1.1}$	$SAD_{1.2}$	$SAD_{1.3}$	$SAD_{1.4}$	SAD_2
la027 (P-side)	57	0.0087442	0.0096013	0.0106240	0.0102000	0.0121560
la027 (NP-side)	57	0.0098804	0.0103430	0.0113060	0.0107470	0.0123570
la038 (P-side)	22	0.0082301	0.0086263	0.0097689	0.0092717	0.0111490
la038 (NP-side)	22	0.0202720	0.0208610	0.0230570	0.0220820	0.0252900
	CAHAI	SAD_3	SAD_4	SAD_5	SAD_6	SAD_7
la027 (P-side)	57	0.017742	0.025349	0.039887	0.061012	0.097131
la027 (NP-side)	57	0.017344	0.024992	0.034701	0.051006	0.077759
la038 (P-side)	22	0.017008	0.026047	0.041279	0.063810	0.088992
la038 (NP-side)	22	0.034370	0.046431	0.065878	0.092356	0.119280

Table 3.13: SAD_j of Patient la012 and Patient la038 in the 8-th visit. Note that P-side means paralysed side and NP-side means non-paralysed side.

	CAHAI	$PNP^1_{1.1}$	$PNP^1_{1.2}$	$PNP^1_{1.3}$	$PNP^1_{1.4}$	PNP^1_2
la027	57	0.8850047	0.9282897	0.9396780	0.9491021	0.9837339
la038	22	0.4059836	0.4135133	0.4236848	0.4198759	0.4408462
	CAHAI	PNP^1_3	PNP^1_4	PNP^1_5	PNP^1_6	PNP^1_7
la027	57	1.0229474	1.0904289	1.1551252	1.1735678	1.2414130
la038	22	0.4948502	0.5609830	0.6265977	0.6909134	0.7460765
	CAHAI	$PNP^2_{1.1}$	$PNP^2_{1.2}$	$PNP^2_{1.3}$	$PNP^2_{1.4}$	PNP^2_2
la027	57	0.0610053	0.0371886	0.0310990	0.0261135	0.0081997
la038	22	0.4224917	0.4149142	0.4048054	0.4085738	0.3880732
	CAHAI	PNP^2_3	PNP^2_4	PNP^2_5	PNP^2_6	PNP^2_7
la027	57	-0.0113436	-0.043259	-0.071980	-0.079854	-0.1077056
la038	22	0.33792670	0.2812440	0.2295604	0.1827927	0.14542521

Table 3.14: New features PNP^1_k and PNP^2_k between Patient la027 and Patient la038 in 8-th visit.

Selected kernels in LMGP	RMSE (Acute)	RMSE (Chronic)
linear kernel	5.89	3.13
powered exponential kernel	5.75	3.12
rational quadratic kernel	7.58	3.24

Table 3.15: Kernels comparison through LMGP

LASSO

LASSO is a regression method that involves penalizing the absolute size of the regression coefficients. It may, however, give rise to a situation where some of the parameter estimates may be exactly zero. The larger the penalty applies, the further estimates are shrunk towards zero. When

$$\sum_{j=1}^{p} |\beta_j| \leq s,$$

then

$$\hat{\boldsymbol{\beta}}^{lasso} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \,.$$

where λ is the regularisation coefficient. Moreover, the threshold s is one-to-one parallel with the constrain $\|\boldsymbol{\beta}\|_1 \leq s$. It is a "path" from the solutions indexed by s. It may, however, based on the λ and s, give rise to a situation where some of the parameter estimates may be exactly zero. The larger the penalty applies, the further estimates are shrunk towards zero. More details of LASSO can be found in Tibshirani (1996). In this chapter, we used the R package, **lars**, for features selection using Mallows C_p. To be more specific, **lars** produces the entire path, with each path corresponding to a series of coefficients and a value of λ. Then we choose the path which has the minimum Mallows C_p. Afterwards, the coefficients and λ will be given automatically, and the LASSO variable selection will be finished based on the minimum Mallows C_p.

Chapter 4

Data reduction using Bayesian Normal Inverse-Gamma Markov tree

Since **DWT** coefficients are another form of the original process data in different frequency domains, we will apply **DWT** coefficients directly instead of scalar features in this chapter. Our first task is to remove the "noise" from the **DWT** coefficients in different scales and this must be done using an apropriate method. This part of **DWT** coefficients includes no or little useful information except "noise". In addition, when patients wear the wrist-worn sensor AX3 for 3 full days, there are numerous "still" periods (e.g., sleeping, sitting), and data collected in those periods also provides little information. So our second task is to look for an appropriate way to shrink down the "'still" periods, so that the **DWT** coefficients with little information will be removed but the others will remain. In order to get rid of the "noise" and the "still" part, the key is to find a proper threshold which depends on the scale and the structure of the **DWT** coefficients. In this chapter, an efficient way, through wavelet regression with Normal Inverse-Gamma Markov tree **(NIG-MT)** model (Ma & Soriano, 2017), is to be used to remove the "noise" and "still" parts automatically. After applying **(NIG-MT)** to our data, more than three quarters of the **DWT** coefficients can be removed with losing negligible information.

Moreover, as we discussed in **Section** 1.1, **Chapter** 1 that the movements are usually described by accelerometer data only in the time-domain, it is better to consider also the frequency domain. After applying the **DWT**, the acclerometer data will transfer to the **DWT** coefficients. Hence, instead of giving an analysis to the accelerometer data directly, an analysis of the **DWT** coefficients will provide more valued information since it contains the information from both time and frequency-domains. Then, a **DWT** tree structure with a certain pattern can be found within the **DWT** coefficients, based on the special property

of the **DWT** tree structure (see **Section** 4.1.1). The usual data reduction is hard to apply it directly; the **NIG-MT** will provide a special data reduction method (especially the free-living data for the stroke patients) based on the **DWT** tree structure which contains the information from both time and frequency-domains.

4.1 Essence of DWT coefficients

Instead of applying both **DWT** and **DWPT** in the original process data, we only focus on **DWT** and obtain 6 decomposition levels on patients' data. After applying **DWT** in **Section** 2.3.3 with patients' **VM** data (the signal vector magnitude of accelerometer data; see **Section** 2.2.3), **DWT** coefficients $\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_6$ are obtained at 6 scales. Supposing the data has length N. The length of **DWT** coefficients in each scale and their frequency domain will be presented in **Table** 4.1. From this chapter, in order to describe the tree structure of detailed coefficients through the **DWT** easily, we use the notation **DWT** coefficients as detailed coefficients. Notations of decomposition scale 1, scale 2, scale 3, scale 4, scale 5 and scale 6 are replaced as level 5, level 4, level 3, level 2, level 1 and level 0 respectively in this chapter.

	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$
Scale	Scale 1 (Level 5)	Scale 2 (Level 4)	Scale 3 (Level 3)
Frequency	0.50Hz - 1Hz	0.25Hz - 0.50Hz	0.125Hz - 0.25Hz
Length	$\frac{N}{2}$	$\frac{N}{4}$	$\frac{N}{8}$
	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$
Scale	Scale 4 (Level 2)	Scale 5 (Level 1)	Scale 6 (Level 0)
Frequency	0.0625Hz - 0.125Hz	0.0312Hz - 0.0625Hz	0.0156Hz - 0.0312Hz
Length	$\frac{N}{16}$	$\frac{N}{32}$	$\frac{N}{64}$

Table 4.1: The length and frequency domain of **DWT** coefficients in each scale.

4.1.1 Tree structures of DWT coefficients

From **Table** 4.1, a very interesting characteristic can be found: **DWT** coefficients at higher levels are twice the number of those at lower levels (e.g., the number of $\mathbf{W}_6$ with length $\frac{N}{64}$ at level 0 and the number of $\mathbf{W}_5$ with length $\frac{N}{32}$ at level 1). What is the inner relationship with the length of **DWT** coefficients between two adjacent levels? The answer is to treat **DWT** coefficients as the tree structure (see **Figure** 4.1): every node at previous level corresponds to two nodes at the next level. To be more specific, a parent node at level j has two children nodes at level $j+1$. Here is a simple example by using the bumps function (which is the test function from Donoho & Johnstone (1994)) and its **DWT** coefficients in **Figure** 4.2. From **DWT** coefficients in the bumps test function,

there is another interesting phenomenon can be found in that if the parent node at level j is far away from 0, then the two children nodes tend to be far away from 0 as well at the next level $j+1$ (Ma & Soriano, 2017). This is an important property among **DWT** coefficients in the tree structure and this is the foundation of **NIG-MT**.

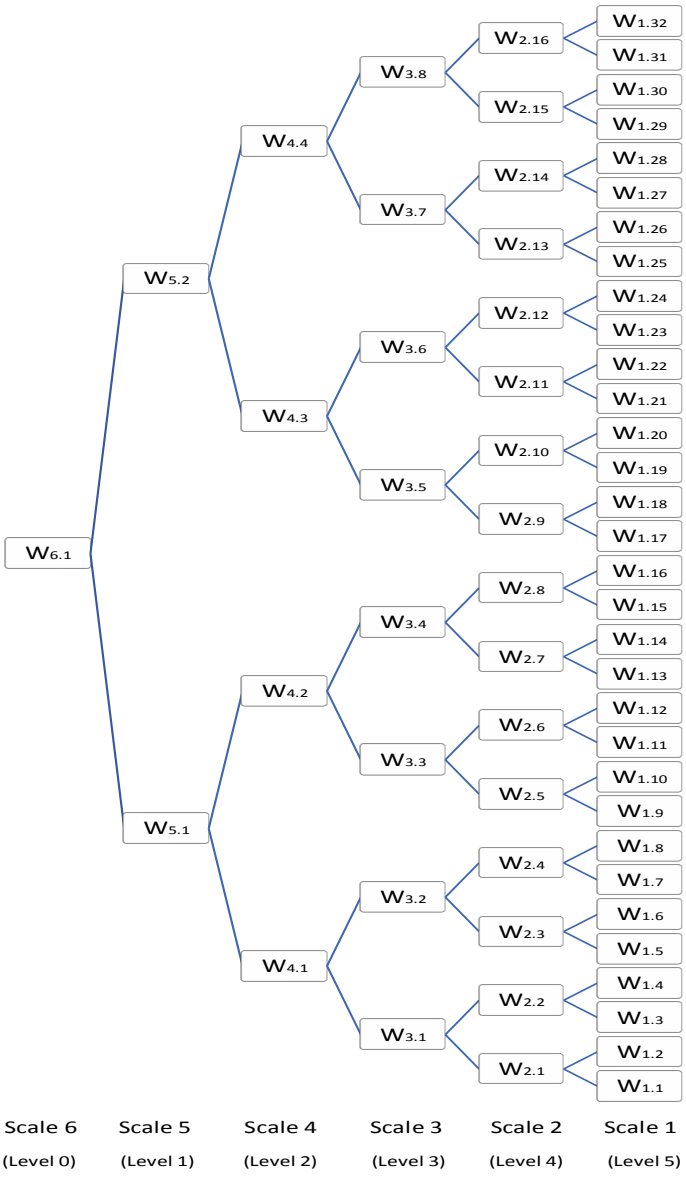


Figure 4.1: A small **DWT** coefficients tree structure when **DWT** decomposition level $j = 6$. The node with index by (j, k), where j is the **DWT** decomposition level and k is the location of the **DWT** coefficients at decomposition level j.

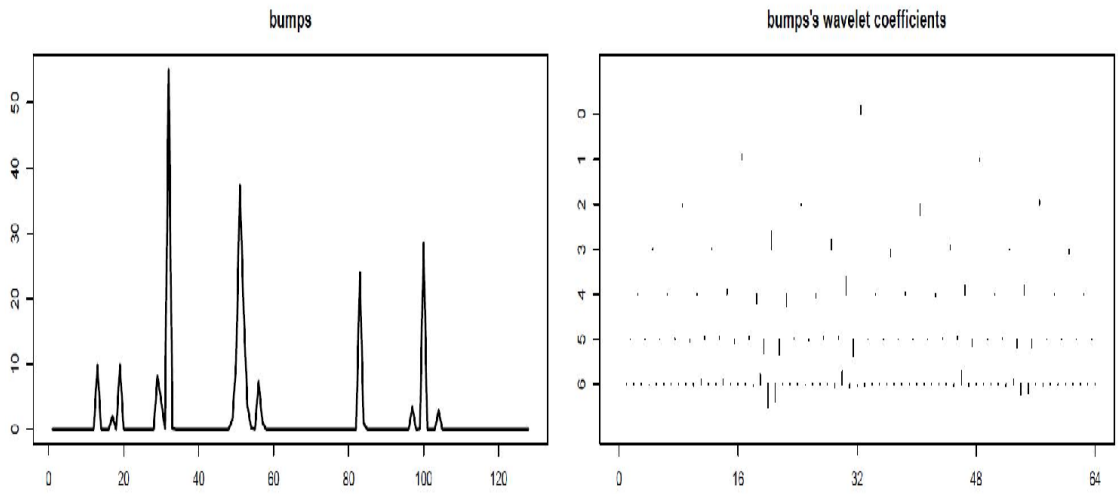


Figure 4.2: **DWT** coefficients from bumps function (the test function from Donoho & Johnstone (1994))

4.1.2 The DWT tree structure of one patient's data

Previously, we introduced the tree structure in **DWT** coefficients. However, after applying **DWT** in one patient's data, the **DWT** tree structure will be more complicated than that in the bumps signal. At the beginning of this chapter, the "noise" part and the "still" part are introduced. Now we will extract a slice of the patient's data (with length 1024) as an example to illustrate the "noise" and the "still" part in the **DWT** tree structure.

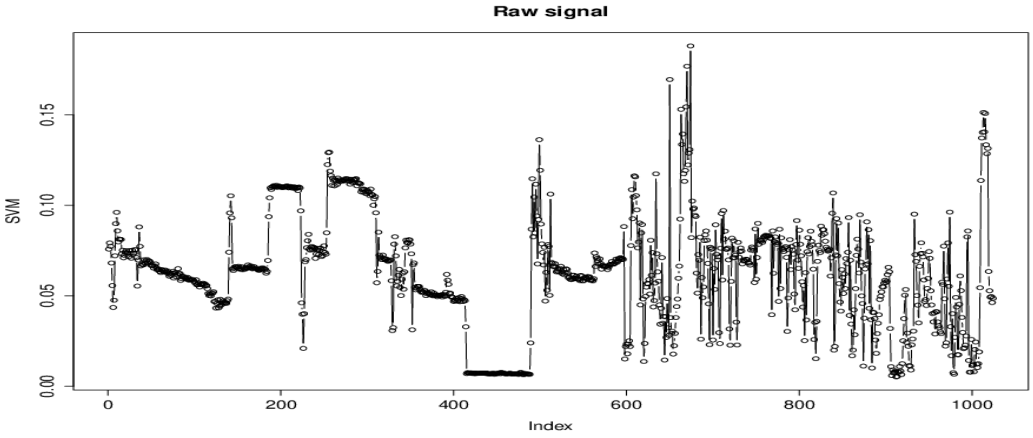


Figure 4.3: The slice of **VM** data from one patient.

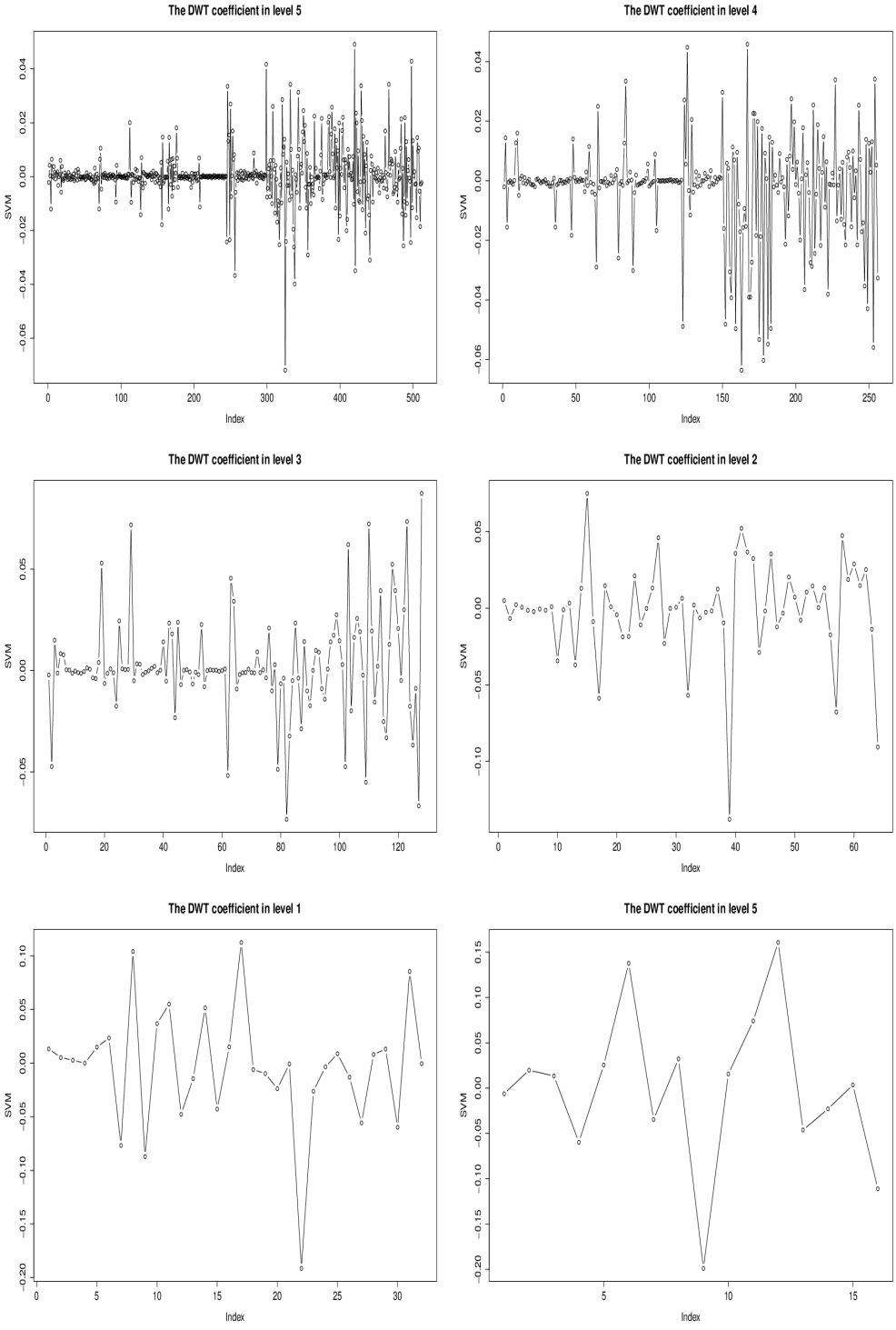


Figure 4.4: **DWT** coefficients from level 0 to level 5 among the slice of the patient's **VM** data.

Firstly, the **VM** data in **Figure** 4.3 shows the patient's different situation during a short period (about 17 min). The patient's activities can be separated into two parts,

"still" part and "active" part. If the patient has little activity, i.e. displays less movement from time 0 to 600, this is defined as "still" time. After time 600, the patient becomes active and moves frequently. This is referred to as "active" time. Now let us look at **DWT** coefficients in **Figure** 4.4 which transformed from the **VM** data through **DWT**. The "still" part in **VM** data corresponds to small **DWT** coefficients (close to 0) from level 0 to level 5. When referring to **DWT** coefficients related to the "active" part in the **VM** data, most of them are large (far away from 0), but there are still lots of **DWT** coefficients close to 0 especially at level 5 and level 4 in the "active" part. These are the "noise", which also provide little information and can be removed.

4.2 Bayesian wavelet regression

Let $y(t)$ be a process, e.g., accelerometer data, and $\mathbf{y} = (y(t_0), y(t_1), ..., y(t_{T-1}))$ are observation at $t = t_i$, $i = 0, ..., T-1$, then assume

$$\mathbf{y} = \mathbf{f} + \epsilon,\ \epsilon \sim N(0, \Sigma_\epsilon),$$

where $\mathbf{f} = (f(t_0), f(t_1), ..., f(t_{T-1}))$ is a vector of $f(t)$ evaluated at $t = t_i$, $i = 0, ..., T-1$ which corresponds to $\mathbf{y}$, the T is an integer power of 2, and $\Sigma_\epsilon = diag(\sigma_0^2, \sigma_1^2, ..., \sigma_{T-1}^2,)$. Assuming $\psi_{j,k}(x) = 2^{-\frac{j}{2}}\psi(2^{-j}x - k)$, $j, k \in \mathbf{Z}^*$ is the orthonormal wavelet basis where $\mathbf{Z}^*$ stands for all positive integer and 0. The corresponding **DWT** coefficient for an unknown function f and noise part ϵ can be written as $\mathbf{z} = \mathcal{W}\mathbf{f}$ and $\mathbf{u} = \mathcal{W}\epsilon$ respectively, where $\mathcal{W}$ is the orthonormal matrix which contains the orthonormal wavelet basis $\psi_{j,k}(x)$ (see **Section** 2.3.3). After applying **DWT** with orthonormal matrix $\mathcal{W}$ to accelerometer data $\mathbf{y}$, we obtain:

$$\begin{aligned} \boldsymbol{d} = \mathcal{W}\mathbf{y} &= \mathcal{W}\mathbf{f} + \mathcal{W}\epsilon \\ &= \boldsymbol{z} + \boldsymbol{u}. \end{aligned} \tag{4.1}$$

then Equaction (4.1) can be re-written as follows:

$$d_{jk} = z_{j,k} + u_{j,k},$$

where $z_{j,k}$ is the **DWT** coefficient from the $\mathbf{f}$ (unknown), and $u_{j,k} \sim N(0, \sigma_{j,k}^2)$, $\sigma_{j,k}$ is unknown. Bayesian inference on Equaction (4.1) is proceeded by placing priors on $z_{j,k}$ and $\sigma_{j,k}^2$ which are introduced in **Section** 4.2.1.

In our wrist-worn sensor AX3 data (see **Figure** 4.3), $\mathbf{y}$ is the accelerometer data. After applying **DWT**, terms d_{jk}, $z_{j,k}$ and $u_{j,k}$ can be expressed as index of j and k, which means the j-th **DWT** decomposition level and the k-th node. Note that, d_{jk} are **DWT**

coefficients from the accelerometer data, $z_{j,k}$ are **DWT** coefficients of $\mathbf{f}$ and $u_{j,k}$ are **DWT** coefficients from the noise.

It is very common that **DWT** coefficients $z_{j,k}$ can be separated to two types. The first type is the one in which its **DWT** coefficient is zero or close to zero. These **DWT** coefficients can be treated as "noise" which contain very little information and can be removed. Moreover, if **DWT** coefficients $z_{j,k}$ that are zero or near zero in our accelerometer data, this means that patients may be "still", and they don't move or even fall asleep. Another type includes the cases where their **DWT** coefficients are non-zero, indicating patients are doing activities, and these "large" **DWT** coefficients provide the main information for the upper limb movement, and thus provide information on modelling the recovery level of upper limb function after stroke. In the following section we use Bayesian inference on the above model, adding the prior on the $z_{j,k}$ and the parameter $\sigma^2_{j,k}$ in $u_{j,k}$ to remove those zero or near zero **DWT** coefficients.

4.2.1 The spike-and-slab mixture prior on $z_{j,k}$

Due to the energy concentration, it is very important that shrinking smaller coefficients (corresponding to noise) towards zero in order to keep large coefficients containing more information in the model (Chipman et al., 1996). The next step is to use a popular Bayesian strategy to deal with it, which is derived from imposing a particular prior structure to the model, and so-called spike-and-slab mixture prior (Chipman et al. (1997), CLYDE et al. (1998) and Clyde & George (2000)) on **DWT** coefficients $z_{j,k}$.

With the mixture prior probability $\pi_{j,k}$, **DWT** coefficients $z_{j,k}$ are separated into two groups which is called a spike-and-slab mixture model and defined as follows:

$$z_{j,k} \sim (1 - \pi_{j,k})\delta(0) + \pi_{j,k} N(0, \tau_j \sigma^2_{j,k}).$$

DWT coefficients $z_{j,k}$ contain a few large **DWT** coefficients and many small **DWT** coefficients. The term $N(0, \tau_j \sigma^2_{j,k})$ describes the large **DWT** coefficients and the term $\delta(0)$ describes small **DWT** coefficients. Also the spike part in spike-and-slab mixture model does not have to be exactly at 0 but can be a Gaussian with zero mean and a much smaller variance. The term $\tau_j = 2^{-\alpha j}\tau$, which is a level-specific parameter, can be presented as the entire level of variability in **DWT** coefficients at the level j, where $\alpha, \tau > 0$. Note that the terms α and τ can be obtained from an empirical Bayes approach by maximizing the marginal likelihood (see **Section** 4.3).

For convenience, we define a latent variable $S_{j,k}$ to indicate the hidden state of $z_{j,k}$ depends on the prior probability $\pi_{j,k}$, such that $S_{j,k} = 0$ if $z_{j,k}$ is zero (or close to zero) and $S_{j,k} = 1$ otherwise. Thus, $Pr(S_{j,k} = 0) = \pi_{j,k}$ and $Pr(S_{j,k} = 1) = 1 - \pi_{j,k}$. The previous spike-and-slab prior can be written as follows:

$$z_{j,k}|S_{j,k} \sim N(0, S_{j,k} \cdot \tau_j \sigma_{j,k}^2),\ S_{j,k} \in \{0,1\}. \tag{4.2}$$

In other words, when $S_{j,k} = 1$, **DWT** coefficient $d_{j,k}$ (from accelerometer data) is far away from 0. When $S_{j,k} = 0$, **DWT** coefficient $d_{j,k}$ (from accelerometer data) is close to 0. The error variance $\sigma_{j,k}^2$ is unknown and inferred from the data. Since the Inverse-Gamma distribution is a conjugate of the Gaussian distribution, we allow $\sigma_{j,k}^2$ to be heterogeneous and with a hyperprior on it:

$$\sigma_{j,k}^2 \overset{\text{i.i.d}}{\sim} Inv - Gamma(v+1, v\sigma_0^2). \tag{4.3}$$

where the terms v and σ_0 are the hyperparameters in NIG-MT, which we can use with the maximum marginal likelihood estimators (MMLE) in **Section** 4.3 to estimate it from the accelerometer data.

4.2.2 Essence of hidden state $S_{j,k}$ in DWT coefficients tree structure

Some properties of **DWT** coefficients have been discussed in **Section** 4.1. **DWT** coefficients are represented as a tree structure (see **Figure** 4.1), the tree structure shows that if the value of parent node is non-zero, the corresponding value of two children nodes may also be away from zero, otherwise, that will be close or equal to zero. This property of **DWT** coefficients in tree structure shows that there is a very strong Markov property between the parent node and its two children nodes, in other words, the value of two children nodes have been only related and determined by the value of their parent node at previous level.

Given the hidden state $S_{j,k} \in \{0,1\}$, after applying **DWT** in the accelerometer data $\mathbf{y}$, we get a tree structure of **DWT** coefficients. Recall that the pair of indices (j,k) represent the j-th level discrete wavelet transform and the k-th node, and use $\mathcal{T}$ to represent indices (j,k) corresponding to all nodes of **DWT** coefficients. Note that, two children nodes of location (j, k) are indexed by ($j+1$, $2k$) and ($j+1$, $2k+1$). Generally, when $j > 1$, the parent of (j, k) is indexed by ($j-1$, $\frac{k}{2}$).

Since the Markov property is represented between the parent node (j, k) and two children nodes ($j+1$, $2k$), ($j+1$, $2k+1$), the latent state $S_{j,k} \in \{0,1\}$, which can be applied with Markov process. The previous model can be expressed by a hidden Markov model evolving into a Markov tree. Under the Markov tree, $\mathcal{S} = \{S_{j,k} : (j,k) \in \mathcal{T}\}$, each node ($j$,$k$) depends on its parent, which is expressed by a Markov transition as followed:

$$Pr(S_{j,k} = s' | S_{j-1,[k/2]} = s) = \rho_{j,k}(s, s'). \tag{4.4}$$

In other words, the transition probabilities $\rho_{j,k}(s,s')$, $s, s' \in \{0,1\}$, which are written as a

transition matrix for each node $(j,k) \in \mathcal{T}$ in the Markov tree as follows:

$$\rho_{j,k} = \begin{bmatrix} \rho_{j,k}(0,0) & \rho_{j,k}(0,1) \\ \rho_{j,k}(1,0) & \rho_{j,k}(1,1) \end{bmatrix} = \begin{bmatrix} max\{1-\eta 2^{-j}, 0\} & min\{\eta 2^{-j}, 1\} \\ 1-\gamma & \gamma \end{bmatrix}.$$

Parameters η and γ satisfy $\eta > 0$ and $0 < \gamma < 1$ respectively. The larger the η, the larger the **DWT** coefficients will be contained at the j-th level in tree structure. In contrast, the smaller η means **DWT** coefficients at the j-th level containing more zero (or close to zero) coefficients. The large γ implies that there is a strong correlation in large coefficients.

Since root nodes (0, k) in scale 0 do not have parents, it is better to set a specified initial state probability $\rho_{0,k} = (\rho_{0,k}(0), \rho_{0,k}(1))$. The probability of $S_{j,k}$ for all location k in scale $j = 0$ can be defined as:

$$Pr(S_{0,k} = s) = \rho_{0,k}(s), \qquad s \in \{0,1\}. \tag{4.5}$$

Combining **Equations** (4.4) and (4.5), the marginal probabilities of $S_{j,k}$ for location k in scale $j = 1, 2, ..., J$ follows:

$$Pr(S_{j,k} = s') = \sum_{s \in \{0,1\}} Pr(S_{j-1,[k/2]} = s) Pr(S_{j,k} = s' | S_{j-1,[k/2]} = s).$$

To be more specific, each node of state probabilities $\rho_{j,k} = (\rho_{j,k}(0), \rho_{j,k}(1))$ in scale $j = 1, 2, ..., J$ is expressed as:

$$\begin{aligned} \rho_{j,k}(0) &= \rho_{j-1,[k/2]}(0)\rho_{j,k}(0,0) + \rho_{j-1,[k/2]}(1)\rho_{j,k}(1,0), \\ \rho_{j,k}(1) &= \rho_{j-1,[k/2]}(0)\rho_{j,k}(0,1) + \rho_{j-1,[k/2]}(1)\rho_{j,k}(1,1). \end{aligned}$$

Until now, we have obtained the probability of each latent state S_{jk} in the Markov tree ($\mathcal{S} = \{S_{j,k} : (j,k) \in \mathcal{T}\}$) by given the specified initial state probability $\rho_{0,k}$ and the transition probability $\rho_{j,k}(s, s')$.

Combining **Equations** (4.2) (4.3) (4.4) and (4.5), the new hierarchical model for the **DWT** coefficients is obtained. It is called the Normal Inverse-Gamma Markov tree model (**NIG-MT**) and is specified by the hyperparameters $\boldsymbol{\theta} = (\alpha, \tau, \upsilon, \sigma_0^2, \eta, \gamma, \rho_{0,k}(1))$ (See Ma & Soriano (2017)).

4.2.3 Inference under the NIG-MT model

In particular, the Markov tree structure and the Normal Inverse-Gamma setup are completely conjugate to the **NIG-MT**: the joint posterior on $\{S_{jk}, z_{j,k}, \sigma_{j,k}^2 : (j,k) \in \mathcal{T}\}$

is still an **NIG-MT**.

Let us consider the n-th functional observations $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ...\mathbf{y}^{(n)}$ (different sets of accelerometer data among different patients) from **Equation** (4.1). In the general case, for the i-th observation, the $\mathbf{d_{j,k}}^{(i)}$ is the **DWT** coefficient at scale j and location k of $\mathbf{y}^{(i)}$. After applying **DWT** with the accelerometer data, the model resulting at each specified node is:

$$d_{jk}^{(i)} = z_{j,k} + u_{j,k}^{(i)} \quad where \quad u_{j,k}^{(i)} \sim N(0, \sigma_{j,k}^2). \tag{4.6}$$

The aim is to find the posterior distribution on $\{S_{j,k}, z_{j,k}, \sigma_{j,k}^2 : (j,k) \in \mathcal{T}\}$ given the observed **DWT** coefficients d_{jk}. Let $m_{j,k}(s)$ to be the marginal likelihood for the node-specific model on (j,k), given $S_{j,k} = s \in \{0,1\}$:

$$m_{j,k}(s) = \int p(\boldsymbol{d}^{(1)}, \boldsymbol{d}^{(2)}, ..., \boldsymbol{d}^{(n)} | S_{j,k} = s, z_{j,k}, \sigma_{j,k}^2) \pi(z_{j,k}, \sigma_{j,k}^2) dz_{j,k} d\sigma_{j,k}^2. \tag{4.7}$$

From the normal-inverse-Gamma conjugacy, the marginal likelihood is in closed form:

$$m_{j,k}(s) = \frac{(v\sigma_0^2)^{v+1}\Gamma(v+n/2+1)}{(2\pi)^{n/2}\Gamma(v+1)} \cdot \left[\frac{\tau_j^{-1}}{n+\tau_j^{-1}}\right]^{s/2} \cdot \left[v\sigma_0^2 + \frac{1}{2}\left(\sum_i (d_{jk}^{(i)})^2\right) - s \cdot \frac{(n\overline{d}_{jk})^2}{n+\tau_j^{-1}}\right]^{-v-n/2-1}, \tag{4.8}$$

where $\overline{d}_{jk} = \sum_i d_{jk}^{(i)}/n$. Note that, the term $\mathcal{D} = (d_{jk}^{(1)}, d_{jk}^{(2)}, ..., d_{jk}^{(n)})$ should be used to represent the total accelerometer data among different patients in the following content.

Since the **NIG-MT** model is completely conjugated to the Markov tree (**MT**) structure and Normal Inverse-Gamma, the posterior of the hidden states S_{jk} is still a **MT**. Referring to the definition of $\phi_{j,k}$ and $\xi_{j,k}$, which can be found in **Equation** (4.9), the posterior state transition probabilities of $S_{j,k}$ are:

$$Pr(S_{j,k} = s' | S_{j-1,[k/2]} = s, \mathcal{D}) = \rho_{j,k}(s,s') \frac{\phi_{j,k}(s')}{\xi_{j,k}(s)}, \ s, s' \in \{0,1\}, \ j \geq 1.$$

Using matrix notation, the posterior state transition matrix of $S_{j,k}$ in order $j = 1, 2, ..., J$ can be expressed as:

$$[\rho_{j,k}|\mathcal{D}] = \begin{bmatrix} \rho_{j,k}(0,0)\frac{\phi_{j,k}(0)}{\xi_{j,k}(0)} & \rho_{j,k}(0,1)\frac{\phi_{j,k}(1)}{\xi_{j,k}(0)} \\ \rho_{j,k}(1,0)\frac{\phi_{j,k}(0)}{\xi_{j,k}(1)} & \rho_{j,k}(1,1)\frac{\phi_{j,k}(1)}{\xi_{j,k}(1)} \end{bmatrix}.$$

The posterior initial state probabilities of $S_{j,k}$ are:

$$Pr(S_{0,k} = s | \mathcal{D}) = \rho_{0,k}(s) \frac{\phi_{0,k}(s)}{\xi_{0,k}(0)}, \ s \in \{0,1\}.$$

Using vector notation, the posterior initial state probabilities of $S_{j,k}$ in order $j = 0$ are given by:

$$[\rho_{0,k}|\mathcal{D}] = \left(\rho_{0,k}(0)\frac{\phi_{0,k}(0)}{\xi_{0,k}(0)},\ \rho_{0,k}(1)\frac{\phi_{0,k}(1)}{\xi_{0,k}(0)}\right).$$

Note that the terms of $\phi_{j,k}(s)$ and $\xi_{j,k}(s)$ can be calculated by using a bottom-up pyramid algorithm, that is very similar to the Mallat's pyramid algorithm (Mallat, 1989) for running **DWT**. The bottom-up pyramid can be written as followed:

$$\begin{aligned}
\phi_{j,k}(s) &= \begin{cases} m_{jk}(s) & for \quad j = J, \\ m_{jk}(s)\cdot\xi_{j+1,2k}(s)\cdot\xi_{j+1,2k+1}(s) & for \quad j = 0,1,2,...,J-1, \end{cases} \\
\xi_{j,k}(s) &= \begin{cases} \sum\limits_{s'\in\{0,1\}} \rho_{j,k}(s,s')\cdot\phi_{j,k}(s') & for \quad j = 1,2,...,J, \\ \sum\limits_{s'\in\{0,1\}} \rho_{0,k}(s')\cdot\phi_{j,k}(s') & for \quad j = 0. \end{cases}
\end{aligned} \tag{4.9}$$

So far, we achieve the posterior state transition probabilities and initial state probabilities in S_{jk} through the $\phi_{j,k}(s)$'s and the $\xi_{j,k}(s)$'s by using the bottom-up pyramid, and use the following top-down pyramid algorithm to compute the posterior marginal probability of $S_{j,k}$. Then the posterior marginal probabilities of $S_{j,k}$ will be obtained:

$$Pr(S_{j,k} = s') = \sum_{s\in\{0,1\}} Pr(S_{j-1,[k/2]} = s|\mathcal{D})Pr(S_{j,k} = s'|S_{j-1,[k/2]} = s, \mathcal{D}),\ j \geq 1.$$

The posterior of σ_{jk}, z_{jk} and the mean of $\mathbf{z}$ will also be achieved. The posterior of variances σ^2_{jk} given S_{jk} is:

$$[\sigma_{jk}|S_{jk},\mathcal{D}] \sim Inv-Gamma\left(v+1+n/2,\ v\sigma_0^2+\frac{1}{2}\left(\sum_i (d^{(i)}_{jk})^2\right) - \cdot\frac{S_{jk}\cdot(n\overline{d}_{jk})^2}{n+\tau_j^{-1}}\right).$$

The posterior of variances z_{jk} given S_{jk}, σ_{jk} is:

$$[z_{jk}|\sigma^2_{jk},S_{jk},\mathcal{D}] \sim N\left(\frac{S_{jk}\cdot n\overline{d}_{jk}}{n+\tau_j^{-1}}\right)\cdot\left(\frac{S_{jk}\cdot\sigma^2_{jk}}{n+\tau_j^{-1}}\right).$$

Then the posterior mean of $\mathbf{z}$ is given by:

$$\widetilde{z}_{jk}| = E\left(z_{j,k}|\mathcal{D}\right) = Pr(S_{jk}=1|\mathcal{D})\cdot\frac{n\overline{d}_{jk}}{n+\tau_j^{-1}}.$$

The average of the observed wavelet coefficients $\overline{d_{jk}}$ is shrunk toward the prior mean 0 with the amount of shrinkage being averaged over the different shrinkage states. By

applying an inverse DWT to $\widetilde{z}_{jk}$ we can get the posterior mean of $\boldsymbol{f}$.

4.3 Prior specification

The Normal Inverse-Gamma Markov tree model (**NIG-MT**) is specified by the following hyperparameters $\boldsymbol{\theta} = (\alpha, \tau, v, \sigma_0, \eta, \gamma, \rho_{0,k}(1))$. In order to force smaller coefficients (corresponding to the noise) to zero and to ensure that large coefficients remain unaffected by using proper priors in the **NIG-MT**, it is important to discuss the specification of hyper-parameters $\boldsymbol{\theta}$ in turn.

The hyperparameters α and τ

The term $\tau_j = 2^{-\alpha j}\tau$ implies that **DWT** coefficients tend to be smaller as the level j increases. The hyperparameter α controls the smoothness of the functional observation, that is, the larger the hyperparameter α, the smoother the functional observation. As recommended in Abramovich et al. (1998), $\alpha = 0.5$. Alternatively, a more flexible way to choose both α and τ is, suggested by Ma & Soriano (2017), using the maximum marginal likelihood estimation (MMLE), since the functional observation tends to be rough in our real word, especially in the free living data (containing too much noise) from AX3.

The hyperparameters v and σ_0

From the equation (4.2) and (4.3), **DWT** coefficients from **f** for each node (j,k) in the **DWT** markov tree structure follow a normal distribution $N(0, S_{j,k} \cdot \tau_j \sigma^2_{j,k})$ and $\sigma^2_{j,k} \overset{\text{i.i.d}}{\sim} Inv - Gamma(v+1, v\sigma_0^2)$. The term $v\sigma_0^2$ mainly corresponds to the noise from $d_{j,k}$ at the lowest decomposition level, especially in the high frequency domain. The hyperparameter σ_0 is very similar to the hyperparameter τ. The large parameter σ_0^2 means that our functional observation tends to be smooth. Regarding the difference between σ_0 and τ, τ is more related to the overall trends of functional observation smoothness. However, σ_0 relies more on the total energy of detailed coefficients in the high frequency domain.

The hyperparameters η, γ and $\rho_{0,k}(1)$

The hyperparameter γ determines the spatial-scale dependency of the wavelet signal. Larger γ values correspond to stronger correlation or clusters in the large wavelet coefficients. On the other hand, the hyperparameter η controls how likely it is to have a non-zero wavelet coefficient at each level. The exponential decaying factor 2^{-j} counters exactly the exponential increase in the expected number of wavelet coefficients at higher resolution, and keeps the prior expected number of "de novo signals" (in the sense that a

node contains a signal but its parents do not) at each resolution fixed at η. To this end, the hyperparameters $\rho_{0,k}(1)$ represent how likely the root nodes d_{0k} at level 0 are to be big or not.

Maximum marginal likelihood

The overall marginal likelihood $P(\mathcal{D}|\theta) = \sum_k \xi_{0,k}(0)$, in location k and level 0, is calculated using the bottom-up pyramid algorithm from **Equation** (4.9). Hence, the maximum marginal likelihood estimators (MMLE), with unconstrained and box-constrained optimization using PORT routines for the hyperparameters, will be:

$$\widehat{\theta} = \arg\max_{\theta} P(\mathcal{D}|\theta).$$

Note that the MMLE is implemented by the R function **nlminb**, and the detail for carrying out the optimization can be found in **Section** 4.4.1).

4.4 Identifying non-zero DWT coefficients

The general case of the **NIG-MT** model has been discussed in the previous section. **Equation** (4.6) gives a general case when we have i functional observations corresponding to i sets of accelerometer data in the i-th stroke patients. When we input the accelerometer data for all stroke patients in the **NIG-MT** model, a problem will appear due to different habits among different stroke patients. To be more specific, if patients wear the wrist-worn sensor AX3 for 3 full days, they have a different routine of daily life during this time. For example, returning to **Figure** 3.2 and **Figure** 3.5 in **Chapter** 3, in a certain period, Patient la038 is very "active" while Patient la027 stays "still". If we input all accelerometer data together, the output of the posterior will be very unstable in the **NIG-MT** model. Such "undesirable" posteriors will be shown in **Figure** 4.6.

Due to the complexity of patients' data mentioned above, rather than inputting the entire data set $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ...\mathbf{y}^{(n)}$ of total patients into **Equation** (4.6), we consider a single set of accelerometer data $\mathbf{y}$ for each patient in the **NIG-MT** model and repeat it until we obtain the posterior for each single patient from **Equation** (4.1). After applying **DWT** with accelerometer data, the model at each specified node, for each patient, from **Equations** (4.6) (4.7) and (4.8) becomes:

$$d_{jk} = z_{j,k} + u_{j,k} \quad where \quad u_{j,k} \sim N(0, \sigma^2_{j,k}). \tag{4.10}$$

Let $m_{j,k}(s)$ be the marginal likelihood for the node-specific model on (j,k), given

$S_{j,k} = s \in \{0, 1\}$:

$$m_{j,k}(s) = \int p(\boldsymbol{d}|S_{j,k} = s, z_{j,k}, \sigma^2_{j,k})\pi(z_{j,k}, \sigma^2_{j,k})dz_{j,k}d\sigma^2_{j,k}. \tag{4.11}$$

From the normal Inverse-Gamma conjugacy, the marginal likelihood is in closed form:

$$m_{jk}(s) = \frac{(v\sigma_0^2)^{v+1}\Gamma(v + 3/2)}{(2\pi)^{1/2}\Gamma(v + 1)} \cdot \left[\frac{\tau_j^{-1}}{1 + \tau_j^{-1}}\right]^{s/2} \cdot \left[v\sigma_0^2 + \frac{1}{2}\left((d_{jk})^2 - s \cdot \frac{(d_{jk})^2}{1 + \tau_j^{-1}}\right)\right]^{-v-3/2}. \tag{4.12}$$

We aim to find the posterior distribution of $\{S_{j,k} : (j, k) \in \mathcal{T}\}$ for each single patient given the observed **DWT** coefficients d_{jk} from accelerometer data. The posterior distribution of $\{S_{j,k} : (j, k) \in \mathcal{T}\}$ consists of two parts; the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ and the posterior marginal probabilities $Pr(S_{j,k} = 0|\mathcal{D})$. We just consider $Pr(S_{j,k} = 1|\mathcal{D})$ in our model since posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D}) = 1 - Pr(S_{j,k} = 0|\mathcal{D})$.

After applying the **NIG-MT** model, the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ deliver the valued information about each **DWT** coefficient d_{jk} with location (j,k) in the whole **DWT** coefficients tree structure. We then know each **DWT** coefficient d_{jk} with location (j,k) will be zero or non-zero. The zero coefficients provide little information on modelling upper limb function and can be removed. This results in a data reduction.

To be more specific, we analyze the posterior marginal probabilities of $S_{j,k}$ for the node-specific model on (j,k). The posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ plays a very important role in the **NIG-MT** model, since the value of $Pr(S_{j,k} = 1|\mathcal{D})$ is between 0 and 1. If the **DWT** coefficients are very small (corresponding to "noise" or "still"), the value of $Pr(S_{j,k} = 1|\mathcal{D})$ will be close to 0 at all **DWT** decomposition levels. Otherwise, the value of $Pr(S_{j,k} = 1|\mathcal{D})$ will be close to 1.

The overall objective is to do the data reduction through the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ in the **NIG-MT** model. For the whole **DWT** coefficient tree structure, we keep the **DWT** coefficient d_{jk} in location (j,k) if its $Pr(S_{j,k} = 1|\mathcal{D})$ is close to one, otherwise, it will be forced to be zero. In other words, the process shrinks smaller coefficients (corresponding to "noise" or "still") to zero and keeps large coefficients (non-zero **DWT** coefficients).

As long as the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ which are given from the data obtained, we will obtain an acceptable outcome with the accelerometer data. More specifically, based on the $Pr(S_{j,k} = 1|\mathcal{D})$ among all the $d_{j,k}$ from acclelerometer data, we can distinguish which is close to zero, the details will be discussed in the next step.

The next step is to decide how to find the appropriate $Pr(S_{j,k} = 1|\mathcal{D})$, which depends

on the hyperparameters $\theta = (\alpha, \tau, v, \sigma_0, \eta, \gamma, \rho_{0,k}(1))$ as discussed in **Section** 4.3. The proper posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ will distinguish non-zero **DWT** coefficients directly shown in **Figure** 4.5, which is a sort of "acceptable" posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$. Big **DWT** coefficients correspond to the $Pr(S_{j,k} = 1|\mathcal{D})$ close to 1, whereas small **DWT** coefficients will be close to 0 with $Pr(S_{j,k} = 1|\mathcal{D})$. However, **Figure** 4.6 shows the $Pr(S_{j,k} = 1|\mathcal{D})$ is about 0.1 irrespective of the value of the corresponding **DWT** coefficients. We therefore need to choose the hyperparameters carefully.

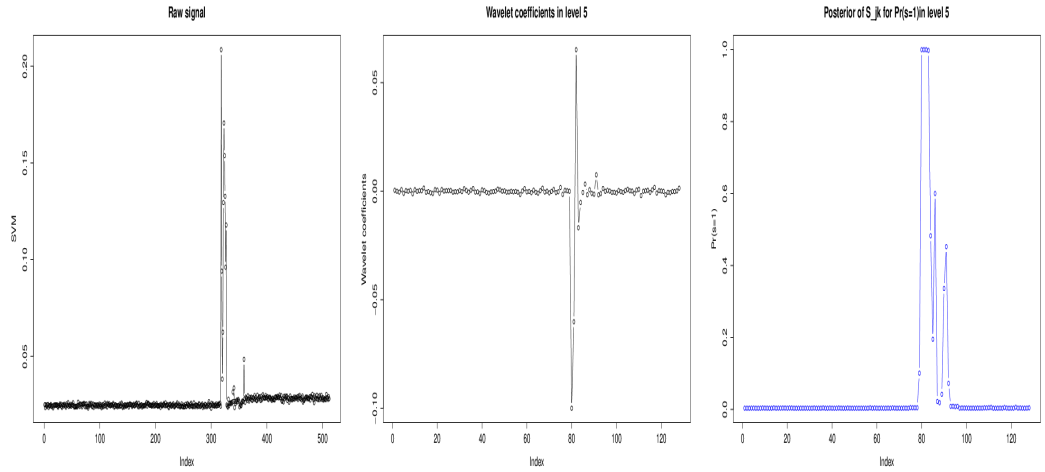


Figure 4.5: "Acceptable" posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$. More specifically, the large **DWT** coefficients correspond to the $Pr(S_{j,k} = 1|\mathcal{D})$ close to 1, whereas small **DWT** coefficients will be close to 0.

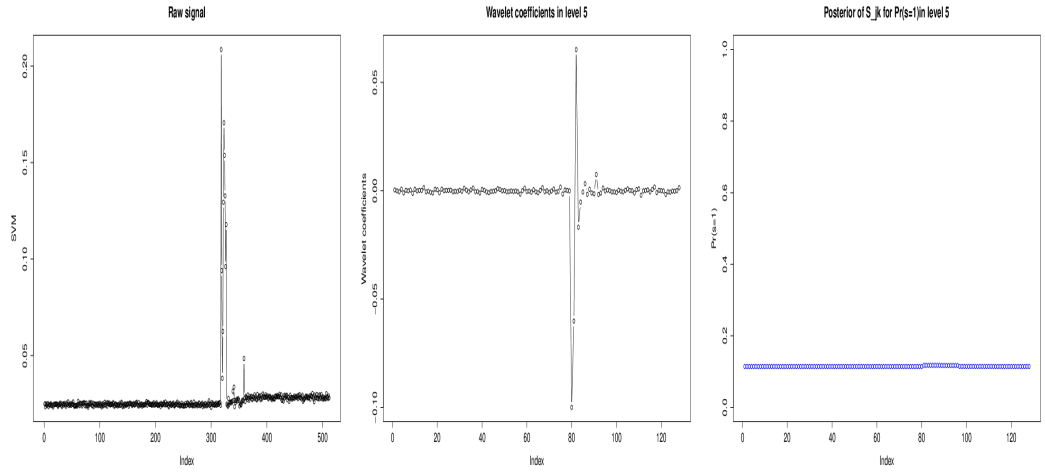


Figure 4.6: "Unacceptable" posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$. More specifically, the value of $Pr(S_{j,k} = 1|\mathcal{D})$ is about 0.1 irrespective of the value of the corresponding **DWT** coefficients

4.4.1 Sensitivity of initial value of hyperparameters in the posterior of $Pr(S_{j,k} = 1|\mathcal{D})$

We choose a slide window from one of the free-living data, apply the **DWT** and focus on **DWT** coefficients at level 5. Through the maximum marginal likelihood estimators (MMLE) and, based on the specification of hyperparameters $\boldsymbol{\theta}$ (see **Section** 4.3), we set a vector of seven hyperparameters from $\boldsymbol{\theta}$ with a wide range of lower and upper bounds. Here we choose the lower bound and upper bound with $\boldsymbol{\theta}_{lower} = (0.1, 0.001, 0.05, 0.1, 0.05, 0.1, 0.1)$ and $\boldsymbol{\theta}_{upper} = (10, 0.2, 5, 20, 20, 0.9, 0.9)$ respectively. We now need the initial values for the hyperparameters $\boldsymbol{\theta}$ to be optimized. Here, we should notice that if the posterior is sensitive to the initial values of the hyperparameters, it means that the numerical routine for global optimisation is not working as well as hoped in our MMLE. The next step is to change one of initial values of hyperparameters and fix the remaining six initial hyperparameters from $\boldsymbol{\theta}$, and then check the sensitivity of different initial hyperparameters in the posterior of S_{jk} at the **DWT** decomposition level 5 around the initial value: $\boldsymbol{\theta}_{(ini)} = (\upsilon_{(ini)}, \sigma 0_{(ini)}, \alpha_{(ini)}, \tau_{(ini)}, \eta_{(ini)}, \gamma_{(ini)}, \rho_{0,k}(1)_{(ini)})$.

It has been found that, from the **Figures** 4.11 to 4.17 in **Appendix** 4.7, the posterior is defined mathematically and are not sensitive to the initial values of the hyperparameters in MMLE by using R package **nlminb**. However, the final posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ is consistent when we input the different initial values in all the hyperparameters: υ, σ_0 α, τ, η, γ and $\rho_{0,k}(1)$.

4.4.2 The threshold P in $Pr(S_{j,k} = 1|\mathcal{D})$

As shown in **Figures** 4.5 and 4.6 , the "acceptable" posterior value of $Pr(S_{j,k} = 1|\mathcal{D})$ is close to either 0 or 1, and so the **DWT** coefficients with a near zero value of $Pr(S_{j,k} = 1|\mathcal{D})$ can be forced to be zero from the whole data set. However, the values of some **DWT** coefficients are neither near 0 nor near 1. We therefore need to set a threshold P such that the $d_{j,k}$ will be treated as zero, if $Pr(S_{j,k} = 1|\mathcal{D}) \leq P$, i.e., the corresponding **DWT** coefficients can be forced to be zero. We will investigate the potential values of P at 0.2, 0.3,... or 0.9. To evaluate those thresholds through the energy concentration in the accelerometer data, we let $\mathbf{LE}_j$ be the losing energy at each **DWT** decomposition level:

$$\mathbf{LE}_j = \frac{\left\|(\mathbf{W}_j - \mathbf{W}_j^*)\right\|^2}{N/2^j} = 2^j \frac{\left\|(\mathbf{W}_j - \mathbf{W}_j^*)\right\|^2}{N},$$

where $\mathbf{W}_j$ represents the detailed coefficients in the j-th **DWT** scale decomposition from accelerometer data and $\mathbf{W}_j^*$ represents detailed the coefficients after shrinking. It should be noted that $\mathbf{LE}_j$ is the point energy loss after shrinking the small **DWT** to zero at **DWT** decomposition level j.

In our study, $\mathbf{ELR}_j$, the energy losing rate, is defined as follows,

$$\mathbf{ELR}_j = \frac{\mathbf{LE}_j}{\mathbf{SSD_j}},$$

where $\mathbf{SSD_j}$ is the point energy at each **DWT** decomposition level (see the details in **Section** 3.2.1). $\mathbf{ELR}_j$ is a ratio with its value between 0 and 1, the bigger the value is, the more energy will be lost through data reduction. To this end, a different threshold, P, will be evaluated through $\mathbf{ELR}_j$ from the energy concentration. Specific results will be reported in the next section.

4.5 Data-reduction for AX3 data

Based on the analysis from the posterior state probabilities of $S_{j,k}$ by using **NIG-MT** model, as we discussed before, non-zero **DWT** coefficients are found through a real case as follows. Now, one set of AX3 data, in the group of acute patients as an example, were collected from the 8-*th* week of Patient la215 (non-paralysed side) and are presented in **Figure** 4.7, through the information obtained from posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$, shrinking **DWT** coefficients adaptively and efficiently at 6 decomposition levels.

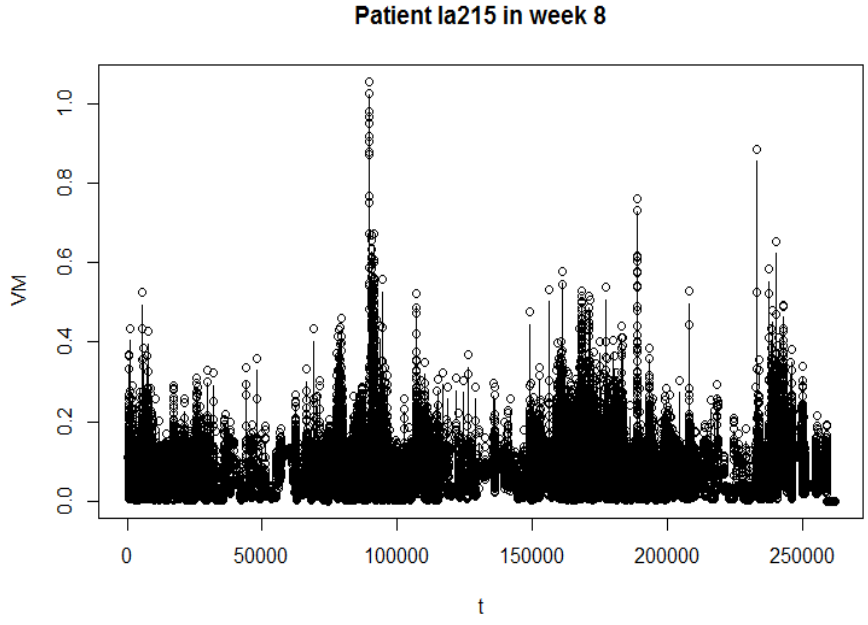


Figure 4.7: The **VM** data of Patient la215 in week 8.

As we discussed before, data in some periods include "noise" only (or corresponding

to the "still" state). Those data provide little information and can be forced to be zero. We apply **NIG-MT** model to fulfill this task.

We first calculate **DWT** for the data set (Patient la215, week 8), and then apply the **NIG-MT** model to this data and calculate the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ at 6 **DWT** decomposition levels corresponding to the threshold $P = 0.2, 0.3, ..., 0.9$. When we deal with the accelerometer data, the value of P should be based on the energy loss rate. The smaller value of ELR_j is the better, basically the sum of ELR_j at different decomposition levels needs to be smaller than 0.1, which means that we at least keep 90% of the energy after shrinking from the raw data. Similarly, we can deal with the other different types of data (e.g., not the accelerometer data) in the same way. In practice, we recommend a selected value of P as 0.5 at first, then check the value of ELR_j; if the sum of ELR_j at different decomposition j is small (e.g., less than 0.1), we can increase the value of threshold P. Here we choose the threshold P to be 0.5 at first, calculating the value of ELR_j and the percentage of shrinking. After **NIG-MT**, the percentage of shrinking is the ratio between non-zero **DWT** coefficients and raw **DWT** coefficients at different decomposition levels. For example, the length of the raw **DWT** coefficients and non-zero **DWT** coefficients (after **NIG-MT**) are m and m^* at level j respectively; the percentage of shrinking at level j is defined as $\frac{m-m^*}{m}$. In other words, it means that the number of raw **DWT** coefficients shrink proportionately at decomposition level j after **NIG-MT**. The ELR_j and percentage of shrinking based on threshold $P = 0.5$ are given as follows:

DWT level	$Pr(S_{j,k} = 1) < P$	$\mathbf{ELR}_j$	Percentage of shrinking
0	0.5	0.0001048405	0.371582
1	0.5	0.0002745973	0.4554443
2	0.5	0.001009772	0.5601807
3	0.5	0.001742959	0.6507263
4	0.5	0.002722701	0.7135925
5	0.5	0.003815816	0.7331772

Table 4.2: The threshold $P = 0.5$ with $\mathbf{ELR}_j$ and Percentage of shrinking.

From **Table** 4.2, we find that energy loss rate $\mathbf{ELR}_j$ is smaller than 0.01 at each decomposition level. Moreover, comparing with the raw DWT coefficients, there are about 37%, 45%, 56%, 65%, 71% and 73% of **DWT** coefficients shrinking through the data reduction procedure at decomposition level 0, 1, 2, 3, 4 and 5 respectively. Since the $\mathbf{ELR}_j$ are very small when we choose the threshold $P = 0.5$, based on our previous discussion of how to select the threshold P in practice, if the sum of energy loss rate $\mathbf{ELR}_j$ at different decomposition level j is smaller than 0.1 with threshold $P = 0.5$, then

the value of threshold P should be increased.

Now we choose the threshold P to be 0.9 as an example, which means that **DWT** coefficients will shrink most of the "noise" and remove the "still" period. In other words, if $Pr(S_{j,k} = 1|\mathcal{D}) > 0.9$, the corresponding $d_{j,k}$ will be kept while the others will be forced to be zero. The posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ and **DWT** coefficients, from level 0 to level 5, are shown in **Figures** 4.8 and 4.9 respectively.

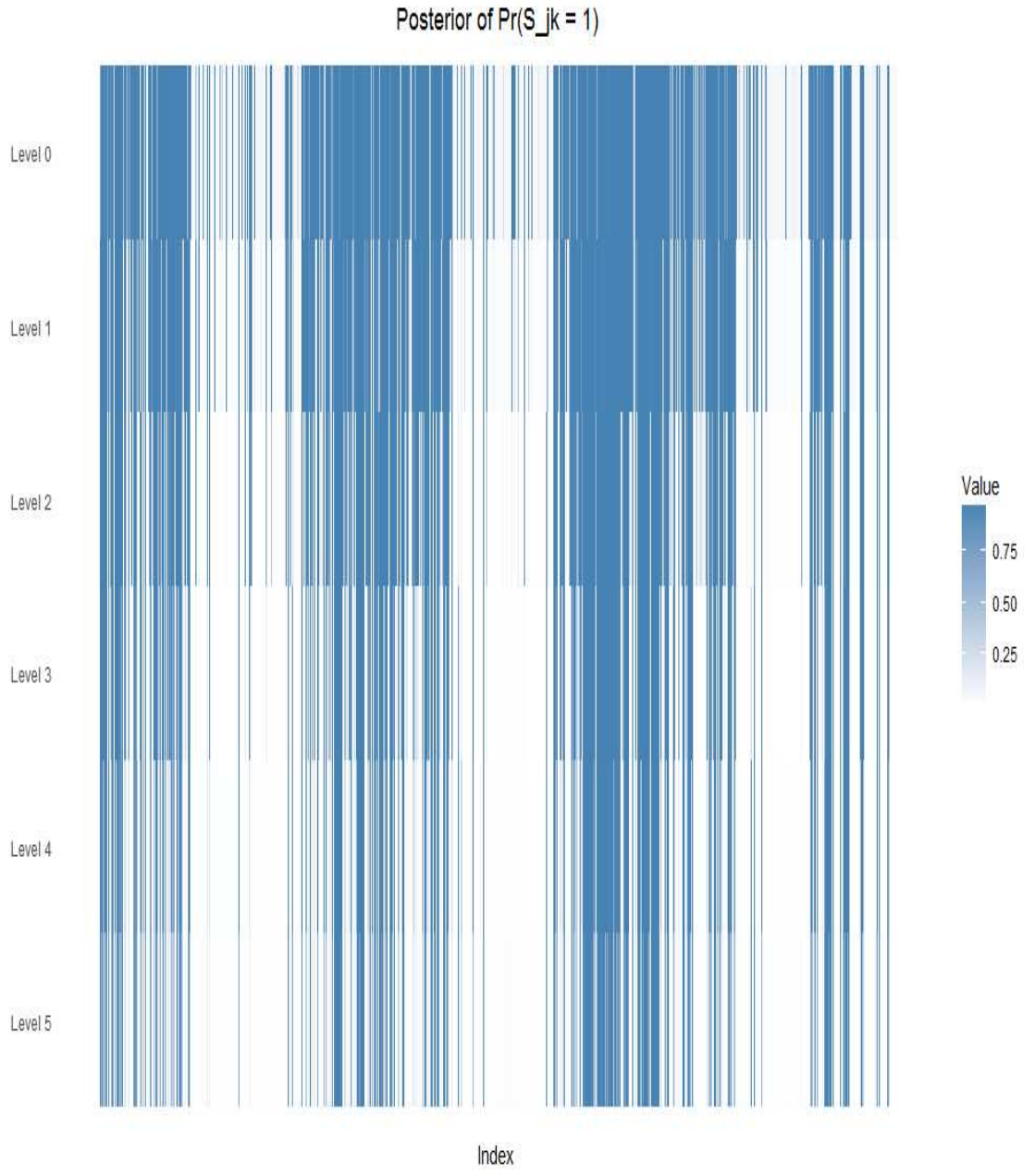


Figure 4.8: The posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ from level 0 to level 5.

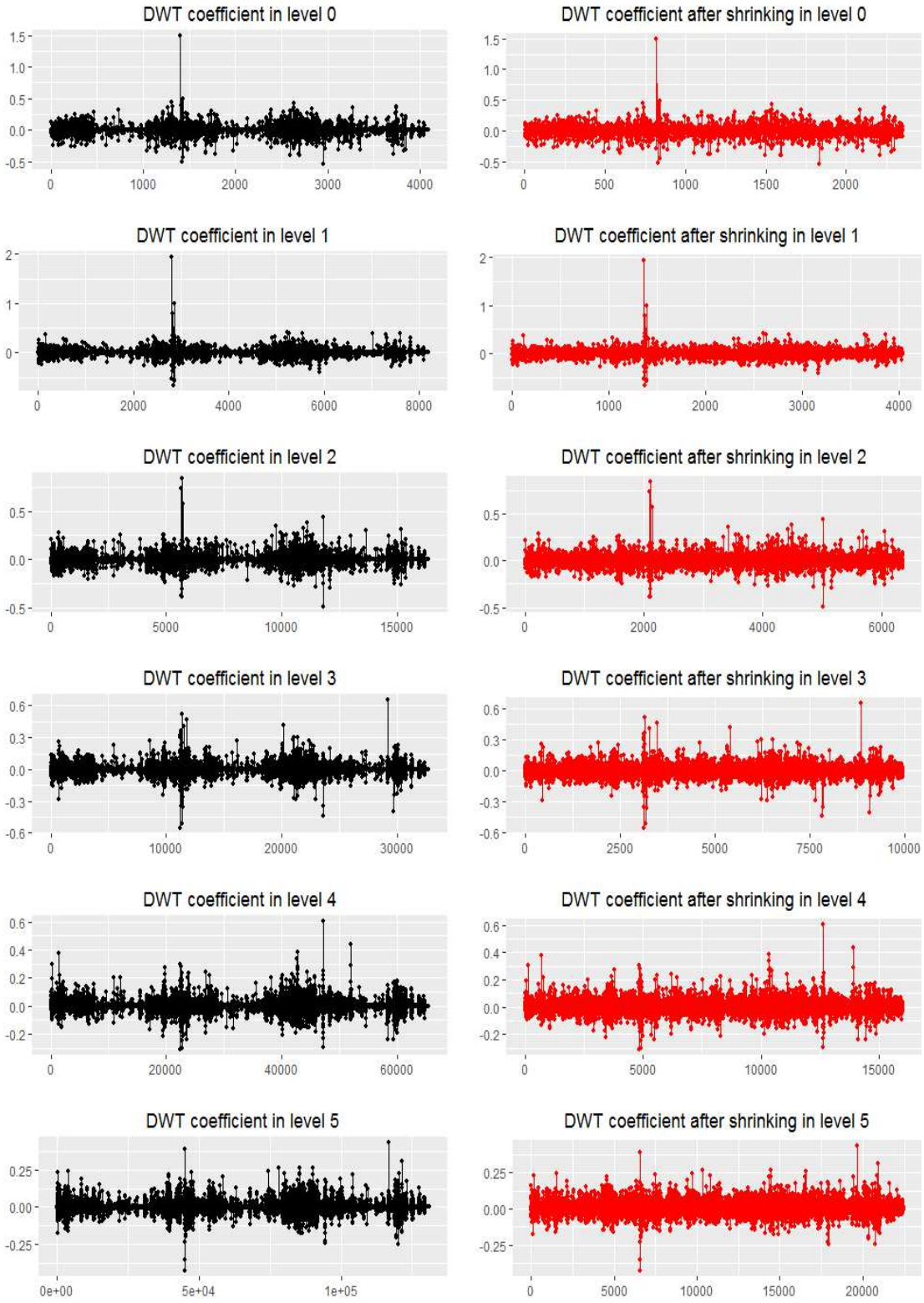


Figure 4.9: **DWT** coefficients and non-zero **DWT** coefficients after shrinking from levels 0 to 5.

Figure 4.8, which shows posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$, gives us a clear picture about which **DWT** coefficients are more useful in the **DWT** tree structure. Upon inspection of the diagram, we see two features. Firstly, for each level (or horizontal band), we see that the band is composed of two sections. One of which consisting of an area where the "density" of the vertical blue lines is noticeably higher than in the area following, that presents "active" period, and vice versa. Secondly, we also note that as the level increases (i.e., from top to bottom) the overall density of these lines generally decreases and that their colour changes to a paler shade, indicating a lower value of $Pr(S_{j,k} = 1|\mathcal{D})$.

Our first observation shows us that $Pr(S_{j,k} = 1|\mathcal{D})$ is a good indicator of the "still period" shown on the **DWT** tree structure. Our second shows us that higher decomposition levels contain more "noise" than lower ones. This is because **DWT** coefficients, at higher levels of **DWT** decomposition, correlate more closely to the high frequency domain and so more noise is generated.

From **Figure** 4.9, after shrinking the small **DWT** coefficients to zero through the **NIG-MT** model, the ones with values of zero (or close to zero) will be deleted and the rest will be retained. These latter values are shown in the right hand column. Comparing with the length of raw **DWT** coefficients before shrinking in the left column, it is surprising that raw **DWT** coefficients will be shrunk dramatically through the data reduction procedure.

Then, we check $\mathbf{ELR}_j$ and percentage of shrinking at each **DWT** decomposition level as follows:

DWT level	$Pr(S_{j,k} = 1) < P$	$\mathbf{ELR}_j$	Percentage of shrinking
0	0.9	0.0003363439	0.425293
1	0.9	0.000811384	0.5076904
2	0.9	0.002385129	0.6119385
3	0.9	0.003885711	0.6961365
4	0.9	0.00496814	0.7557831
5	0.9	0.007925812	0.8287735

Table 4.3: The threshold $P = 0.9$ with $\mathbf{ELR}_j$ and Percentage of shrinking.

From **Table** 4.3, we can see the excellent performance of **NIG-MT** models. First of all, the very small sum of values $\mathbf{ELR}_j$ when $j = 0, 1, ..., 5$ indicate the loss of information is negligible, even if we use the threshold of $P = 0.9$. Hence we can therefore use a relatively large value for the threshold. Secondly, at level 5, 4, 3, more than 70% of the **DWT** coefficients can be forced to be zero. Bear in mind, the number of **DWT** coefficients corresponds to those three levels are $\frac{n}{2}$, $\frac{n}{4}$ and $\frac{n}{8}$ respectively (n is the length of the accelerometer data). The overall percentage of zero **DWT** coefficients is 76.3%.

Comparing **Tables** 4.2 and 4.3, the energy loss rate shows that the energy loss is no more than 1% at each decomposition level in both threshold $P = 0.5$ and $P = 0.9$. The percentage of shrinking shows that the number of remaining non-zero coefficients for $P = 0.5$ will also be small when comparing the number with choosing $P = 0.9$. Hence, the approache is not sensitive to the selection of thresholds.

The data reduction method based on the **NIG-MT** model has been applied the whole of the data of all patients. The data with reduced size will be used for further analysis. The approach has at least two advantages: **(i)** it results in more efficient computing due to the small size of the data after shrinking, **(ii)** if the noise and 'still period' are removed, the data (after data reduction) can be analysed directly based on the remaining non-zero DWT coefficents with a special tree structure (see **Section** 4.1.1). Firstly, since the DWT coefficients in different decomposition levels present the information among the different frequency domain, the non-zero DWT coefficients are easier to analyse than the DWT coefficients with noise and 'still periods'. Secondly, instead of using the summary statistics features in **Chapter** 3, we analyse the remaining non-zero DWT coefficients in the DWT tree structure directly, as this may provide more information and to further analyse the 'details' of the **DWT** coefficients.

Compared to the raw data, this reduces the size of 3/4 of the raw **DWT** coefficients. If the remaining 1/4 non-zero **DWT** coefficients give the same performance through the LMGP model as the raw **DWT** coefficients in **Chapter** 3, it means that the capability of the data reduction is resultful. We will reanalyse the accelerometer data by using these new features and the model discussed in **Chapter** 3 with the reduced data set.

4.5.1 Feature re-extraction based on non-zero DWT coefficients

Non-zero **DWT** coefficients are detected through the data reduction by using **NIG-MT** model for all patients' data. Non-zero **DWT** coefficients, from all patients, will be analysed using the methods discussed in **Chapter** 3. We calculate the ratios again by using non-zero **DWT** coefficients after data reduction. The definition of the 2 new features is as follows:

$$\mathbf{SAD}_j^* = \frac{\left\|\mathbf{W}_j^*\right\|_1}{N/2^j} = 2^j \frac{\left\|\mathbf{W}_j^*\right\|_1}{N},$$

$$\mathbf{SAD}_j^{**} = \frac{\left\|\mathbf{W}_j^{**}\right\|_1}{N_j^{**}} = \frac{\left\|\mathbf{W}_j^{**}\right\|_1}{N_j^{**}}.$$

The term $\mathbf{W}_j^*$ represents the **DWT** coefficients in the j-th scale after giving the small coefficients a value of zero. In this case, though, the zero values are not deleted and so $\mathbf{W}_j^*$ has the same length as $\mathbf{W}_j$ (see also **Section** 3.2.1 of **Chapter** 3). In contrast, $\mathbf{W}_j^{**}$ represents the **DWT** coefficients in the j-th scale after forcing small coefficients to 0 and

deleting them. In other words, the length of $\mathbf{W}_j^{**}$ in different **DWT** decomposition scales only contains non-zero **DWT** coefficients, which are denoted by N_j^{**}. More specifically, we aim to find the different features from data reduction. Despite $\left\|\mathbf{W}_j^{*}\right\|_1 = \left\|\mathbf{W}_j^{**}\right\|_1$, the terms of N_j^{*} and N_j^{**} are quite different. Hence, the value of $\mathbf{SAD}_j^{*}$ and $\mathbf{SAD}_j^{**}$ will be different since $\mathbf{SAD}_j^{**}$ only contains non-zero **DWT** coefficients.

Similar to the definition of wavelet features, the definition of four types of new features is given as follows:

$$\mathbf{PNP_k^1}(SAD^*) = \frac{\mathbf{SAD}_k^{*p}}{\mathbf{SAD}_k^{*np}},$$
$$\mathbf{PNP_k^2}(SAD^*) = \frac{\mathbf{SAD}_k^{*np} - \mathbf{SAD}_k^{*p}}{\mathbf{SAD}_k^{*np} + \mathbf{SAD}_k^{*p}};$$

and

$$\mathbf{PNP_k^1}(SAD^{**}) = \frac{\mathbf{SAD}_k^{**p}}{\mathbf{SAD}_k^{**np}},$$
$$\mathbf{PNP_k^2}(SAD^{**}) = \frac{\mathbf{SAD}_k^{**np} - \mathbf{SAD}_k^{**p}}{\mathbf{SAD}_k^{**np} + \mathbf{SAD}_k^{**p}}.$$

For the different threshold values of P from 0.2 to 0.9, we calculate those features using the reduced data set of non-zero **DWT** coefficients. The correlation between those features and the response variable, the CAHAI scores, based on the threshold values P is reported in **Tables** 4.4 to 4.7 (see **Appendix** 4.7).

Comparing with the correlation between the **PNP** and clinical assessed CAHAI score after data reduction (by using all acute patients), **NIG-MT** does not reduce the correlation with the response variable compared to the ones calculated from the full data set. It implies, again, that despite shrinking the 3/4 **DWT** coefficients with 'noise' and 'still' period, the new features from the remaining 1/4 non-zero **DWT** coefficients have the almost same correlation with the clinical assessed CAHAI score. Hence, the capability of data reduction is resultful.

We use the longitudinal data analysis model discussed in **Chapter** 3 to predict the clinical assessed CAHAI score with the features calculated from the reduced data. We also calculated the RMSEs using leave-one-patient-out cross validation for acute patients. The results are shown in **Figure** 4.10.

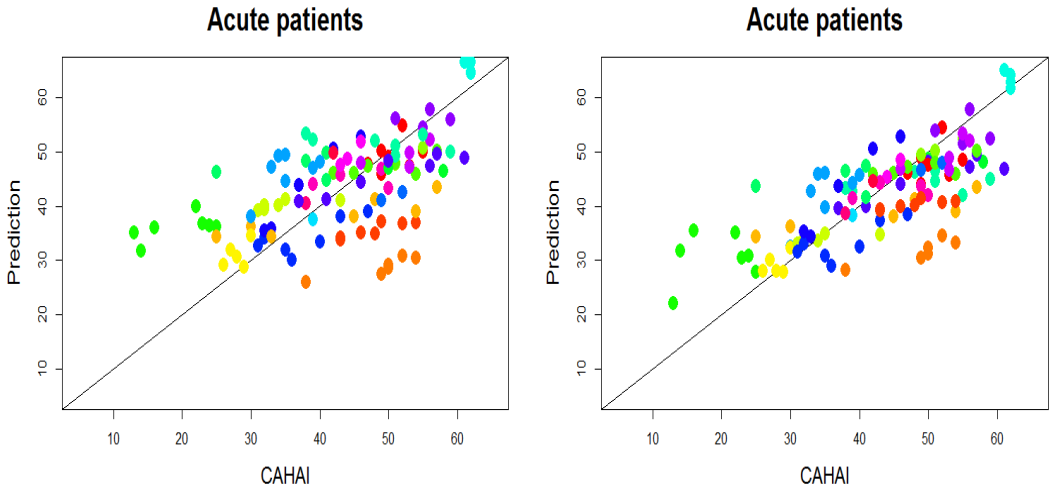


Figure 4.10: Prediction vs clinical assessed CAHAI by using leave-one-patient-out cross-validation (acute patients) based on the non-zero features. Left panel: RMSE without GP is 7.028202. Right panel with GP is 5.473805. Note that the different colours represent the different patients. It shows that the errors based on features from non-zero **DWT** coefficients are very similar to those from raw **DWT** coefficients. In other words, only the 1/4 non-zero **DWT** coefficients have the same performance as the features in **Chapter** 3. This implies that the procedure of NIG-MT is useful for data reduction.

By comparison with the RMSE from acute patients in **Chapter** 3, we see that the results of RMSE from the reduced data are almost identical with that from the whole data. This means that the re-extraction of features after data reduction keeps the same level of information as the previous features. Recall that the size of the reduced data is only about a quarter of the full data (of **DWT** coefficients), resulting after removing the "noise" and "still period". It will be easier to look for the "details" from the non-zero **DWT** coefficients on the special tree structure without 'noise' and 'still' period with the small data set.

4.6 Conclusion

The wavelet method is a very valuable tool for dealing with the signal (e.g., accelerometer data). **DWT** coefficients are another form of the original process data from different frequency domain. However, much useless information, corresponding to the "noise" part and "still" part with the original process data, are present in **DWT** coefficients. We therefore propose to use **NIG-MT** to **DWT** coefficients to do the data reduction, based on the energy loss rate ELR_j and the recommendation (see **Section** 4.5) of how to choose the threshold P in practice for other types of data. The large raw **DWT** coefficient tree structure will be shrunk and transferred to 'clean' one, making it easier to focus on the

specific part in the whole tree strucutre. For our data, about three quarters of **DWT** coefficients can be removed with less than 1% information loss at each decomposition level. This is a good data pre-processing approach, after removing the 'noise' and 'still' period. It is beneficial for us to carry out an analysis on the small size data in both the time and frequency-domains which gives the same performance with the raw processing data. In addition, the covariance tree structure of **DWT** coefficients is estimated in **Chapter** 5. This will be used to conduct multivariate functional principal component analysis **fPCA** in the next chapter. Then two approaches in this **Chapter** and **Chapter** 5 will be combined and give final delicate analysis in **Chapter** 6.

4.7 Appendix

Sensitivity of the posterior for the initial values of the hyperparameters in MMLE by using R package nlminb

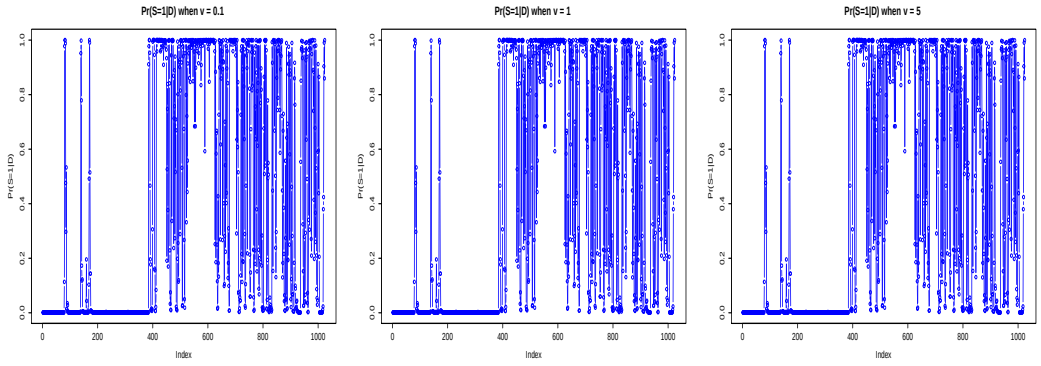


Figure 4.11: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $v = 0.1, 1, 5$ at level 5.

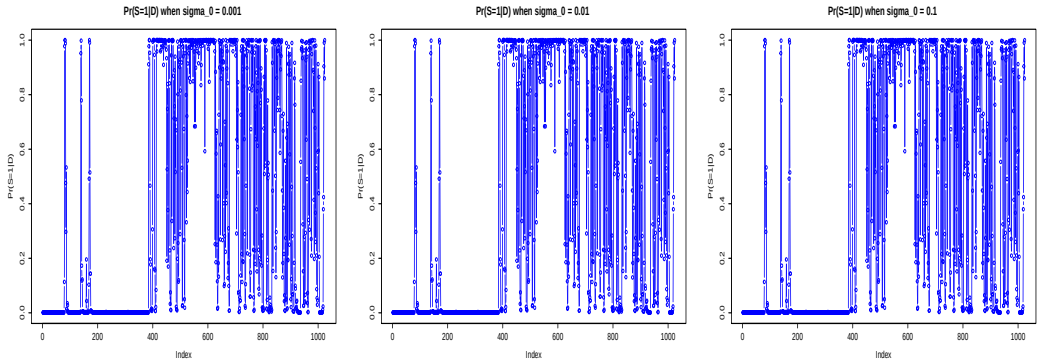


Figure 4.12: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\sigma_0 = 0.001, 0.01, 0.1$ at level 5.

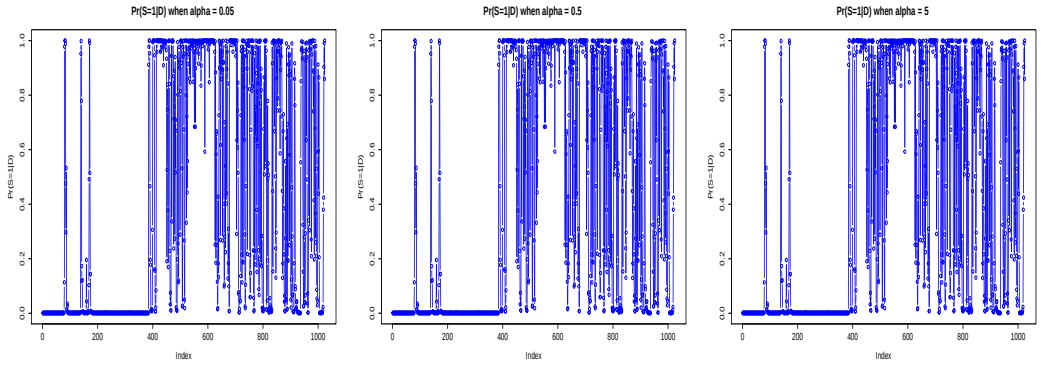


Figure 4.13: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\alpha = 0.05, 0.5, 5$ at level 5.

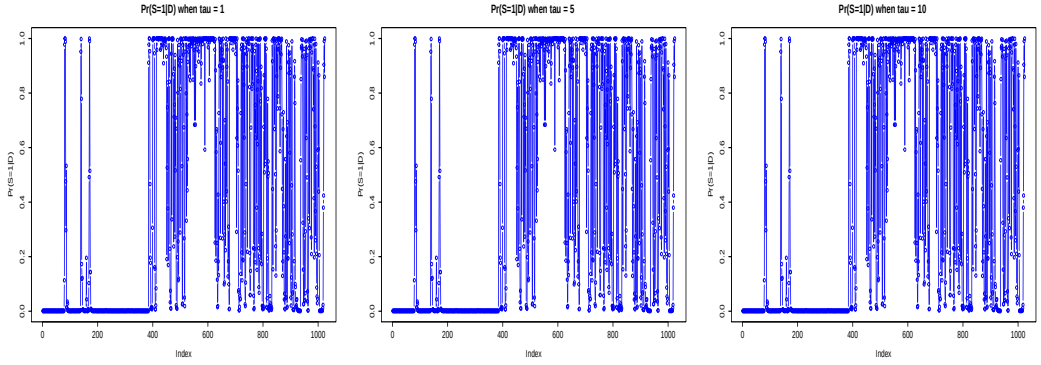


Figure 4.14: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\tau = 1, 5, 10$ at level 5.

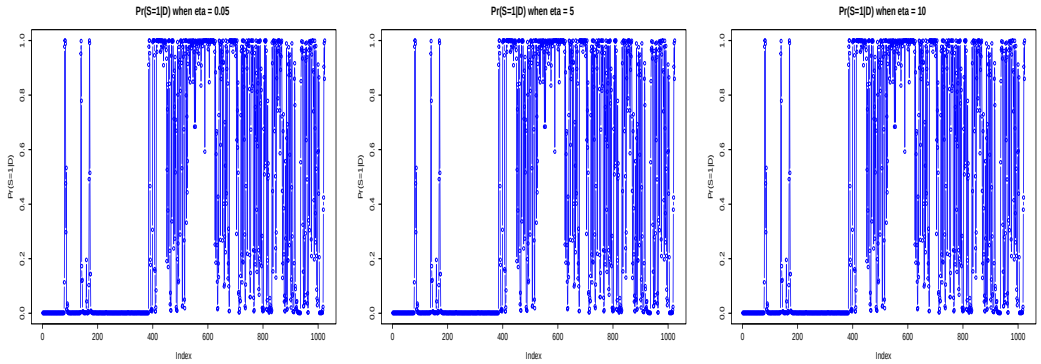


Figure 4.15: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\eta = 0.05, 5, 10$ at level 5.

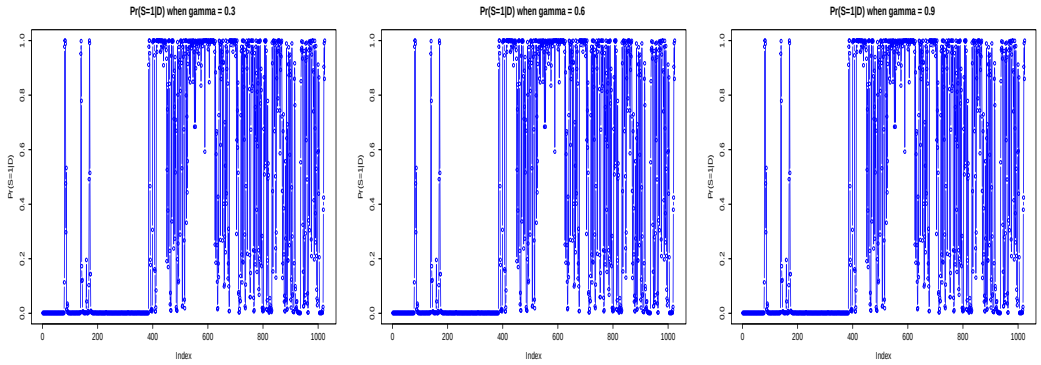


Figure 4.16: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\gamma = 0.3, 0.6, 0.9$ at level 5.

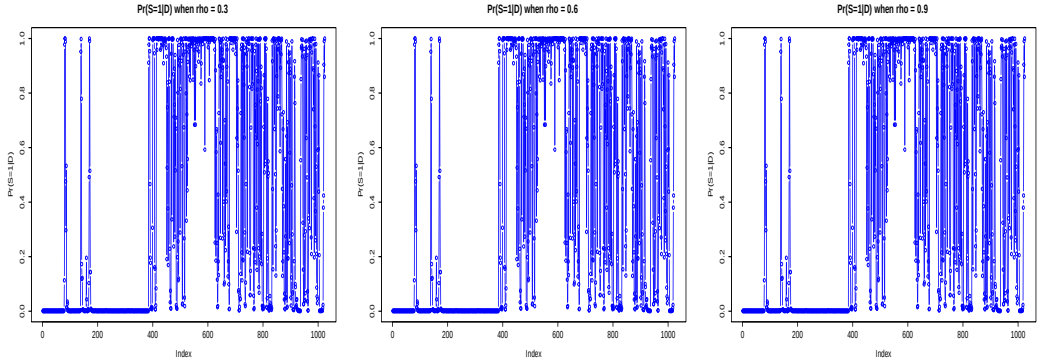


Figure 4.17: Posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ when initial $\rho_{0,k}(1) = 0.3, 0.6, 0.9$ at level 5.

The correlation between the 4 new features with CAHAI based on the threshold values P

From **Tables** 4.4 to 4.7, the last column named “Raw" is the correlation between CAHAI score and features calculated from the full data set of **DWT** coefficients. The remaining columns are the correlation between clinical assessed CAHAI score and the features through the data reduction based on the threshold $P = 0.2, 0.3, ..., 0.9$.

Threshold	20%	30%	40%	50%	60%	70%	80%	90%	Raw
Level 5 (Scale 1)	0.681	0.679	0.669	0.653	0.637	0.649	0.634	0.601	0.696
Level 4 (Scale 2)	0.69	0.686	0.683	0.681	0.678	0.672	0.663	0.644	0.694
Level 3 (Scale 3)	0.68	0.68	0.678	0.676	0.674	0.669	0.664	0.655	0.667
Level 2 (Scale 4)	0.629	0.628	0.627	0.625	0.624	0.622	0.619	0.615	0.605
Level 1 (Scale 5)	0.509	0.511	0.512	0.509	0.509	0.508	0.508	0.507	0.489
Level 0 (Scale 6)	0.395	0.391	0.394	0.397	0.4	0.402	0.408	0.415	0.349

Table 4.4: The correlation between $\mathbf{PNP^1_k}(SAD^*)$ with CAHAI.

Threshold	20%	30%	40%	50%	60%	70%	80%	90%	Raw
Level 5 (Scale 1)	0.693	0.694	0.695	0.696	0.697	0.698	0.698	0.691	0.696
Level 4 (Scale 2)	0.687	0.688	0.689	0.689	0.69	0.691	0.692	0.694	0.694
Level 3 (Scale 3)	0.663	0.663	0.664	0.664	0.665	0.665	0.666	0.667	0.667
Level 2 (Scale 4)	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.605
Level 1 (Scale 5)	0.483	0.483	0.482	0.482	0.482	0.481	0.481	0.48	0.489
Level 0 (Scale 6)	0.344	0.343	0.342	0.341	0.34	0.339	0.338	0.334	0.349

Table 4.5: The correlation between $\mathbf{PNP^1_k}(SAD^{**})$ with CAHAI.

Threshold	20%	30%	40%	50%	60%	70%	80%	90%	Raw
Level 5 (Scale 1)	-0.71	-0.706	-0.694	-0.676	-0.659	-0.67	-0.652	-0.611	-0.715
Level 4 (Scale 2)	-0.712	-0.707	-0.704	-0.702	-0.697	-0.69	-0.68	-0.659	-0.709
Level 3 (Scale 3)	-0.699	-0.698	-0.697	-0.694	-0.692	-0.687	-0.683	-0.673	-0.683
Level 2 (Scale 4)	-0.647	-0.647	-0.647	-0.646	-0.645	-0.644	-0.641	-0.637	-0.626
Level 1 (Scale 5)	-0.537	-0.539	-0.541	-0.538	-0.539	-0.538	-0.538	-0.536	-0.516
Level 0 (Scale 6)	-0.43	-0.426	-0.427	-0.43	-0.431	-0.433	-0.438	-0.444	-0.375

Table 4.6: The correlation between $\mathbf{PNP^2_k}(SAD^*)$ with CAHAI.

Threshold	20%	30%	40%	50%	60%	70%	80%	90%	Raw
level 5(Scale 1)	-0.712	-0.714	-0.715	-0.716	-0.717	-0.718	-0.72	-0.717	-0.715
level 4(Scale 2)	-0.702	-0.703	-0.703	-0.704	-0.705	-0.706	-0.707	-0.709	-0.709
level 3(Scale 3)	-0.678	-0.679	-0.679	-0.679	-0.68	-0.68	-0.681	-0.683	-0.683
level 2(Scale 4)	-0.621	-0.621	-0.621	-0.621	-0.621	-0.621	-0.621	-0.622	-0.626
level 1(Scale 5)	-0.511	-0.51	-0.51	-0.51	-0.509	-0.509	-0.509	-0.508	-0.516
level 0(Scale 6)	-0.37	-0.369	-0.368	-0.367	-0.366	-0.365	-0.363	-0.36	-0.375

Table 4.7: The correlation between $\mathbf{PNP^2_k}(SAD^{**})$ with CAHAI.

Chapter 5

Classification based on a slide window approach and multivariate fPCA

In this chapter, from the data reduction discussed in **Chapter** 4, we will investigate the tree structure through all the **DWT** coefficients. Although about three quarters of the **DWT** coefficients can be forced to 0 and removed, the size of the data set is still very large. We will use the slide window approach and multivariate functional principal component analysis (**fPCA**)(Ramsay & Silverman (2005)) to further extract the useful information from the pattern of small **DWT** tree structures and also reduce the data size. In this chapter we use the methods of slide window and multivariate **fPCA** with designed data at first by using the refined information from small **DWT** tree structures. We will then apply, in the next chapter, these techniques together with **NIG-MT** to extract the useful pattern's information from free-living data for stroke rehabilitation assessment. As we discussed before, there is a tree structure in the **DWT** coefficients; the parent node at level j has two children nodes at level $j+1$ and the value of two children nodes depends on the value of parent node, which means each node at the top decomposition level 0, and its children nodes at the following decomposition levels, can be treated as a tree structure. The correlation in those **DWT** coefficients depends on the tree structure. We will first calculate the covariance matrix based on the tree structure before we conduct a multivariate **fPCA**.

The designed data (walking, running and opening a jar) is collected from the wrist-worn sensor AX3 (see **Section** 5.1). In this chapter, we investigate the small **DWT** coefficients tree structure, from the designed data, in order to extract the pattern's information from three movements through the multivariate **fPCA**. This is then used to classify the three movements through different classifcation methods (**Section** 5.4).

5.1 About the design data

For three different individuals, the design data were collected for both left and right hands with acceleration being measured along each axis. The data were collected for three activities: walking, running and opening jars. The duration of the data collection was one minute and the sampling frequency was 100 Hz, i.e., 100 observations were recorded per second. The data includes a_x,a_y,a_z and a time-stamp, where a_x,a_y,a_z are the accelerations along each axis measured in g units ($1\text{g} = 9.8m/s^2$). We then combine 3 coordinate axes (X,Y,Z) to get the signal vector magnitude (raw **VM data**, not the 1 second-wise **VM** data) (**Section** 2.2.3). From the design data (left and right hand from 1 person) for the three activities are shown in **Figures** 5.1, 5.2 and 5.3 as follows:

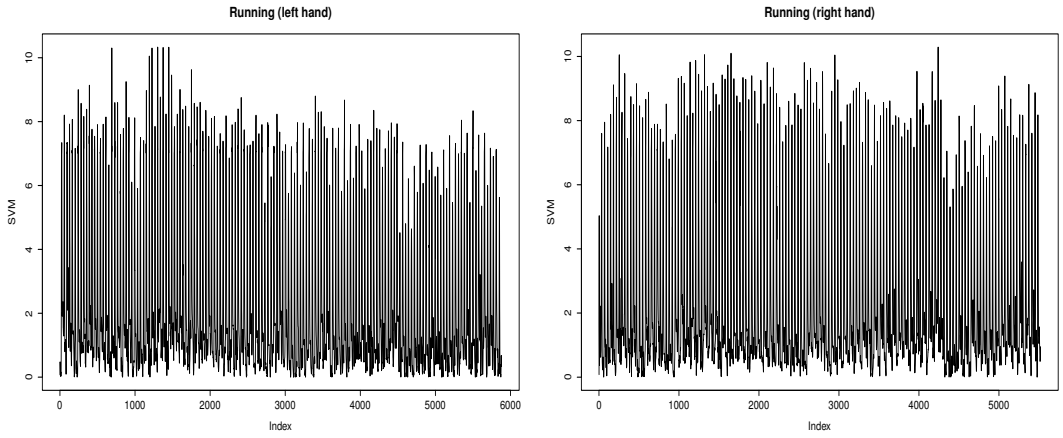


Figure 5.1: The data for running activity. Left panel: Left hand. Right panel: Right hand. Note that the Index refers to the observations within the designed data in 1 minute.

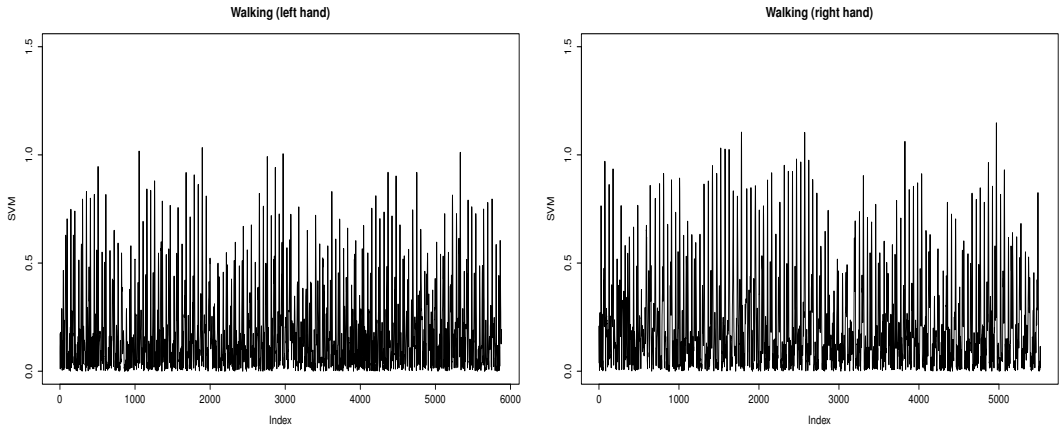


Figure 5.2: The data for walking activity. Left panel: Left hand. Right panel: Right hand. Note that the Index refers to the observations within the designed data in 1 minute.

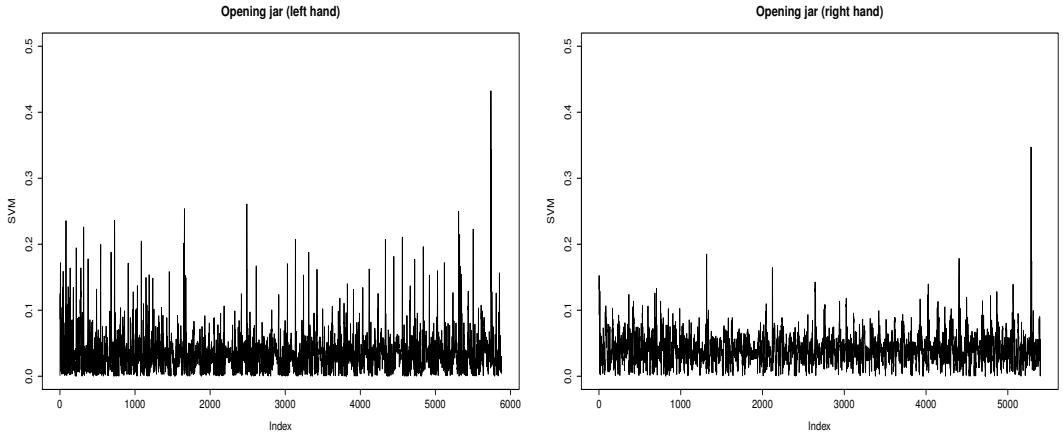


Figure 5.3: The data for opening jar activity. Left panel: Left hand. Right panel: Right hand. Note that the Index refers to the observations within the designed data in 1 minute.

If we look very carefully at the bottom of each figure, all three movements contain small values (close to 0), especially in the activities of walking and opening jars. This implies that these three movements include similar properties of low frequency information. Moreover, it is also difficult to find the pattern through the raw data. To be more specific, despite the fact that we collect the data for 1 min, these three movements consist of their patterns. If we take "running" as an example, both hands will keep a regular pattern, it will be repeated again and again through the entire data. To capture the patterns efficiently and remove the negative impact from low frequency information properly, we transform the raw data into the wavelet domain and analyze it with a **DWT** tree structure in **Section** 5.2.

5.2 Identifying the DWT coefficient tree structure

Before we determine the small **DWT** tree structure, we recall the **DWT** coefficients based on the **Section** 2.3.3, then we apply **DWT** with total of 6 levels, the **DWT** coefficients $\mathbf{W}$ is a column vector, and $\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_J, \mathbf{V}_J]^{\mathrm{T}}$, which are decided by the decomposition level J. In our study, we mainly focus on the detailed coefficients $\mathbf{W}_j$ at decomposition level j which represents the differences in adjacent weighted averages from scale 1 to scale J, $J = 6$.

After applying the **DWT** to data with length N, the detailed coefficients $\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_J$ will be of length $\frac{N}{2^j}$ at each decomposition level j. In other words, the N length data transforms to a total of J sets detailed coefficients with length of $\frac{N}{2^1}, \frac{N}{2^2}, ..., \frac{N}{2^J}$ after applying **DWT**. N is usually very large in the patients' data, so the total number of **DWT** coefficients is also very large. However, most of **DWT** coefficients are of no use and can be removed through the data reduction based on the discussion given in **Chapter** 4. Note

that in this chapter we do not need to consider the data reduction (through the **NIG-MT**) since the designed data is not too large size in the time-domain and is only designed for certain movements. We will find the special tree structure in the next subsection and calculate the covariance matrix. The dimension of **DWT** coefficients will further be reduced by using multivariate **fPCA** in each slide window (**Section** 5.3.3).

5.2.1 Patterns within the three movements

Reconsidering the designed data with the three movements (walking, running and opening jars), as we discussed in **Section** 5.1, and based on the **DWT** coefficients from the designed data, we recall that each of them has its own pattern through the time and frequency-domains. We aim to look for the special pattern from the whole **DWT** coefficients with these three movements. To obtain the pattern properly, we first separate the designed data into different slide windows, each of which includes 128 points covering a period of 1.28 seconds. Here, we take left hand's designed data for three movements in one individual as an example and, after separating the data into slide windows, the slide windows' data can be shown in **Figure** 5.4.

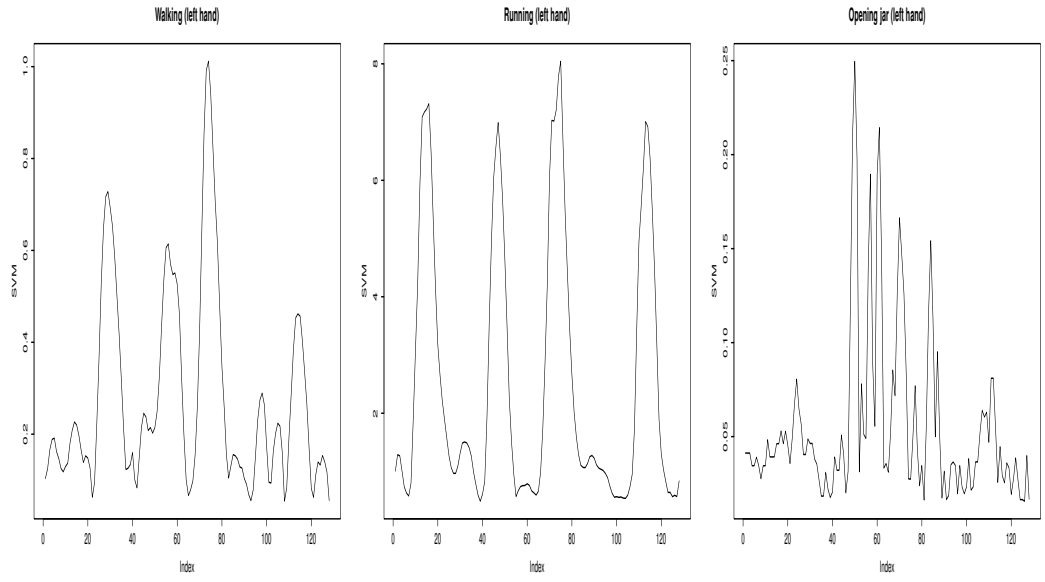


Figure 5.4: The raw data of left hand in one slide window. Left panel: Walking, Middle panel: Running, Right panel: Opening jars. Note that the Index refers to the observations within the designed data in slide window with 1.28 second.

Instead of obtaining the whole tree structure of **DWT** coefficients from the entire designed data, the **DWT** coefficients from each slide window can be treated as a "small" tree, which corresponds to specific patterns in whole **DWT** coefficients. To obtain the small **DWT** tree structure properly, we first select 128 points in each slide window, then

the results of using the length of 64 and 192 points in each slide window will be analyzed at the end of this chapter (see **Section** 5.4.2) for comparision. Bear in mind, through, that our final goal is to give an analysis for the stroke patients' accelerometer data (free-living data). If the information extracted from patterns in the slide window from designed data can distinguish these three movements and it performs well, this means that we may apply it to the stroke patients' data as well.

We assume that the **DWT** decomposition level $J = 6$ contains enough low-frequency component for the three movements of the designed data. After applying the **DWT** in each slide window with decomposition level $J = 6$, the data can be transferred to the detailed coefficients $\mathbf{W}_j$, which has a **DWT** tree structure; the **DWT** tree structure in one slide window is given in **Figure** 5.5. The special frequency domain and the length of each level of tree are given in **Table** 5.1.

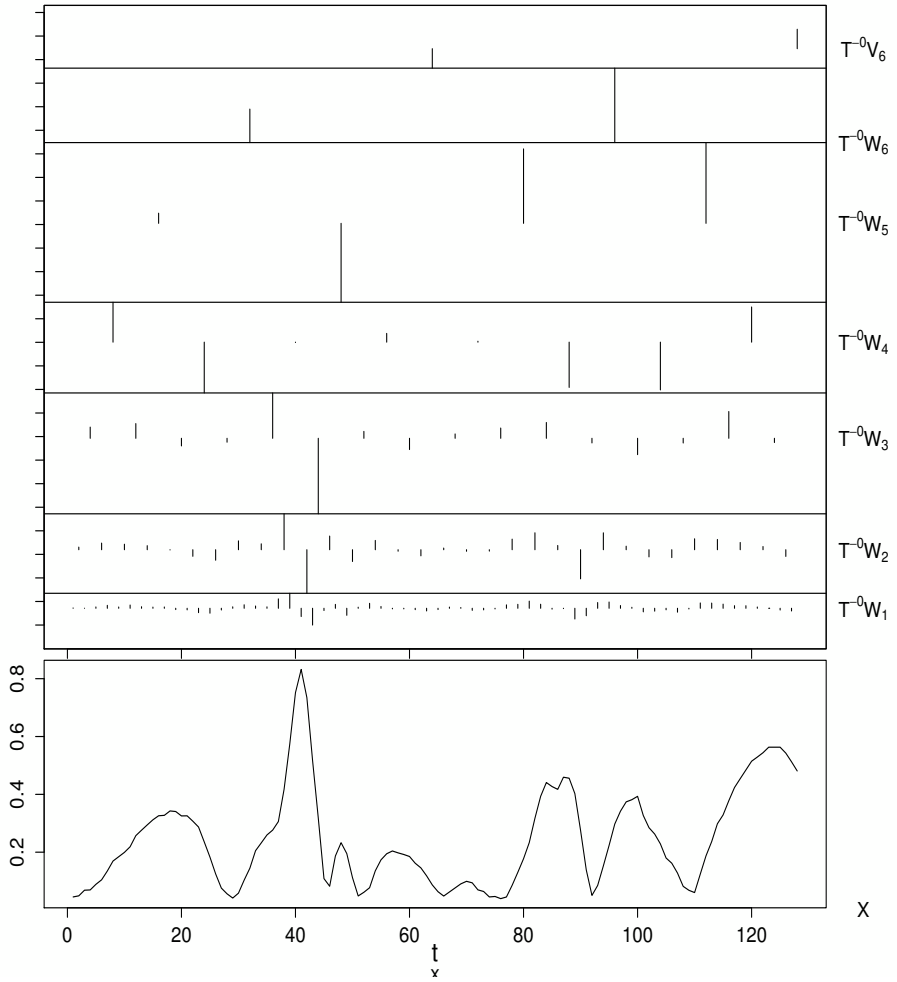


Figure 5.5: **DWT** tree structure in one slide window. Note that this plot uses the function **plot.dwt** from the R package **wavelets**.

	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$
Scale	Scale 1 (Level 5)	Scale 2 (Level 4)	Scale 3 (Level 3)
Frequency	50hz - 100hz	25hz - 50hz	12.5hz - 25hz
Length	64	32	16
	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$
Scale	Scale 4 (Level 2)	Scale 5 (Level 1)	Scale 6 (Level 0)
Frequency	6.25hz - 12.5hz	3.125hz - 6.25hz	1.5625hz - 3.125hz
Length	8	4	2

Table 5.1: The details of **DWT** tree structure in one slide window.

The detailed coefficients $\mathbf{W}_4, \mathbf{W}_3$, and $\mathbf{W}_2$ are from **DWT** decomposition levels 2 to 4, which contains the frequency information at 6.25hz - 12.5hz, 12.5hz - 25hz and 25hz - 50hz respectively (See **Table**: 5.1). As we know, we transform the designed data to the **DWT** coefficients in the wavelet domain by using **DWT** at 6 levels, based on our designed data, i.e., the detailed coefficients at level 5 is 50-100Hz. We can treat it as "noise", especially for the movements of walking and opening jars. The remaining 5 levels, $\mathbf{W}_6, \mathbf{W}_5, \mathbf{W}_4, \mathbf{W}_3$, and $\mathbf{W}_2$, present the most information for those three activities.

5.3 Extracting the information from the pattern of small DWT tree structure in each slide window

As we discussed before, after we obtain the tree structure of **DWT** coefficients in slide window, (see **Table** 5.1 and **Figure** 5.5), the raw designed data can be separated into different slide windows and we need to find the different patterns in these slide windows. However, if we select 128 points from the designed data as the length of each slide window, the pattern in it for three movements contains in total 62 **DWT** coefficients. There are 2, 4, 8, 16 and 32 **DWT** coefficients at level 0, 1, 2, 3 and 4 respectively. From the tree structure in the slide window (**Figure** 5.5), we see that $\mathbf{W}_6$ and $\mathbf{W}_5$ have 2 and 4 points at level 0 and level 1 respectively. This will make analysis much easier and we will use $\mathbf{W}_6$ and $\mathbf{W}_5$ as potential candidate covariates in the models we will discuss later in **Section** 5.3.4. However, $\mathbf{W}_4$ $\mathbf{W}_3$ and $\mathbf{W}_2$ have 8, 16 and 32 points (totally 56 points) at the level 2, level 3 and level 4, and this makes direct analysis more difficult. An effective way is to transform these detailed coefficients (level 2, level 3 and level 4) to lower-dimensional data by using the **fPCA** via the special covariance structure in **Section** 5.3.1.

5.3.1 Covariance matrix in small DWT tree structure at decomposition level 2, 3 and 4

Our first challenge involves combining the **DWT** coefficients in the **DWT** structures from level 2 to level 4. Recalling the designed data, walking, running and opening jars, three **DWT** tree structures can be obtained for each activity. The relationship from point by point at each level of **DWT** tree structure should be measured by the covariance matrix. To be more specific, for each movement, the **DWT** coefficients in the small tree structure at level 2 can be expressed as:

$$\mathbf{D}_2^j = (D_{2_1}^j, D_{2_2}^j, ..., D_{2_8}^j), \tag{5.1}$$

where $D_{2_i}^j$ is the i-th **DWT** coefficient at decomposition level 2 for $i = 1, 2, ..., 8$. j stands for the j-th individual for $j = 1, 2, ..., m$.

Similarly, **DWT** coefficients in the tree structure at level 3 and 4 can be expressed as:

$$\mathbf{D}_3^j = (D_{3_1}^j, D_{3_2}^j, ..., D_{3_{16}}^j), \tag{5.2}$$

and

$$\mathbf{D}_4^j = (D_{4_1}^j, D_{4_2}^j, ..., D_{4_{32}}^j). \tag{5.3}$$

Using the vector and matrix notations, **Equations** (5.1), (5.2) and (5.3) can be rewritten as:

$$\left[\begin{array}{cccc} \mathbf{D_{2_1}} & \mathbf{D_{2_2}} & \cdots & \mathbf{D_{2_8}} \end{array} \right], \tag{5.4}$$

$$\left[\begin{array}{cccc} \mathbf{D_{3_1}} & \mathbf{D_{3_2}} & \cdots & \mathbf{D_{3_{16}}} \end{array} \right], \tag{5.5}$$

$$\left[\begin{array}{cccc} \mathbf{D_{4_1}} & \mathbf{D_{4_2}} & \cdots & \mathbf{D_{4_{32}}} \end{array} \right], \tag{5.6}$$

where each element in **Equations** (5.4), (5.5) and (5.6) is m-dimension vector, including the related **DWT** coefficients for m-th individual in a slide window.

We can therefore calculate the covariance matrix of **DWT** coefficients at each level. At level 2, the 8×8 covariance matrix can be calculated by:

$$\mathbf{C}_2 = \left[\begin{array}{cccc} \mathrm{cov}\,(\mathbf{D_{2_1}}, \mathbf{D_{2_1}}) & \mathrm{cov}\,(\mathbf{D_{2_1}}, \mathbf{D_{2_2}}) & \cdots & \mathrm{cov}\,(\mathbf{D_{2_1}}, \mathbf{D_{2_8}}) \\ \mathrm{cov}\,(\mathbf{D_{2_2}}, \mathbf{D_{2_1}}) & \mathrm{cov}\,(\mathbf{D_{2_2}}, \mathbf{D_{2_2}}) & \cdots & \mathrm{cov}\,(\mathbf{D_{2_2}}, \mathbf{D_{2_8}}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cov}\,(\mathbf{D_{2_8}}, \mathbf{D_{2_1}}) & \mathrm{cov}\,(\mathbf{D_{2_8}}, \mathbf{D_{2_2}}) & \cdots & \mathrm{cov}\,(\mathbf{D_{2_8}}, \mathbf{D_{2_8}}) \end{array} \right], \tag{5.7}$$

where $\mathrm{cov}\,(\mathbf{D_{2_i}}, \mathbf{D_{2_k}})$ is the sample covariance between $\mathbf{D_{2_i}}$ and $\mathbf{D_{2_k}}$ for $(i, k) = 1, 2, ..., n$, $n = 8$ at level 2; $\mathbf{D_{2_i}}$ and $\mathbf{D_{2_k}}$ includes the data from m subjects.

Figure 5.6 presents the heat maps of the covariance matrix using the data collected from three subjects, and we can see the clear difference between those three activities.

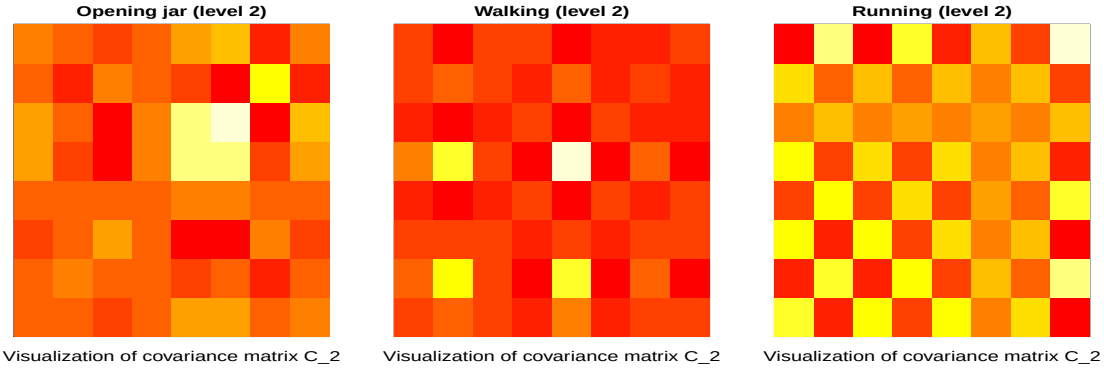


Figure 5.6: The visualization of covariance matrix $\mathbf{C}_2$ with three movements at level 2. Left to Right: Opening jars, Walking and Running.

Similarly, we can calculate the covariance matrix at level 3 and 4 with dimensions 16×16 and 32×32 respectively:

$$\mathbf{C}_3 = \begin{bmatrix} \operatorname{cov}(\mathbf{D_{3_1}}, \mathbf{D_{3_1}}) & \operatorname{cov}(\mathbf{D_{3_1}}, \mathbf{D_{3_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{3_1}}, \mathbf{D_{3_{16}}}) \\ \operatorname{cov}(\mathbf{D_{3_2}}, \mathbf{D_{3_1}}) & \operatorname{cov}(\mathbf{D_{3_2}}, \mathbf{D_{3_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{3_2}}, \mathbf{D_{3_{16}}}) \\ \vdots & \vdots & \vdots & \vdots \\ \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{3_1}}) & \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{3_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{3_{16}}}) \end{bmatrix}, \tag{5.8}$$

and

$$\mathbf{C}_4 = \begin{bmatrix} \operatorname{cov}(\mathbf{D_{4_1}}, \mathbf{D_{4_1}}) & \operatorname{cov}(\mathbf{D_{4_1}}, \mathbf{D_{4_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_1}}, \mathbf{D_{4_{32}}}) \\ \operatorname{cov}(\mathbf{D_{4_2}}, \mathbf{D_{4_1}}) & \operatorname{cov}(\mathbf{D_{4_2}}, \mathbf{D_{4_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_2}}, \mathbf{D_{4_{32}}}) \\ \vdots & \vdots & \vdots & \vdots \\ \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{4_1}}) & \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{4_2}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{4_{32}}}) \end{bmatrix}. \tag{5.9}$$

The heat maps are shown in **Figures** 5.7 and 5.8 respectively.

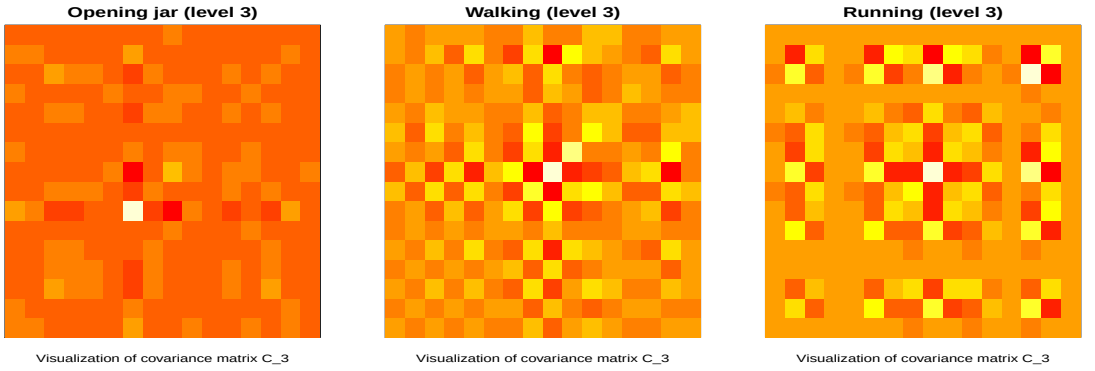


Figure 5.7: The visualization of covariance matrix $\mathbf{C}_3$ with three movements at level 3. Left to Right: Opening jars, Walking and Running.

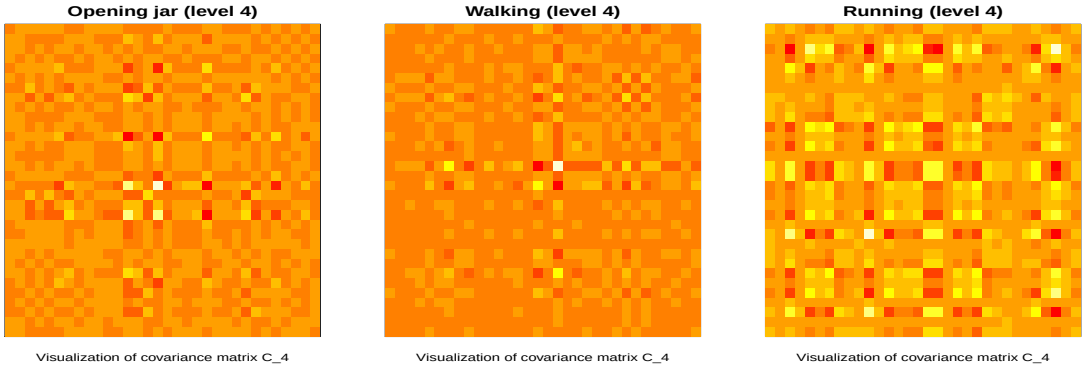


Figure 5.8: The visualization of covariance matrix $\mathbf{C}_4$ with three movements at level 4. Left to Right: Opening jars, Walking and Running.

5.3.2 Combining levels 2, 3 and 4

Until now, for each movement, the covariance matrix is obtained at each individual level, the next step is to combine the **DWT** coefficients in the **DWT** tree structure from level 2 to level 4 together. An easy way to understand this is to start by combining level 2 and level 3. After combining the **DWT** coefficients at level 2 and level 3, **Equations** 5.1 and 5.2 can be written as:

$$(\mathbf{D_2^j}, \mathbf{D_3^j}) = (D^j_{2_1}, D^j_{2_2}, ..., D^j_{2_8}, D^j_{3_1}, D^j_{3_2}, ..., D^j_{2_{16}}), \tag{5.10}$$

where j means j-th individual. Also using vector and matrix notations, **Equations** (5.10) can be rewritten as:

$$\begin{bmatrix} \mathbf{D_{2_1}} & \mathbf{D_{2_2}} & \cdots & \mathbf{D_{2_8}} & \mathbf{D_{3_1}} & \mathbf{D_{3_2}} & \cdots & \mathbf{D_{3_{16}}} \end{bmatrix}, \tag{5.11}$$

where each element in **Equations** (5.11) is m-dimension vector, including the related **DWT** coefficients for m-th individual in a slide window. The covariance matrix is calculated by:

$$\mathbf{C}_{2,3} = \begin{bmatrix} \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{2_8}}) & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{3_{16}}}) \\ \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{2_8}}) & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{3_{16}}}) \\ \vdots & & \vdots & \vdots & & \vdots \\ \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{2_8}}) & \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{3_{16}}}, \mathbf{D_{3_{16}}}) \end{bmatrix}. \tag{5.12}$$

The heat map of the covariance matrix is presented in **Figure** 5.9. It also shows the different patterns clearly for three different activities.

Figure 5.9: The visualization of covariance matrix $\mathbf{C}_{2,3}$ with three movements at combining levels 2 and 3. Left to Right: Opening jars, Walking and Running.

Similarly, from **Equations** (5.4), (5.5) and (5.6), we can combine all the 3 levels by:

$$\left[\begin{array}{cccccccc} \mathbf{D_{2_1}} & \cdots & \mathbf{D_{2_8}} & \mathbf{D_{3_1}} & \cdots & \mathbf{D_{3_{16}}} & \mathbf{D_{4_1}} & \cdots & \mathbf{D_{4_{32}}} \end{array}\right]. \tag{5.13}$$

The covariance matrix is given by:

$$\mathbf{C}_{2,3,4} = \left[\begin{array}{cccccccc} \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{4_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_1}}, \mathbf{D_{4_{32}}}) \\ \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{4_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{2_2}}, \mathbf{D_{4_{32}}}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{2_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{3_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{4_1}}) & \cdots & \operatorname{cov}(\mathbf{D_{4_{32}}}, \mathbf{D_{4_{32}}}) \end{array}\right]. \tag{5.14}$$

The heat map is shown in **Figure** 5.10, we have the similar finding to **Figure** 5.9

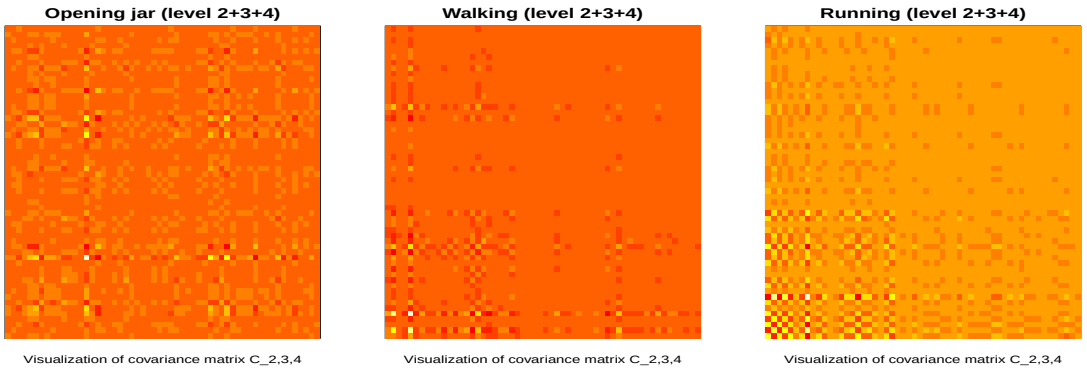


Figure 5.10: The visualization of covariance matrix $\mathbf{C}_{2,3,4}$ with three movements at combining levels 2, 3 and 4. Left to Right: Opening jars, Walking and Running.

5.3.3 Calculating PC score through multivariate fPCA

So far, the special covariance structure from combining levels 2, 3 and 4 is obtained through the **DWT** tree structure (see **Table** 5.1). **Figure** 5.10 presents the correlation of

each activity. The next step is to apply multivariate **fPCA** (Ramsay & Silverman (2005)) and find the PCs score through the covariance structure. Multivariate **fPCA** is used to reduce the dimensionality of the data by transforming the **DWT** coefficients at combining levels 2, 3 and 4 into a lower dimensional space with minimal loss of information. Moreover, Principal Components (PCs) are obtained to retain most of the variation presented at combining levels 2, 3 and 4 from the small **DWT** tree structure.

For these three movements, the PC scores from multivariate **fPCA** can be obtained from the covariance matrix (see **Equation** (5.14)); 6 PC scores (3 for first principal component and 3 for second principal component) are calculated with three movements. To be more specific, instead of analysing the **DWT** coefficients from level 2 to level 4 separately, the **DWT** coefficients at combining levels 2, 3 and 4 can be transformed to 2 PC scores. The **DWT** tree structure in each slide window in **Section** 5.2.1 **Table** 5.1 becomes:

	$\mathbf{W}_6$	$\mathbf{W}_5$	Combining levels 2, 3 and 4
Scale	Level 0 (Scale 6)	Level 1 (Scale 5)	Level 2+3+4 (Scale 4,3,2)
Frequency	1.5625hz - 3.125hz	3.125hz - 6.25hz	6.25hz - 50hz
Length	2 DWT coefficients	4 DWT coefficients	2 PC scores

Table 5.2: The special tree structure after multivariate **fPCA**.

We therefore use the 8 candidate variables in our classification model instead of using the original 62 variables. The idea can be used as a general method. When the sample size in each slide window is larger, the method is more efficient.

Until now, the most important entires in **Table** 5.2 are the PC1 score and PC2 score in combining levels 2, 3 and 4. After applying multivariate **fPCA**, in each slide window, $\mathbf{W}_4$ $\mathbf{W}_3$ and $\mathbf{W}_2$ (totally 56 points) at level 2, level 3 and level 4 are transformed to the first 2 PC scores, which account for around 99.9% of the information. PC1 accounts for about 90% of the total variance.

The PC scores (PC1 and PC2) through multivariate **fPCA** and the other variables listed in **Table** 5.2, provide enough information to distinguish between these three movements in each slide window. **Figure** 5.11 shows the PC scores of the first PC calculated for 36 slide windows for one subject, it shows clear differences with three movements. **Figure** 5.12 presents the scores of the second PC; it provides further information to classify those three movements. The results calculated from the other hand and the other two subjects are presented in **Appendix** 5.6; the results are similar.

PC scores from left hand for three individuals

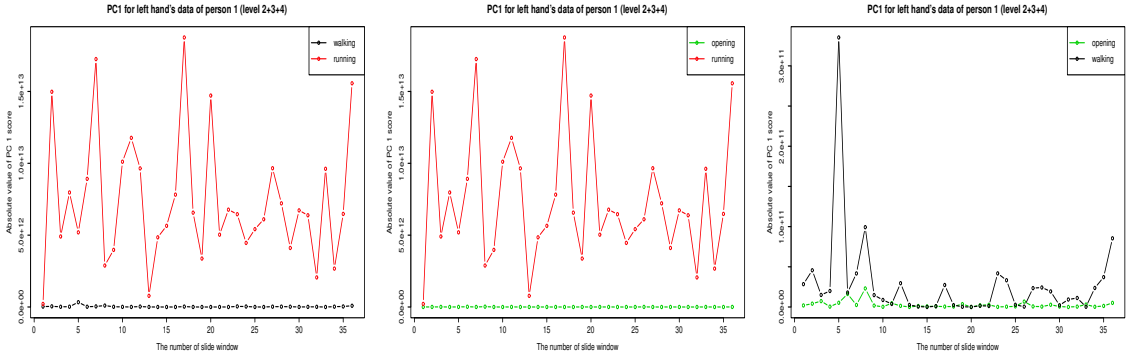


Figure 5.11: The first PC absolute score with three movements in the first person's data (left hand).

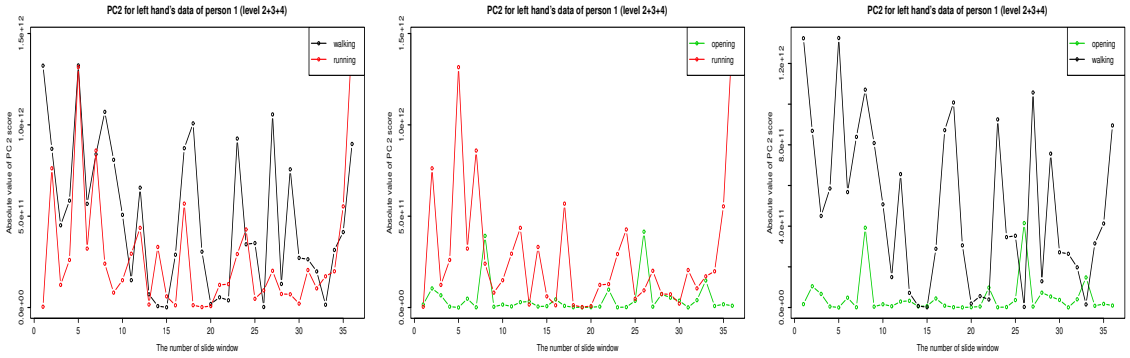


Figure 5.12: The second PC absolute score with three movements in the first person's data (left hand).

5.3.4 The potential candidate in small DWT tree structures at decomposition level 0 and 1

It can be seen from **Table** 5.2 that the labels "opening jars", "walking" and "running" correspond to the pattern of each slide window, and there are 8 covariates extracted from the pattern of each slide window. The 8 covariates in the classification model for each slide window based on the small **DWT** tree structure are given in **Table** 5.3.

	Level 0 (Scale 6)	Level 1 (Scale 5)	Level 2+3+4 (Scale 4,3 and 2)
Covariates	x_{01}, x_{02}	x_{11}, x_{12}, x_{13} and x_{14}	PC1 score and PC2 score

Table 5.3: The 8 covariates in each slide window

Since the value of wavelet coefficients will be both positive and negative, analysis is not particularly convenient. One method is to carry out the absolute transform of each

point at levels 0 and 1 (this idea is similar to the scalar featrues **SAD** in **Section** 3.2.1, **Chapter** 3), the details of absolute transform at levels 0 and 1 can be found in **Appendix** 5.6. After above procedure, the number of covariates reduces from 8 (see **Table** 5.3) to 4 (see **Table** 5.4) which can save computing resources in the final model. x_0 and x_1 have a similar meaning to the features discussed in **Section** 3.2.1, **Chapter** 3, providing a summary information at level 0 and level 1. The new covariates in each slide window from **Table** 5.3 become:

	Level 0 (Scale 6)	Level 1 (Scale 5)	Level 2+3+4 (Scale 4,3 and 2)
Covariates	x_0	x_1	PC1 score and PC2 score

Table 5.4: The 4 covariates in each slide window

Previously, the covariates PC1 score and PC2 score were shown in **Section** 5.3.3. The covariates x_0 and x_1 for both hands are given in **Figures** 5.28 and 5.29 (left hand), **Figures** 5.30 and 5.31 (right hand), which can be found in **Appendix** 5.6. They may provide further information for the classification of those three activities. More specifically, the x_0 and x_1 have 3 different curves which may present the properties of 3 activities in different slide windows.

5.4 Classification based on the small DWT tree structure from slide window

Preece et al. (2009) extract the scalar features through the **DWT** for classification of dynamic activities from the designed accelerometer data. However, the scalar features are a kind of summary statistics information, which ignore the detailed information from the coefficients of whole **DWT** tree structures. We now focus on the pattern of small **DWT** tree structure in each slide window. Four covariates (**Table** 5.4) can be extracted from them through the methods which are discussed in **Sections** 5.3.3 and 5.3.4. Using this more detailed information to evaluate the performance of these 4 covariates in the slide windows, we employed three classification models by using these 4 covariates. In other words, if the three classification models distinguish the three designed movements well, it means that we refine the "appropriate" information from the small **DWT** tree structure.

The three classification models are as follows: (i) Gaussian process classifier (GPC) (Ramsay & Silverman, 2005) by using function **gausspr** with Radial Basis kernel function and logit link function in R package **kernlab**; (ii) Support Vector Machine (SVM) through the function **svm** in R package **e1071** with Radial Basis kernel function; (iii) Random forest (RF) with R package **randomForest**.

Now we have 3 classes of movements: opening jars, walking and running, which

correspond to labels z=1, z=2 and z=3 for each slide window; a total of 4 covariates x_0, x_1, PC1 score and PC2 score. The confusion matrix, shown in **Table** 5.5, is used to measure the performance of the classification model. It contains the predicted class in each row and each column represents the actual class. So the confusion matrix with 3 classes is as follows:

	Predict = Walking	Predict =Opening	Predict = Running
Label = Walking	a	b	c
Label = Opening	d	e	f
Label = Jumping	g	h	i

Table 5.5: The confusion matrix with 3 movements.

We use 3 main indices of 3 classes, opening jars, walking and running, as followed:

$$\begin{aligned} \text{Recall}^{\mathbf{W}} &= \frac{a}{a+b+c}, \\ \text{Precision}^{\mathbf{W}} &= \frac{a}{a+d+g}, \\ \text{F-score}^{\mathbf{W}} &= \frac{2 * \text{ Recall}^{\mathbf{O}}\text{Precision}^{\mathbf{W}}}{\text{Recall}^{\mathbf{W}} + \text{Precision}^{\mathbf{W}}}, \end{aligned} \tag{5.15}$$

where **W** represents the activity of walking. Similarly, we can define the measures for opening jars:

$$\begin{aligned} \text{Recall}^{\mathbf{O}} &= \frac{e}{d+e+f}, \\ \text{Precision}^{\mathbf{O}} &= \frac{e}{b+e+h}, \\ \text{F-score}^{\mathbf{O}} &= \frac{2 * \text{ Recall}^{\mathbf{O}}\text{Precision}^{\mathbf{O}}}{\text{Recall}^{\mathbf{O}} + \text{Precision}^{\mathbf{O}}}, \end{aligned} \tag{5.16}$$

and for running:

$$\begin{aligned} \text{Recall}^{\mathbf{R}} &= \frac{i}{g+h+i}, \\ \text{Precision}^{\mathbf{R}} &= \frac{i}{c+f+i}, \\ \text{F-score}^{\mathbf{R}} &= \frac{2 * \text{ Recall}^{\mathbf{R}}\text{Precision}^{\mathbf{R}}}{\text{Recall}^{\mathbf{R}} + \text{Precision}^{\mathbf{R}}}. \end{aligned} \tag{5.17}$$

5.4.1 The performance of the three classification models by using 4 covariates from small DWT tree structure

The results of this classification are given for the left hand and right hand separately, and the performance is investigated by using leave-one-patient-out cross validation in three

classification models. The comparison of recall, precision and F-score for different methods is given in **Table** 5.6. Other details are shown from **Figure** 5.32 to **Figure** 5.37 and from **Table** 5.7 to **Table** 5.18, which can be found in **Appendix** 5.6.

GPC	Left hand			Right hand		
	Recall	Precision	F-score	Recall	Precision	F-score
Label = Walking	0.960	0.992	0.976	0.959	0.983	0.971
Label = Opening	0.992	0.962	0.977	0.984	0.960	0.972
Label = Running	1	1	1	1	1	1
SVM	Left hand			Right hand		
	Recall	Precision	F-score	Recall	Precision	F-score
Label = Walking	0.937	0.992	0.963	0.911	1	0.953
Label = Opening	0.992	0.940	0.965	1	0.918	0.957
Label = Running	1	1	1	1	1	1
RF	Left hand			Right hand		
	Recall	Precision	F-score	Recall	Precision	F-score
Label = Walking	1	0.992	0.996	1	1	1
Label = Opening	0.992	1	0.996	1	1	1
Label = Running	1	1	1	1	1	1

Table 5.6: The Recall, Precision and F-score for the 3 movements in both hands by using Gaussian process classifier (GPC), Support Vector Machine (SVM) and Random forest (RF); each slide window has length 128 and uses the designed data.

The Random forest model has the best performance with these three classification methods, followed by the Gaussian process classifier and Support Vector Machine models. The classification indices (Recall, Precision and F-score), given in **Table** 5.6, are all over 90% for both hands, which means the three classification models show the good performance when dealing with the classification of the three movements. In other words, the useful information is extracted from the pattern of small **DWT** tree structure in slide window based on the time and frequency-domains.

5.4.2 The sensitiveness of selecting the length of slide window

The results shown in **Section** 5.4.1 are based on the slide window with a length 128 points with the designed data (totally 62 **DWT** coefficients from level 0 to level 4). The sensitivity of the number of **DWT** coefficients are used in each slide window needs to be investigated. Besides selecting the slide window with length 128 (correspond to about 2 seconds' data), the comparison for classification indices (Recall, Precision and F-

score), from three classification models based on the length of 64 (corresponding to about 1 second's data) to 192 (corresponding to about 3 seconds' data), are given as follows:

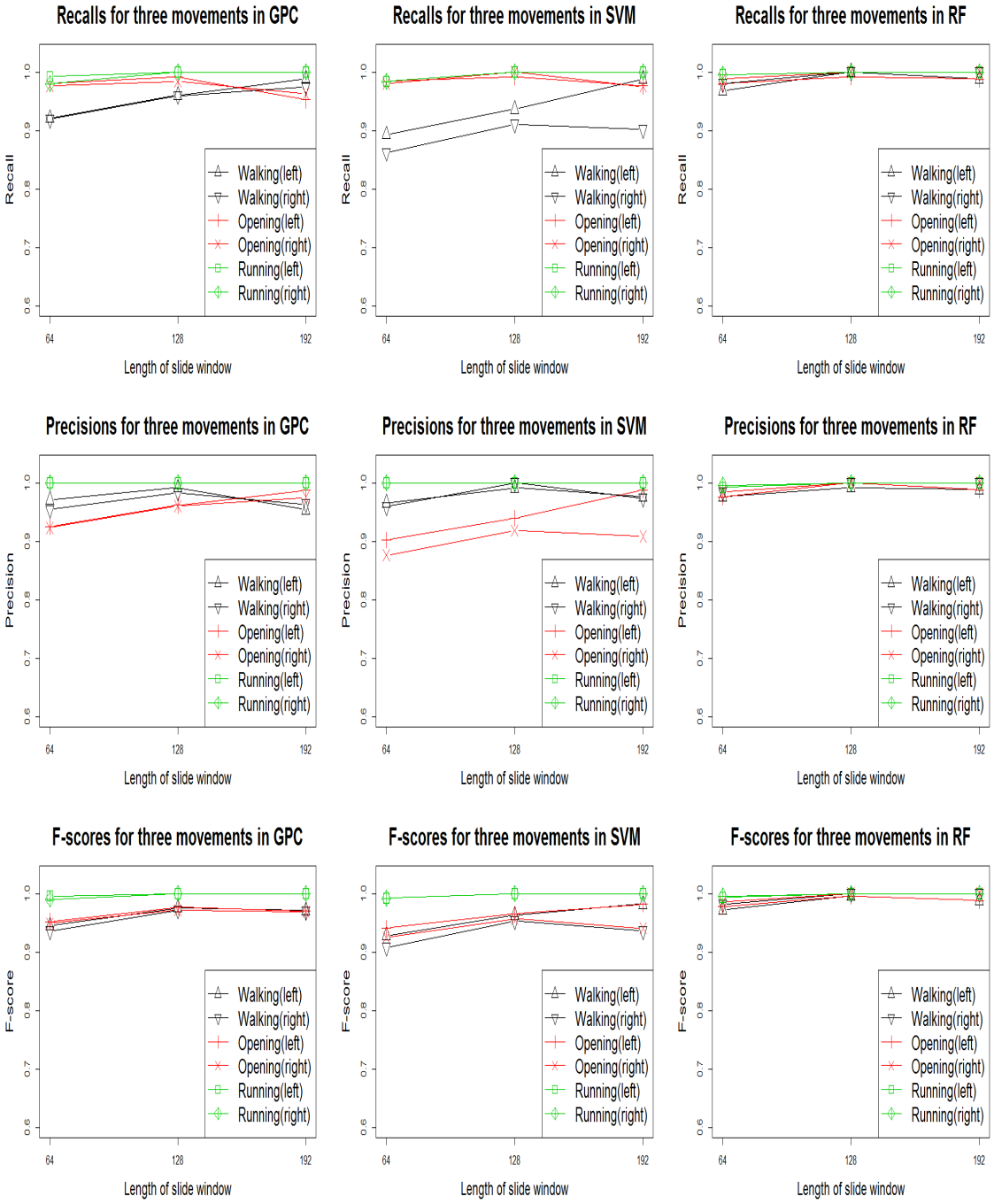


Figure 5.13: The recalls, precisions and F-scores for three classification methods in three movements. Note that the left and right stand for the left hand and the right hand.

In summary, **Figure** 5.13 shows that the classification indices (Recall, Precision and

F-score) are not sensitive for the number of wavelet coefficients are used in each slide window. The details of confusion matrix and the classification indices (Recall, Precision and F-score), based on the length of 64 (corresponding to about 1 second's data) and 192 (corresponding to about 3 seconds' data), are given from **Table** 5.19 to **Table** 5.22 in **Appendix** 5.6. In other words, the method in this chapter can be used as a general method. Despite the large sample size in the slide window gives relative good classification results based on the designed data, but we still recommend choosing the proper sample size since we need to deal with the relatively complex free-living data. To be more specific, unlike the designed data, the large sample size in the slide window corresponds to a pattern with relative long period in the free-living data may contain too much movements, this may reduce the efficiency of the algorithm.

5.5 Conclusion

Using design data from a wrist-worn sensor, AX3, the slide window and multivariate **fPCA** approaches are investigated and the useful information from the pattern of small **DWT** tree structure are analysed. We use them to carry out the classification by using three classification models to distinguish three movements, opening jars, walking and running. The three classification model, based on the extracted new covariates (PC1 score, PC2 score, x_1 and x_0) from each small **DWT** tree structure, all present good results. However, the comparison of different options for choosing the sample size in slide window are not sensitive.

Although the useful information is extracted with small sizes for both hand, and it performs well by investigating three classification models for three movements, we build the classification model for each hand separately. We may also use the data from both hands. This method is different to scalar features in **Chapter** 3, focusing on the pattern from each small **DWT** tree structure. This is an alternative approach in looking for the new features, which will be used for the stroke rehabilitation assessment in free-living environments. This problem will be discussed in the next chapter.

Related studies involving for classification or clustering based on the accelerometer data (collected by cheap wearable device) has attracted a lot of researchers' interest in both statistics and computer science. Siirtola et al. (2009) present a method for classifying sport activities. The simple features (e.g., variance and mean) from clustering data which imply that the data after clustering can provide some valuable information for the different movments. Yin & Huang (2015) focus on fault detection and isolation for vehicle suspension systems through confirming the number of clusters based on principal component analysis; Van Kuppevelt et al. (2018) present a data-driven approach for clustering the accelerometer data also using principal component analysis. Both of them present

that the principal components may play an enssential role to reduce the data set in accelerometer data. Preece et al. (2009) present a method to extract the features based on wavelet-domain to classify activities. Instead of applying the accelerometer in time-domain only, it is an alternative approach that gives an analysis of **DWT** coefficients in both time and frequency-domains from the accelerometer data. Nguyen et al. (2007) investigate unsupervised pattern recognition approaches to cluster free-living activities, which suggests that clustering the pattern's information from accelerometer data is a potential approach in the free-living environment. Tang et al. (2020) propose a method for clustering and extracting new features to build the predictive model for stroke rehabilitation. Combining it with previous methods, could be useful in clustering and extracting the features from the pattern based on the wavelet domain through the principal component analysis, then using these clustering features to measure the recovery level for the stroke patients. The methods proposed in this chapter have shown good potential for solving similar problems involving bigger data.

Here is the supplementary instruction that why not use the NIG-MT in this chapter. The data reduction (NIG-MT) aims to remove the noise and 'quiet period' from the accelerometer data at the computation level. After the data reduction, we can obtain a smaller size of the data (comparing with the raw data, it reduces the size by three quarters). The small size of the data (after removing the noise and 'quiet period') will make it easier to do further analysis in **Chapter** 6. We should notice that the designed data in this chapter is just 1 min (See **Section** 5.1). To be more specific, compared to the free-living data, the designed data has relatively small data size and designs for the certain movements; so we don't need to apply the data reduction. However, the free-living data for **Chapter** 6 (the real free-living data) records for 3 full days which contains too many unknown actions, so it is complex and it is necessary to apply the data reduction. After we remove the 'noise' and 'still periods', the DWT tree structure is easier to analyze.

5.6 Appendix

PC score from left hand in three individuals

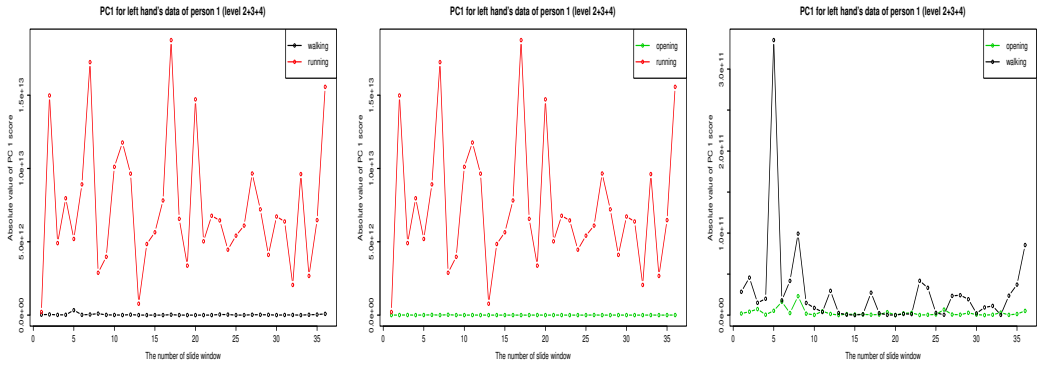


Figure 5.14: The first PC absolute score for the three movements in the first person's data (left hand).

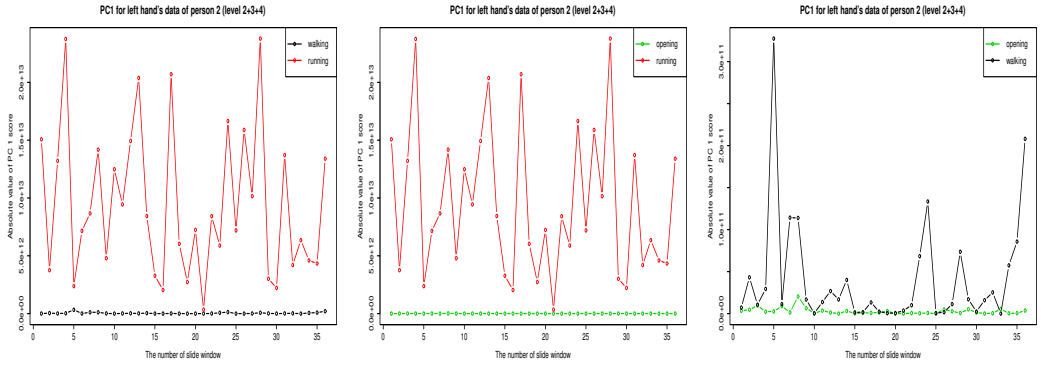


Figure 5.15: The first PC absolute score for the three movements in the second person's data (left hand).

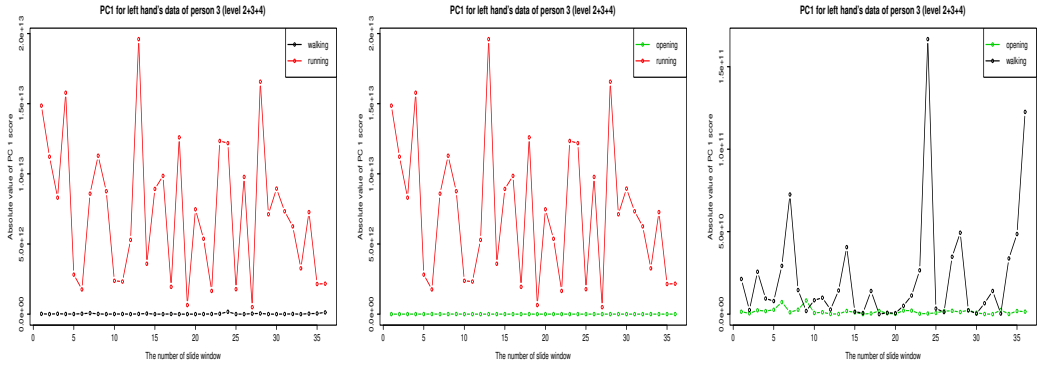


Figure 5.16: The first PC absolute score for the three movements in the third person's data (left hand).

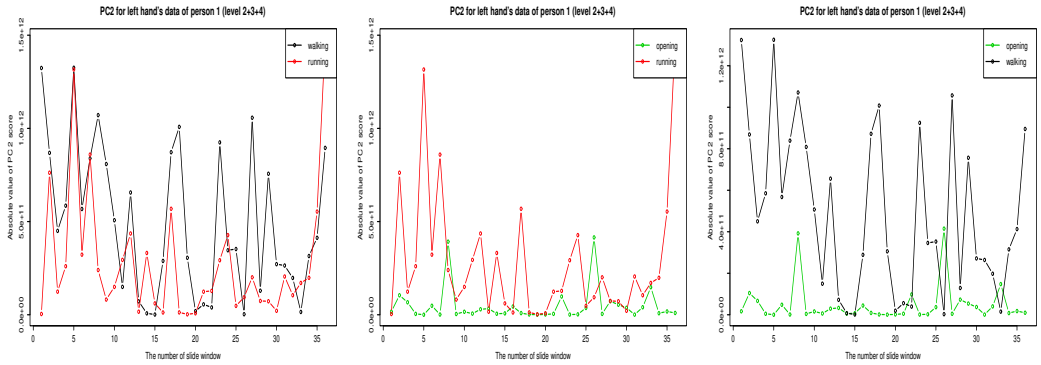


Figure 5.17: The second PC absolute score for the three movements in the first person's data (left hand).

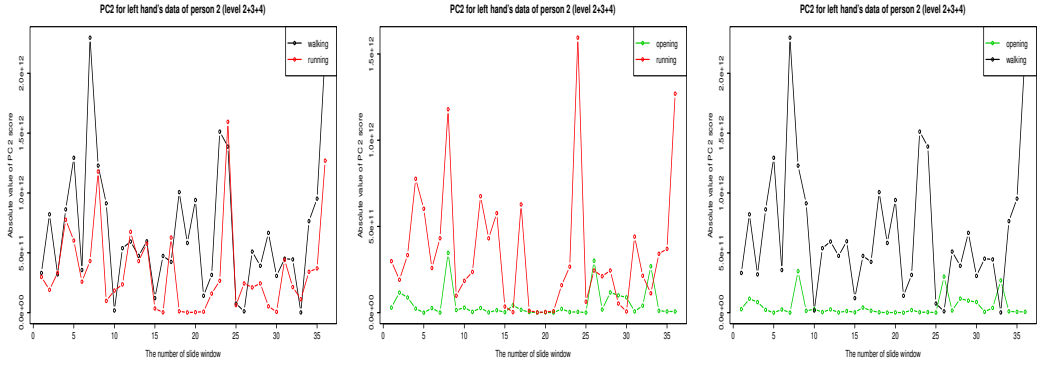


Figure 5.18: The second PC absolute score for the three movements in the second person's data (left hand).

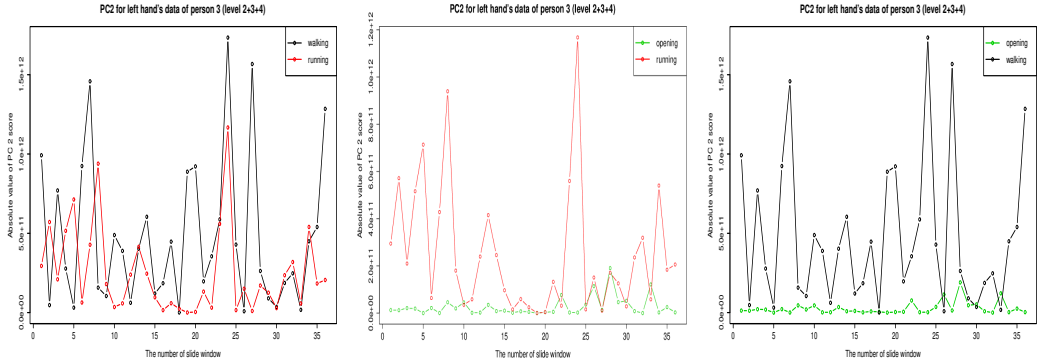


Figure 5.19: The second PC absolute score for the three movements in the third person's data (left hand).

PC score from right hand for three individuals

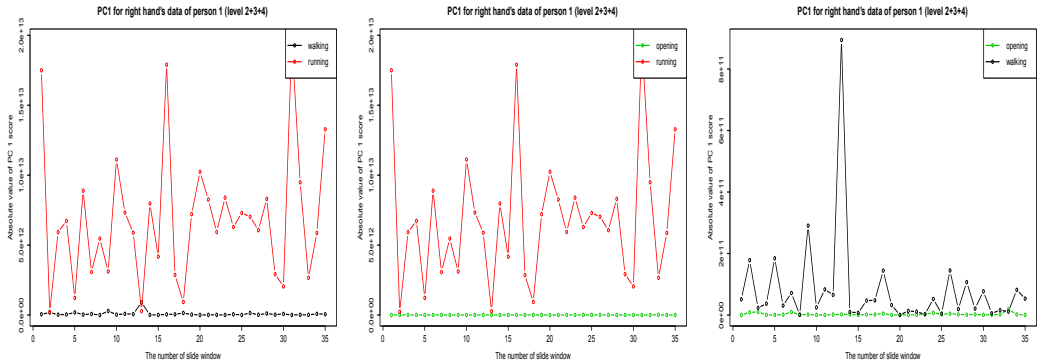


Figure 5.20: The first PC absolute score for the three movements in the first person's data (right hand).

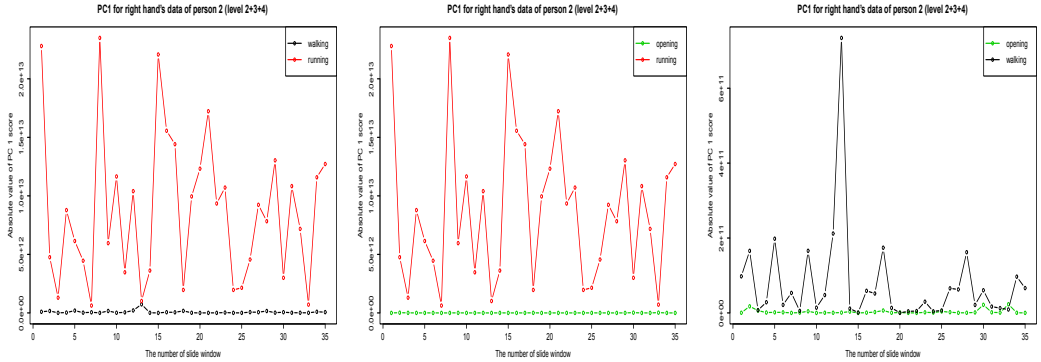


Figure 5.21: The first PC absolute score for the three movements in the second person's data (right hand).

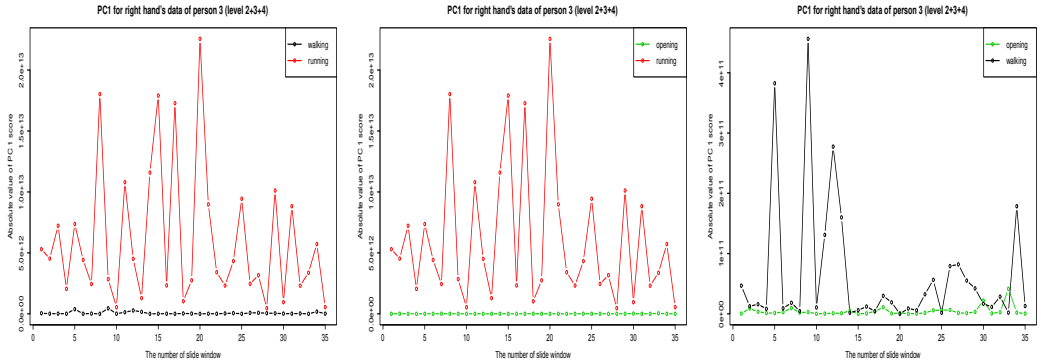


Figure 5.22: The first PC absolute score for the three movements in the third person's data (right hand).

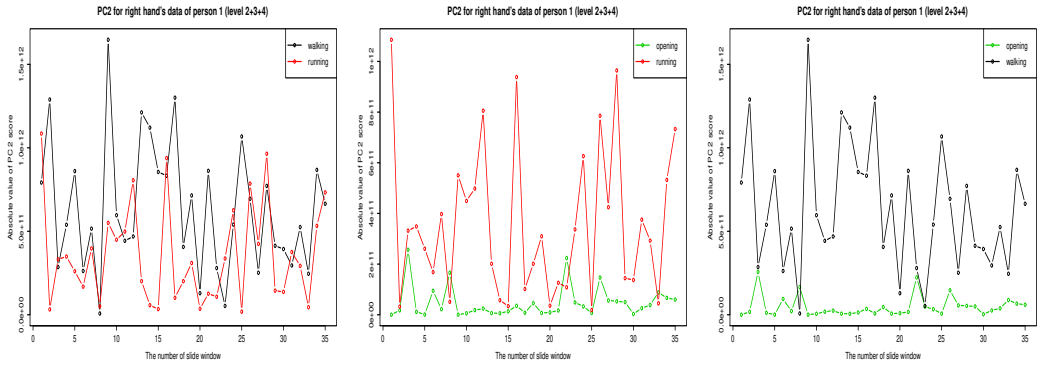


Figure 5.23: The second PC absolute score for the three movements in the first person's data (right hand).

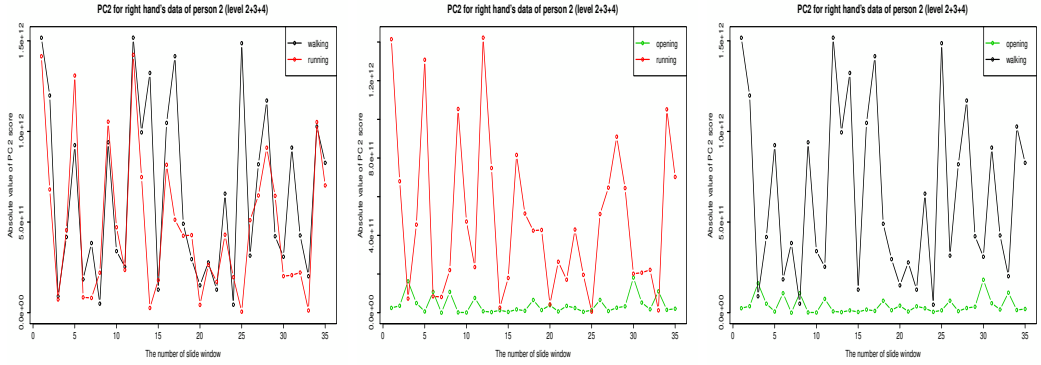


Figure 5.24: The second PC absolute score for the three movements in the second person's data (right hand).

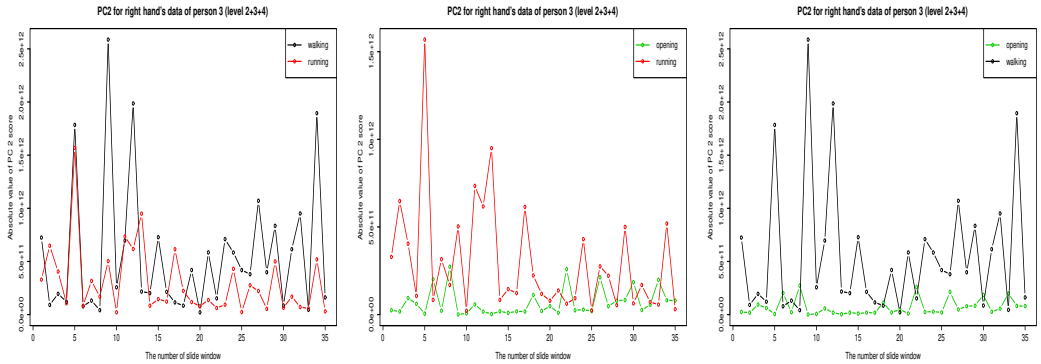


Figure 5.25: The second PC absolute score for the three movements in the third person's data (right hand).

The absolute transform at levels 0 and 1

Now we take the covariates at level 0 as an example and the absolute transform of covariates x_{01} for three movements in different slide windows is shown in **Figure** 5.26.

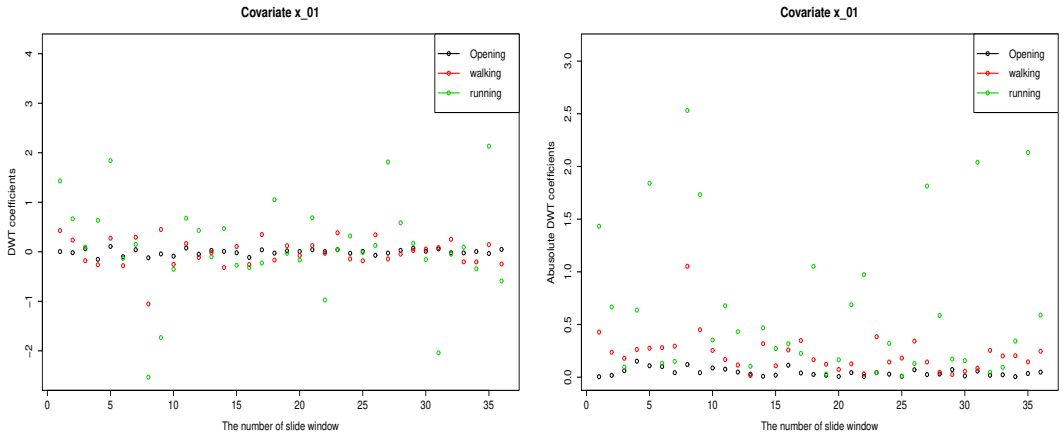


Figure 5.26: The absolute transform of covariate x_{01} at level 0.

Similarly, we can perform the absolute transform for covariates x_{02}, x_{11}, x_{12}, x_{13} and x_{14}. Now the slide window at the level 0 and level 1 contains 2 and 4 covariates respectively. However, from **Figure** 5.26, the covariate x_{01} for three movements is so similar, that it is hard to distinguish between the three movements.

Now let $x_0 = (x_{01} + x_{02})/2$ and $x_1 = (x_{10} + x_{12} + x_{13} + x_{14})/4$ be new covariates at level 0 and level 1. After combining x_{01}, x_{02} at level 0 and x_{11}, x_{12}, x_{13}, x_{14} at level 1 respectively, **Figure** 5.27 shows the three movements which can be distinguished efficiently by new covariates x_0 and x_1 for each individual.

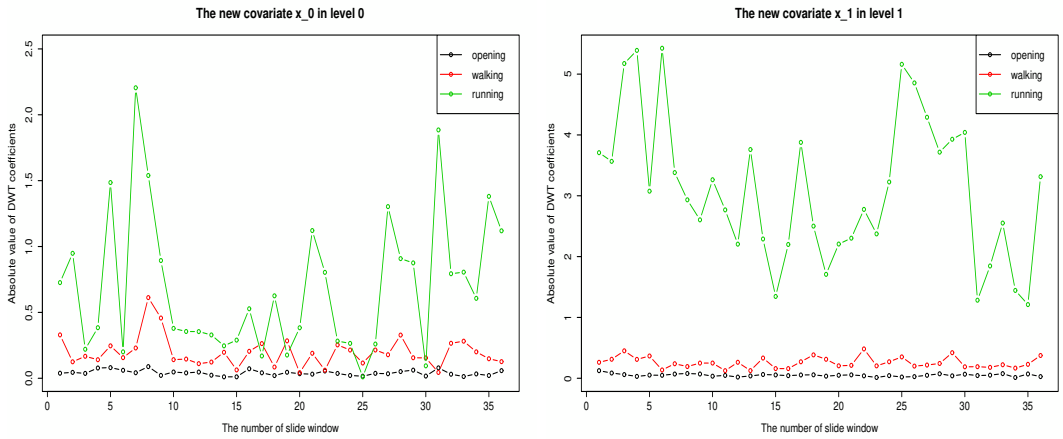


Figure 5.27: The absolute value of covariates x_0 and x_1 at level 0 and level 1.

The x_0 and x_1 from three movements in three people's data

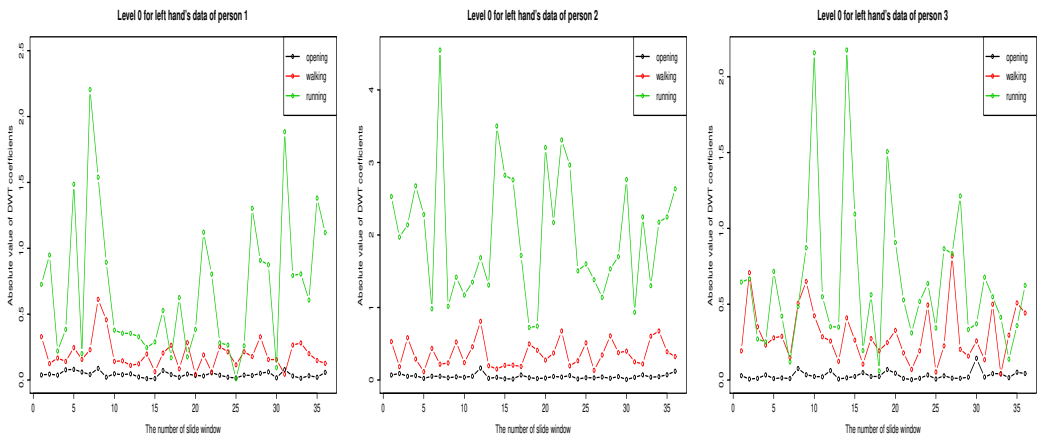


Figure 5.28: x_0 of three person's data with three movements (left hand).

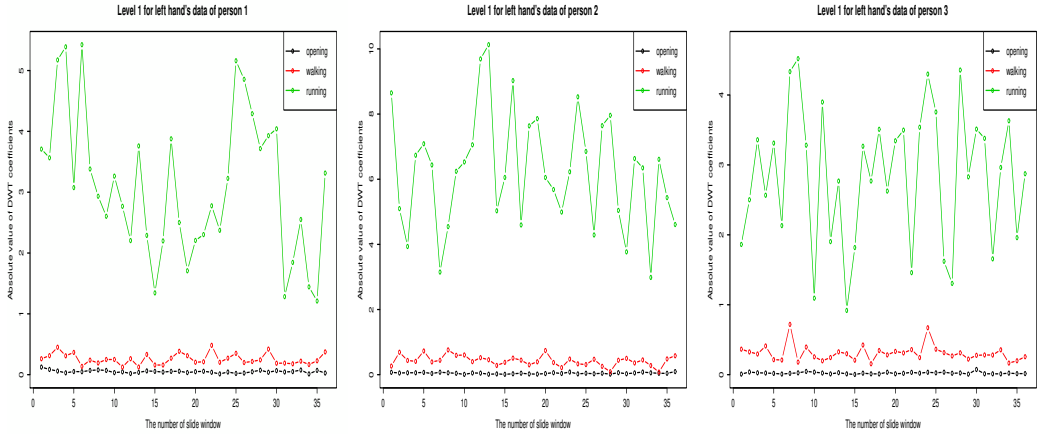


Figure 5.29: x_1 of three person's data with three movements (left hand).

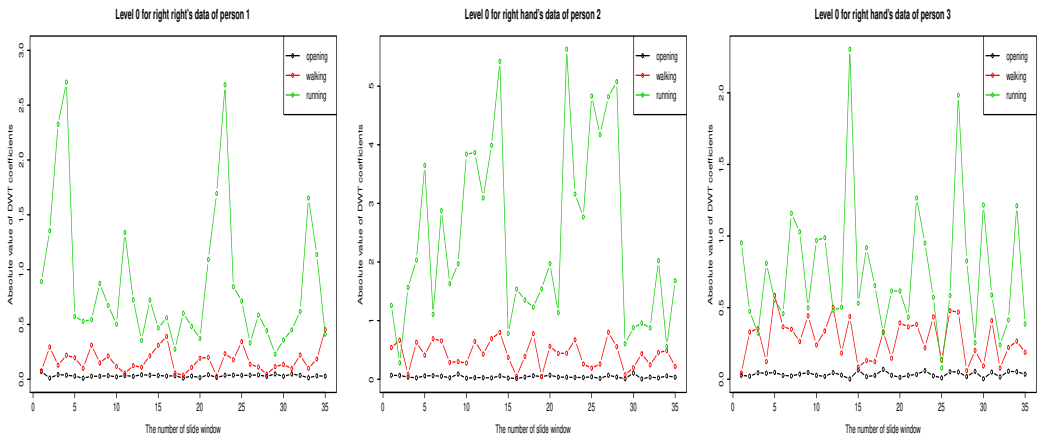


Figure 5.30: x_0 of three person's data with three movements (right hand).

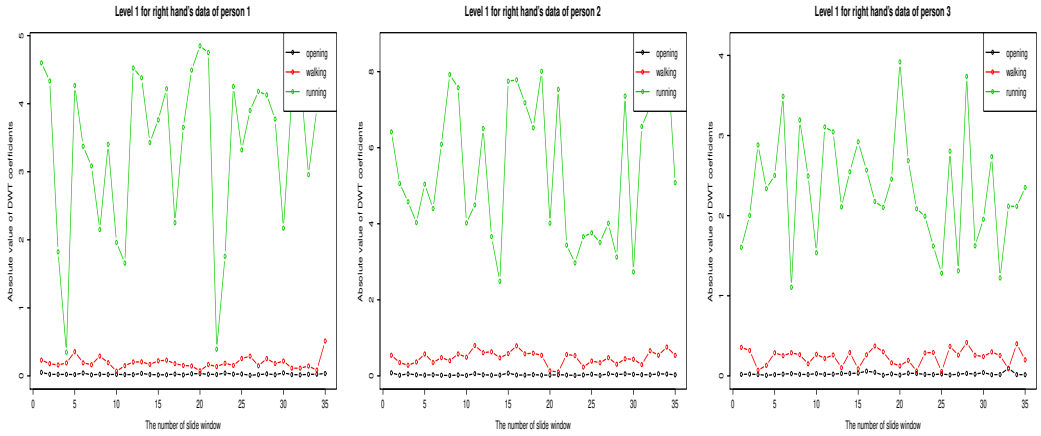


Figure 5.31: x_1 of three person's data with three movements (right hand).

Gaussian process classifier for the left hand

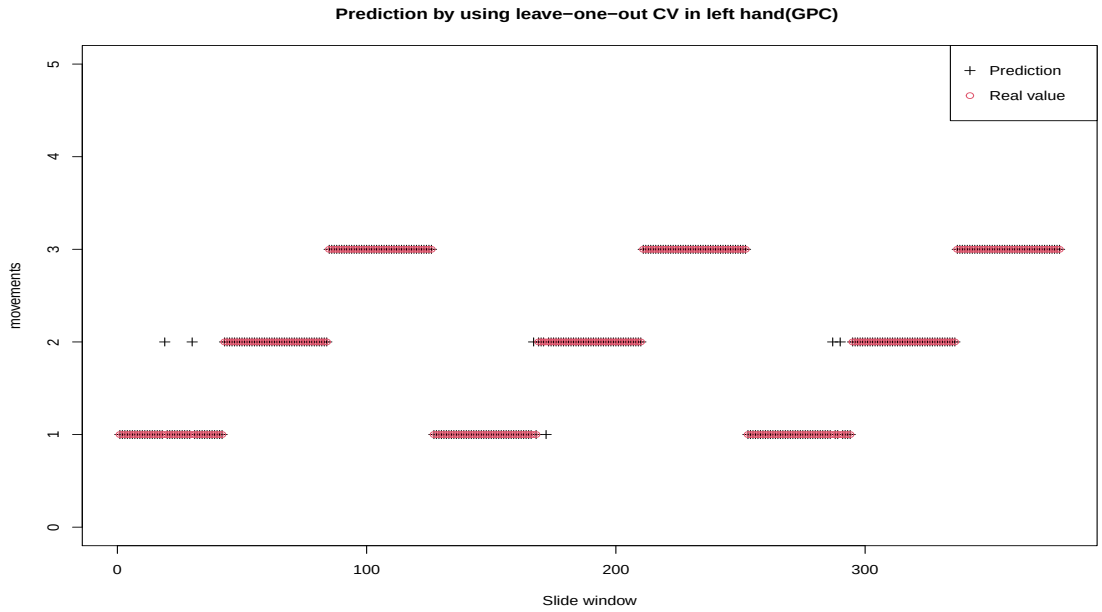


Figure 5.32: Prediction against real value by using leave-one-patient-out cross validation in the left hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Gaussian process classifier.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	121	5	0
Label = Opening	1	125	0
Label = Running	0	0	126

Table 5.7: The confusion matrix for 3 movements in the left hand by using Gaussian process classifier.

	Recall	Precision	F-score
Label = Walking	0.9603175	0.9918033	0.9758065
Label = Opening	0.9920635	0.9615385	0.9765625
Label = Running	1	1	1

Table 5.8: The Recall, Precision and F-score for the 3 movements in the left hand by using Gaussian process classifier.

Gaussian process classifier for the right hand

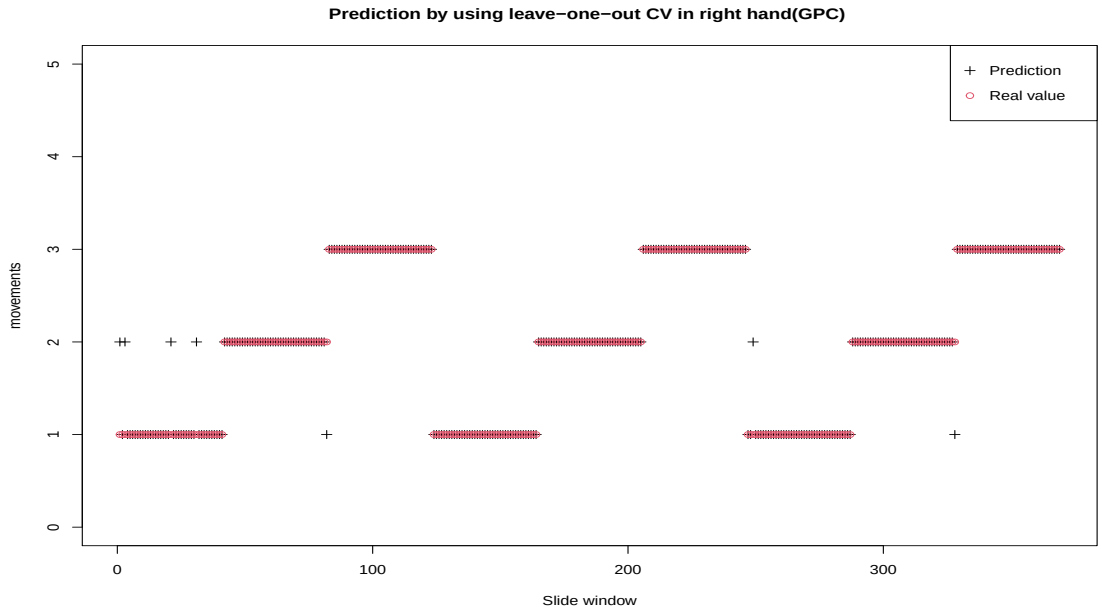


Figure 5.33: Prediction against real value by using leave-one-patient-out cross validation in the right hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Gaussian process classifier.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	118	5	0
Label = Opening	2	121	0
Label = Running	0	0	123

Table 5.9: The confusion matrix for 3 movements in the right hand by using Gaussian process classifier.

	Recall	Precision	F-score
Label = Walking	0.9593496	0.9833333	0.9711934
Label = Opening	0.9837398	0.9603175	0.9718876
Label = Running	1	1	1

Table 5.10: The Recall, Precision and F-score for the 3 movements in the right hand by using Gaussian process classifier.

Support Vector Machine for the left hand

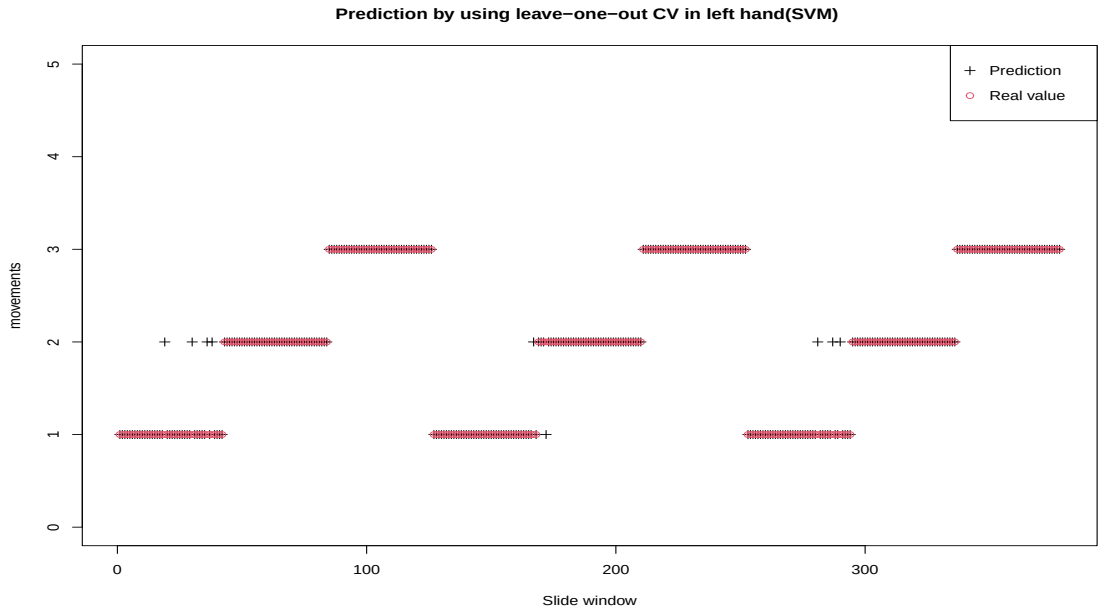


Figure 5.34: Prediction against real value by using leave-one-patient-out cross validation in the left hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Support Vector Machine.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	118	8	0
Label = Opening	1	125	0
Label = Running	0	0	126

Table 5.11: The confusion matrix for 3 movements in the left hand by using Support Vector Machine.

	Recall	Precision	F-score
Label = Walking	0.9365079	0.9915966	0.9632653
Label = Opening	0.9920635	0.9398496	0.965251
Label = Running	1	1	1

Table 5.12: The Recall, Precision and F-score for the 3 movements in the left hand by using Support Vector Machine.

Support Vector Machine for the right hand

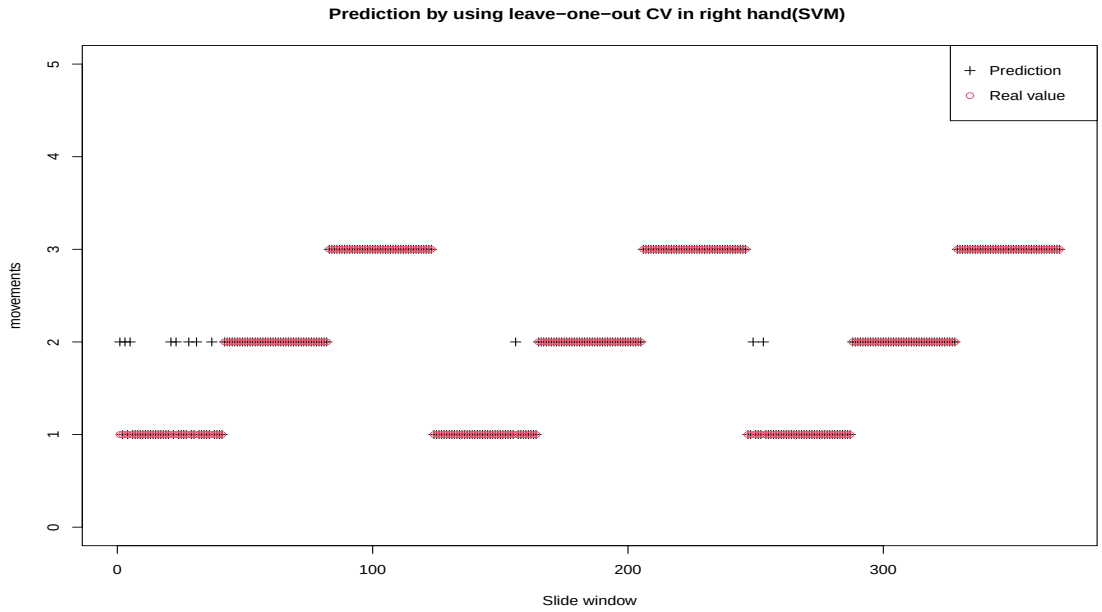


Figure 5.35: Prediction against real value by using leave-one-patient-out cross validation in the right hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Support Vector Machine.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	112	11	0
Label = Opening	0	123	0
Label = Running	0	0	123

Table 5.13: The confusion matrix for 3 movements in the right hand by using Support Vector Machine.

	Recall	Precision	F-score
Label = Walking	0.9105691	1	0.9531915
Label = Opening	1	0.9179104	0.9571984
Label = Running	1	1	1

Table 5.14: The Recall, Precision and F-score for the 3 movements in the right hand by using Support Vector Machine.

Random forest for the left hand

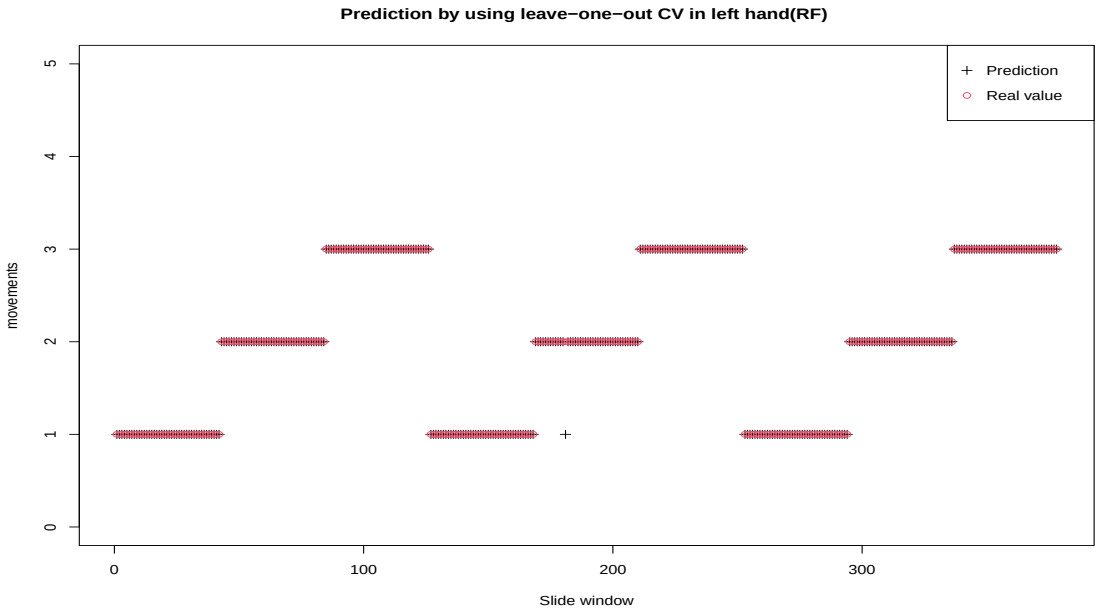


Figure 5.36: Prediction against real value by using leave-one-patient-out cross validation in the left hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Random forest.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	126	0	0
Label = Opening	1	125	0
Label = Running	0	0	126

Table 5.15: The confusion matrix for 3 movements in the left hand by using Random forest.

	Recall	Precision	F-score
Label = Walking	1	0.992126	0.9960474
Label = Opening	0.9920635	1	0.9960159
Label = Running	1	1	1

Table 5.16: The Recall, Precision and F-score for the 3 movements in the left hand by using Random forest.

Random forest for the right hand

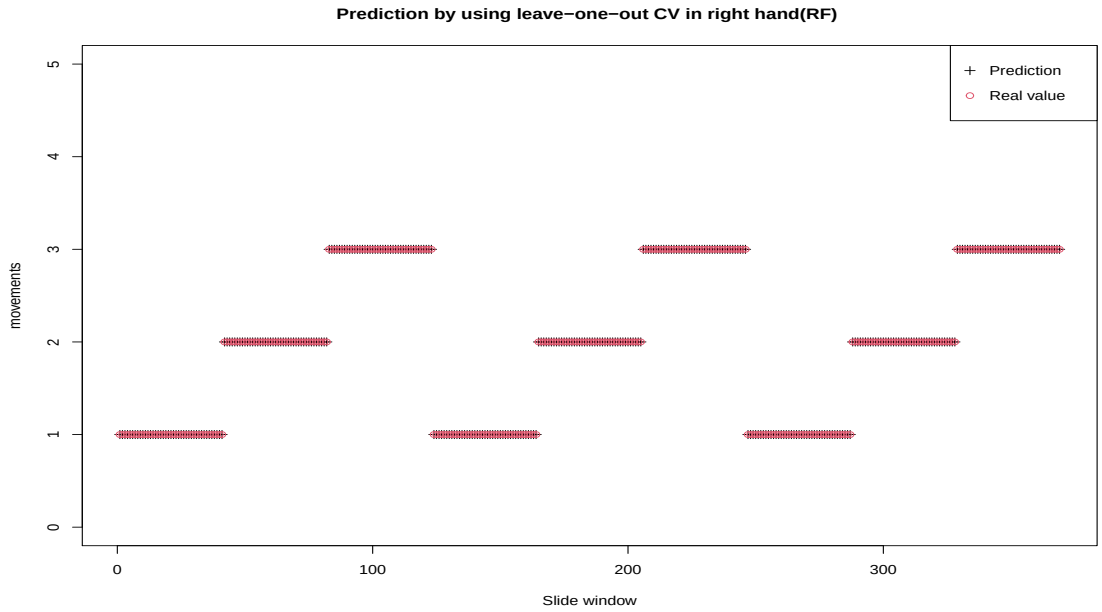


Figure 5.37: Prediction against real value by using leave-one-patient-out cross validation in the right hand. Note that z=1 is the movement of walking, z=2 is the movement of opening jars and z=3 is the movement of running by using Random forest.

	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	123	0	0
Label = Opening	0	123	0
Label = Running	0	0	123

Table 5.17: The confusion matrix for 3 movements in the right hand by using Random forest.

	Recall	Precision	F-score
Label = Walking	1	1	1
Label = Opening	1	1	1
Label = Running	1	1	1

Table 5.18: The Recall, Precision and F-score for the 3 movements in the right hand by using Random forest.

Length selection of 64 in each slide window from the desinged data

The confusion matrix

Left hand (GPC)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	232	20	0
Label = Opening	5	247	0
Label = Running	2	0	250
Right hand (GPC)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	226	20	0
Label = Opening	6	240	0
Label = Running	5	0	241
Left hand (SVM)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	225	27	0
Label = Opening	4	248	0
Label = Running	4	0	248
Right hand (SVM)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	212	34	0
Label = Opening	5	241	0
Label = Running	4	0	242
Left hand (RF)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	244	6	2
Label = Opening	5	247	0
Label = Running	1	0	251
Right hand (RF)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	241	4	1
Label = Opening	3	243	0
Label = Running	1	0	245

Table 5.19: The confusion matrix for 3 movements in both hands by using Gaussian process classifier (GPC), Support Vector Machine (SVM) and Random forest (RF); each slide window has length 64 and uses the designed data.

The classification indices (Recall, Precision and F-score)

Left hand (GPC)	Recall	Precision	F-score
Label = Walking	0.921	0.971	0.945
Label = Opening	0.980	0.925	0.952
Label = Running	0.992	1	0.996
Right hand (GPC)	Recall	Precision	F-score
Label = Walking	0.919	0.954	0.936
Label = Opening	0.976	0.923	0.949
Label = Running	0.980	1	0.990
Left hand (SVM)	Recall	Precision	F-score
Label = Walking	0.893	0.966	0.928
Label = Opening	0.984	0.902	0.941
Label = Running	0.984	1	0.992
Right hand (SVM)	Recall	Precision	F-score
Label = Walking	0.862	0.959	0.9080
Label = Opening	0.980	0.876	0.925
Label = Running	0.984	1	0.992
Left hand (RF)	Recall	Precision	F-score
Label = Walking	0.968	0.976	0.972
Label = Opening	0.980	0.976	0.978
Label = Running	0.996	0.992	0.994
Right hand (RF)	Recall	Precision	F-score
Label = Walking	0.980	0.984	0.982
Label = Opening	0.988	0.984	0.986
Label = Running	0.996	0.996	0.996

Table 5.20: The Recall, Precision and F-score for the 3 movements in both hands by using Gaussian process classifier (GPC), Support Vector Machine (SVM) and Random forest (RF); each slide window has length 64 and uses the designed data.

Length selection of 192 in each slide window from the desinged data

The confusion matrix

Left hand (GPC)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	83	1	0
Label = Opening	4	80	0
Label = Running	0	0	84
Right hand (GPC)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	79	2	0
Label = Opening	3	78	0
Label = Running	0	0	81
Left hand (SVM)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	83	1	0
Label = Opening	2	82	0
Label = Running	0	0	84
Right hand (SVM)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	73	8	0
Label = Opening	2	79	0
Label = Running	0	0	81
Left hand (RF)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	83	1	0
Label = Opening	1	83	0
Label = Running	0	0	84
Right hand (RF)	Predict = Walking	Predict = Opening	Predict = Running
Label = Walking	81	0	0
Label = Opening	0	81	0
Label = Running	0	0	81

Table 5.21: The confusion matrix for 3 movements in both hands by using Gaussian process classifier (GPC), Support Vector Machine (SVM) and Random forest (RF); each slide window has length 192 and uses the designed data.

The classification indices (Recall, Precision and F-score)

Left hand (GPC)	Recall	Precision	F-score
Label = Walking	0.9880952	0.954023	0.9707602
Label = Opening	0.952381	0.9876543	0.969697
Label = Running	1	1	1
Right hand (GPC)	Recall	Precision	F-score
Label = Walking	0.9753086	0.9634146	0.9693252
Label = Opening	0.962963	0.975	0.9689441
Label = Running	1	1	1
Left hand (SVM)	Recall	Precision	F-score
Label = Walking	0.9880952	0.9764706	0.9822485
Label = Opening	0.9761905	0.9879518	0.9820359
Label = Running	1	1	1
Right hand (SVM)	Recall	Precision	F-score
Label = Walking	0.9012346	0.9733333	0.9358974
Label = Opening	0.9753086	0.908046	0.9404762
Label = Running	1	1	1
Left hand (RF)	Recall	Precision	F-score
Label = Walking	0.9880952	0.9880952	0.9880952
Label = Opening	0.9880952	0.9880952	0.9880952
Label = Running	1	1	1
Right hand (RF)	Recall	Precision	F-score
Label = Walking	1	1	1
Label = Opening	1	1	1
Label = Running	1	1	1

Table 5.22: The Recall, Precision and F-score for the 3 movements in both hands by using Gaussian process classifier (GPC), Support Vector Machine (SVM) and Random forest (RF); each slide window has length 192 and uses the designed data.

Chapter 6

Stroke rehabilitation assessment through the clustering features

In **Chapters** 4 and 5, the **DWT** tree structure was used in the crucial task of data reduction and dimension reduction. In other words, the posterior marginal probabilities $Pr(S_{j,k} = 1|\mathcal{D})$ can be applied to remove the "noise" and "still period" data through the whole **DWT** tree structures. The whole **DWT** tree structure can be divided into smaller tree structures through the slide window approach and the multivariate **fPCA** can also be used to transform the **DWT** coefficients to the PC scores from decomposition level 2 to level 4 through the small **DWT** tree structures. In this chapter, we investigate the stroke patients' accelerometer data, the free-living data, for the stroke rehabilitation assessment. This combines the ideas from **Chapters** 4 and 5, based on the small **DWT** coefficients tree structure from free living environments.

To be more specific, we focus on the **DWT** coefficients which contain the information in both time and frequency-domains. It is better to analyze the patterns within the whole **DWT** coefficients tree structure if we want to give a more precise analysis in both time and frequency-domains. However, it is hard to analyze the **DWT** coefficients directly due to the special tree structure and large sample size from the free-living data. To solve this problem, we first keep **DWT** coefficients without 'noise' and 'still period' through the **NIG-MT** in **Chapter** 4. To further reduce the remaining **DWT** coefficients, the useful information (new covariates) is extracted via slide window and multivariate **fPCA** approaches discussed in **Chapter** 5.

In **Section** 3.2.1, **Chapter** 3, we discussed the idea that there are two types of patient, i.e., "can do but doesn't want to" and "couldn't do but wants to". Unfortunately, if we analyze the data from one hand only, using the new covariates from slide window and multivariate **fPCA** approaches set out in **Chapter** 5 for clinical assessed CAHAI scores prediction, the results are not satisfactory. Similar to **Chapter** 3, the task of combining

both hands information becomes a critical part. In this chapter, Gaussian Mixture Models (**GMM**), will be applied to define the new features based on the extracted information (new covariates) from the patterns of small **DWT** tree structures. Also we further develop a new stroke rehabilitation assessment system with these new features.

Overall, when looking for patterns within the whole **DWT** coefficients from the free-living data, a new data reduction system for the **DWT** coefficients can be developed via the **NIG-MT**, slide window and multivariate **fPCA** in **Chapters** 4 and 5. After this, the large scale of **DWT** coefficients from free-living data will be transferred to a smaller data set, which is easy to give a further more precise analysis in both time and frequency-domains.

6.1 Background of the stroke patient's free-living data

In this chapter, we use the same data set as **Chapter** 3 and describe the background of clinical assessed CAHAI score, participants, data collection and data preprocessing were introduced in **Section** 2.2, **Chapter** 2.

6.2 Data reduction through the NIG-MT model

Based on **Section** 4.4 in **Chapter** 4, let $y(t)$ be a process which represents the **VM** data (the signal vector magnitude of accelerometer data; see **Section** 2.2.3):

$$\mathbf{y}^{(j,k,p)} = (y^{(j,k,p)}(t_0), y^{(j,k,p)}(t_1), ..., y^{(j,k,p)}(t_{T-1})),$$
$$\mathbf{y}^{(j,k,np)} = (y^{(j,k,np)}(t_0), y^{(j,k,np)}(t_1), ..., y^{(j,k,np)}(t_{T-1})),$$

where $\mathbf{y}^{(j,k,p)}$ and $\mathbf{y}^{(j,k,np)}$ are observations at $t = t_i$, $i = 0, ..., T-1$ among the j-th patient at the k-th visit, $j = 1, ..., n$, $k = 1, ..., k_j$, n is the total number of the patients and k_j is varied for each patient and can be up to 8, p and np denote the paralysed side and non-paralysed side respectively for different patients.

After applying the **DWT** and transforming the accelerometer data to the **DWT** coefficients at the 6 decomposition levels. Based on the energy loss rate ***ELR*** in **Tables** 4.2 and 4.3 of **Section** 4.5, **Chapter** 4 also gives an introduction of how to select the threshold P in practice. We still select the threshold $P = 0.9$ in the $Pr(S_{j,k} = 1|\mathcal{D})$ via the **NIG-MT** model for the free-living data. For the j-th patient at the k-th visit, the small **DWT** coefficients will be forced to be zero and a total of 2 **DWT** coefficient tree structures (1 for paralysed side and 1 for non-paralysed side) will be obtained after shrinking:

$$\begin{aligned}\mathbf{DWTs}^{(j,k,p)} &= [\mathbf{W}_1^{(j,k,p)}, \mathbf{W}_2^{(j,k,p)}, ..., \mathbf{W}_6^{(j,k,p)}]^{\mathrm{T}}, \\ \mathbf{DWTs}^{(j,k,np)} &= [\mathbf{W}_1^{(j,k,np)}, \mathbf{W}_2^{(j,k,np)}, ..., \mathbf{W}_6^{(j,k,np)}]^{\mathrm{T}}.\end{aligned} \tag{6.1}$$

where $\mathbf{DWTs}^{(j,k,p)}$ and $\mathbf{DWTs}^{(j,k,np)}$ are two column vectors for paralysed side and non-paralysed side, $\mathbf{W}_1^{(j,k,p)}$ to $\mathbf{W}_6^{(j,k,p)}$ are the **DWT** coefficients at 6 decomposition levels for the j-th patient at the k-th visit in paralysed side (present) and $\mathbf{W}_1^{(j,k,np)}$ to $\mathbf{W}_6^{(j,k,np)}$ are the corresponding **DWT** coefficients in non-paralysed side.

6.2.1 Obtain the proper non-zero DWT tree structure

After obtaining the **DWT** tree structure in **Equation** (6.1), we can remove the zero values from the **DWT** tree structure through data reduction. There is, however, particular phenomenon that needs to be dealt with carefully. From **Table** 4.3 in **Section** 4.5, **Chapter** 4, about 80% and 40% of **DWT** coefficients at levels 0 and 5 respectively are forced to be zero through the **NIG-MT** model, which means that the **DWT** tree structure in **Equation** (6.1) contains more zero at the level 5 than at the level 0. If we just remove all the zero **DWT** coefficients, they will not form a real tree structure (since every node at the previous level corresponds to two nodes at the next level). Recall the property of the **DWT** tree structure from **Section** 4.1, **Chapter** 4, that if the parent node at level j is close to zero (or equal to zero), then the two children nodes will tend to be close to zero (or equal to zero) as well at the next level $j+1$. Why more **DWT** coefficients are shrunk to zero at the bottom **DWT** decomposition level by using the **NIG-MT** model? To explain this particular phenomenon, the **DWT** coefficients, at higher levels of **DWT** decomposition, correlate more closely to the high frequency domain and so more noise is generated. Then we go back to the hyperparameters η, γ in **Section** 4.3, **Chapter** 4. Due to the fact that the exponential decaying factor 2^{-j} counters exactly the exponential increase in the expected number of wavelet coefficients in higher resolution, the posterior marginal probability $Pr(S_{j,k} = 1|\mathcal{D})$ will be smaller at the bottom decomposition level than it at the top and, therefore, the **DWT** decomposition will be shrunk more at the bottom decomposition level.

To obtain the **DWT** tree structure properly, after shrinking the smaller coefficients to be zero, instead of removing all of zero **DWT** coefficients from all decomposition levels, we just remove those at the level 0 and their children nodes at following levels (from level 1 to level 5). To be more specific, if we remove the zero **DWT** coefficients with length n at level 0 , the following children nodes will be removed with length $2n$, $4n$, $8n$, $16n$ and $32n$ from level 1 to level 5 respectively. This means that we can get the proper **DWT** tree structure (every node at the previous level can be corresponded to two nodes at the next level). The proper **DWT** tree structure will be applied effectively and efficiently in

the next section.

6.3 Identification of the useful information from the pattern of small DWT tree structure

After the data reduction and the correct **DWT** tree structure is assembled (see **Section** 6.2.1), each proper **DWT** tree structure will delete zero of the **DWT** coefficients at the level 0 and its children nodes at the following levels (from level 1 to level 5) will also be removed.

6.3.1 Calculate PC score through multivariate fPCA

Each proper **DWT** tree structure will be separated into different smaller **DWT** tree structures. The information in each small **DWT** tree structure is given in **Table** 6.1. Based on the **VM** data (the signal vector magnitude of accelerometer data; see **Section** 2.2.3), the small **DWT** tree structure (2 **DWT** coefficients at level 0) corresponds to the accelerometer data with length 128. Note that we will try the different small **DWT** tree structures (1, 3, 5 and 9 **DWT** coefficients at level 0) corresponding to the accelerometer data with length 64, 192, 320 and 576; the results with different small **DWT** tree structure will be reported in **Section** 6.6. To be specific, we separate the accelerometer data into different slide windows with length 128, each slide window corresponding to a small **DWT** tree structure as shown in **Table** 6.1. Hence, each small **DWT** tree structure contains the information of slide window for about two minutes.

	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$
Scale	Scale 1 (Level 5)	Scale 2 (Level 4)	Scale 3 (Level 3)
Length	64	32	16
	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$
Scale	Scale 4 (Level 2)	Scale 5 (Level 1)	Scale 6 (Level 0)
Length	8	4	2

Table 6.1: The details in small **DWT** tree structure.

As we discussed in **Chapter** 5, the 4 new covariates PC1, PC2, x_o and x_1 will be obtained for each small **DWT** tree structure through the multivariate **fPCA**. Recalling the procedure of obtaining the small **DWT** tree structure in **Chapter** 5, we separate the **VM** data to slide windows in the first step, then apply the **DWT** and get the small tree structure through these slide windows. In contrast, in this chapter, the whole data set from each patient is applied by the **DWT** to obtain the whole **DWT** tree structure at first, then reduce the whole **DWT** tree structure into different small **DWT** tree structures.

In other words, the small **DWT** tree structures correspond to the slide windows in the entire accelerometer data. In this chapter, the length of small **DWT** tree structures at different decomposition levels is designed to be consistent with that of **Chapter** 5. The difference is that the small **DWT** tree structure corresponds to the information of 1.28 seconds and 128 seconds from designed data in **Chapter** 5 and the free-living data in this chapter respectively.

To be more specific, we transform the designed data to different slide windows based on small **DWT** tree structure in **Chapter** 5. For each slide window, it contains 4 covariates, $\mathbf{x} = [\text{PC1}, \text{PC2}, x_0, x_1]$, the 2 PC scores combines **DWT** coefficients at level 2, 3 and 4 which accounts for 99.9% of the total variance. For more details please refer to **Sections** 5.2, 5.3 and 5.3.4 in **Chapter** 5. In this chapter, after obtaining the proper **DWT** tree structure from free-living data for the j-th patient at the k-th visit in **Equation** (6.1), we separate it into different slide windows with 2 **DWT** coefficients at level 0. Then we transform it to 16 covariates $\mathbf{x} = [\text{PC1}, \text{PC2}, ..., \text{PC14}, x_0, x_1]$; the 14 PC scores combines **DWT** coefficients at level 2, 3, 4 and 5 and this accounts for about 85% total variance. Note that the free-living data is more complex than the designed data, despite the fact that **DWT** coefficients at level 5 can be treated as 'noise' which is mentioned in **Section** 5.2.1, **Chapter** 5. When we deal with the free-living data in practice, it is better to also consider the **DWT** coefficients at level 5. Moreover, the number of PCs extracted from each slide window depends on the data, and we recommend that the number of PCs from multivariate **fPCA** should accout for at least 80% information in practice.

As we discussed at the beginning of the chapter, since the special tree strcutre and large sample size in the **DWT** coefficients, it is hard to analyze the **DWT** coefficients directly. After applying the data reduction system with **NIG-MT**, slide windows and multivariate **fPCA** for the free-living data, the total 126 **DWT** coefficients from each small **DWT** tree structure in **Table** 6.1 transfrom to the 16 covariates in **Table** 6.2 among the free-living data.

	Level 0 (Scale 6)	Level 1 (Scale 5)	Level 2+3+4+5 (Scale 5, 4, 3 and 2)
Covarites	x_0	x_1	PC1, ... , PC14 scores

Table 6.2: The 16 covariates extract from free-living data in each slide window through the data reduction system (NIG-MT + slide window + multivariate fPCA).

Bear in mind, however, that our data reduction system forces about three quarters of **DWT** coefficients to zero through the **NIG-MT** at first. Then, based on the discussion in **Section** 6.2.1, after the proper **DWT** tree structure is obtained, we keep about 60% **DWT** coefficients in **Equation** (6.1) (see **Table** 4.3 in **Section** 4.5, **Chapter** 4). To be more specific, about 40% of **DWT** coefficients are forced to zero at level 0, we therefore remove the zero coefficients at the level 0 and their children nodes at the following levels.

Comparing with the raw **DWT** coefficients from free-living data, we keep about 60% sample size for it. Using the slide window and multivariate **fPCA** approches, we extract the 16 covariates from each slide window which has in total 126 **DWT** coefficients. In other words, we just keep the samples size with 16/126 (12.6%) of the remaining 60% approximately (after obtaining the proper **DWT** tree structure) of the original **DWT** coefficients. Hence, our data reduction system keeps about $60\% \times 12.6\% = 7.6\%$ of the original size of **DWT** coefficients from free-living data.

For the j-th patient at the k-th visit, based on our data reduction system, the useful information is extraced from the small **DWT** tree strucutre, and the raw **DWT** coefficients from free living data is transformed to:

$$\begin{aligned}\mathbf{x}_{(w,j,k,p)} &= [\mathrm{PC1}_{(w,j,k,p)}, \mathrm{PC2}_{(w,j,k,p)}, ..., \mathrm{PC14}_{(w,j,k,p)}, x_{0(w,j,k,p)}, x_{1(w,j,k,p)}],\\ \mathbf{x}_{(w,j,k,np)} &= [\mathrm{PC1}_{(w,j,k,np)}, \mathrm{PC2}_{(w,j,k,np)}, ..., \mathrm{PC14}_{(w,j,k,np)}, x_{0(w,j,k,np)}, x_{1(w,j,k,np)}],\end{aligned} \tag{6.2}$$

where p and np denote the paralysed side and non-paralysed side respectively, w, j and k represent the labels of slide windows, patients and visits.

6.4 Identify new features through Gaussian Mixture Models

Our primary goal is to build a predictive model for stroke rehabilitation assessment. As discussed at the beginning of the chapter, since two types of stroke patients ("can do but doesn't want to" and "couldn't do but wants to") show very different habits in their daily life (see the detailed discussion in **Section** 3.2.1 in **Chapter** 3). The task of combining both hands information properly becomes crucial. The application of Gaussian Mixture Models will be a tool for clustering the new data set in **Equation** (6.2).

6.4.1 Gaussian Mixture Models (GMM)

GMM (Bishop, 2007) is a popular clustering strategy which has been widely used due to both theoretical and computational considerations. Using the **GMM**, the transformed data from **Equation** (6.2) can be separated into homogeneous groups, which can capture the information contained in the accelerometer data. After applying multivariate **fPCA** in a **DWT** tree structure, the accelerometer data is transformed to the new data set contains 14 covariates for the patients. For convenience, **Equation** (6.2) can be re-written as:

$$\mathbf{x}_{(w,j,k,l)} = [\mathrm{PC1}_{(w,j,k,l)}, \mathrm{PC2}_{(w,j,k,l)}, ..., \mathrm{PC14}_{(w,j,k,l)}, x_{0(w,j,k,l)}, x_{1(w,j,k,l)}], \tag{6.3}$$

where w is the label for slide window, j and k are the labels for patients and visits, $l = p, np$ which corresponds to the paralysed side and non-paralysed side. Since we have, in total, 26 acute patients and 33 chronic patients and each patient has data from two hands, the new dataset includes all the patients' information.

GMM includes several Gaussian distribution, each is identified by $Z \in (1, ..., N)$, where N is the number of clusters of our dataset. Each component is Gaussian with: **(i)** a mean μ defines its centre, **(ii)** a covariance Σ defines its width, **(iii)** a mixing probability π which defines how big or small the component will be. In this section, we set 3 clusters in our dataset $\mathbf{x}_{(i,j,k)}$, and the **GMM** will be:

$$\mathbf{x}_{(w,j,k,l)} \sim \pi_1 \mathrm{N}(\mu_1, \Sigma_1) + \pi_2 \mathrm{N}(\mu_2, \Sigma_2) + \pi_3 \mathrm{N}(\mu_3, \Sigma_3). \tag{6.4}$$

Each $\mathbf{x}_{(w,j,k,l)}$ will have an indicator $\gamma_{(w,j,k,l)}$ corresponding to the cluster $Z \in (1, 2, 3)$. To this end, the optimal model will be obtained based on BIC for EM (Expectation-Maximization) initialized by hierarchical clustering for parameterized **GMM** (using the function **Mclust** in R package **mclust**). The details can be checked in Fraley & Raftery (2002), Fraley et al. (2012) and Fraley & Raftery (2007). Each component corresponds to a type of "activeness" in the slide window.

6.4.2 New clustering features

Through the **GMM**, $\mathbf{x}_{(w,j,k,l)}$ are allocated to 3 mixture components based on the indicator $\gamma_{(w,j,k,l)} = 1, 2, 3$. The information, which is merged into a single cluster, in different slide windows is considered to be potentially from the same pattern. From **Equation** (6.3), each $\mathbf{x}_{(w,j,k,l)}$ contains 16 elements, 3 clusters using **GMM** is given in **Table** 6.3.

Cluster 1	Cluster 2	Cluster 3
$\mathrm{PC1}_{(w^{c1},j,k,l)}$, $\mathrm{PC2}_{(w^{c1},j,k,l)}$	$\mathrm{PC1}_{(w^{c2},j,k,l)}$, $\mathrm{PC2}_{(w^{c2},j,k,l)}$	$\mathrm{PC1}_{(w^{c3},j,k,l)}$, $\mathrm{PC2}_{(w^{c3},j,k,l)}$
$\mathrm{PC3}_{(w^{c1},j,k,l)}$, $\mathrm{PC4}_{(w^{c1},j,k,l)}$	$\mathrm{PC3}_{(w^{c2},j,k,l)}$, $\mathrm{PC4}_{(w^{c2},j,k,l)}$	$\mathrm{PC3}_{(w^{c3},j,k,l)}$, $\mathrm{PC4}_{(w^{c3},j,k,l)}$
$\mathrm{PC5}_{(w^{c1},j,k,l)}$, $\mathrm{PC6}_{(w^{c1},j,k,l)}$	$\mathrm{PC5}_{(w^{c2},j,k,l)}$, $\mathrm{PC6}_{(w^{c2},j,k,l)}$	$\mathrm{PC5}_{(w^{c3},j,k,l)}$, $\mathrm{PC6}_{(w^{c3},j,k,l)}$
$\mathrm{PC7}_{(w^{c1},j,k,l)}$, $\mathrm{PC8}_{(w^{c1},j,k,l)}$	$\mathrm{PC7}_{(w^{c2},j,k,l)}$, $\mathrm{PC8}_{(w^{c2},j,k,l)}$	$\mathrm{PC7}_{(w^{c3},j,k,l)}$, $\mathrm{PC8}_{(w^{c3},j,k,l)}$
$\mathrm{PC9}_{(w^{c1},j,k,l)}$, $\mathrm{PC10}_{(w^{c1},j,k,l)}$	$\mathrm{PC9}_{(w^{c2},j,k,l)}$, $\mathrm{PC10}_{(w^{c2},j,k,l)}$	$\mathrm{PC9}_{(w^{c3},j,k,l)}$, $\mathrm{PC10}_{(w^{c3},j,k,l)}$
$\mathrm{PC11}_{(w^{c1},j,k,l)}$, $\mathrm{PC12}_{(w^{c1},j,k,l)}$	$\mathrm{PC11}_{(w^{c2},j,k,l)}$, $\mathrm{PC12}_{(w^{c2},j,k,l)}$	$\mathrm{PC11}_{(w^{c3},j,k,l)}$, $\mathrm{PC12}_{(w^{c3},j,k,l)}$
$\mathrm{PC13}_{(w^{c1},j,k,l)}$, $\mathrm{PC14}_{(w^{c1},j,k,l)}$	$\mathrm{PC13}_{(w^{c2},j,k,l)}$, $\mathrm{PC14}_{(w^{c2},j,k,l)}$	$\mathrm{PC13}_{(w^{c3},j,k,l)}$, $\mathrm{PC14}_{(w^{c3},j,k,l)}$
$x_{0(w^{c1},j,k,l)}$, $x_{1(w^{c1},j,k,l)}$	$x_{0(w^{c2},j,k,l)}$, $x_{1(w^{c2},j,k,l)}$	$x_{0(w^{c3},j,k,l)}$, $x_{1(w^{c3},j,k,l)}$

Table 6.3: Three clusters by using **GMM**.

Similar to looking for ratios in **Section** 3.2.1, **Chapter** 3, the information of both hands is taken into consideration. Firstly, we separate the slide window $\mathbf{x}_{(w,j,k,l)}$ with

labels of clusters $c1$ $c2$ and $c3$ for each patient. After that, the slide window $\mathbf{x}_{(w,j,k,l)}$ of the j-th patient at k-th visit can also be divided into paralysed side and non-paralysed side:

$$
\begin{aligned}
\mathbf{PC1}^{c1}_{\mathrm{p}} &= \left\|\mathrm{PC1}_{(w^{c1}\mathrm{p})}\right\|_1, \mathbf{PC1}^{c1}_{\mathrm{np}} = \left\|\mathrm{PC1}_{(w^{c1}\mathrm{np})}\right\|_1, \\
\mathbf{PC2}^{c1}_{\mathrm{p}} &= \left\|\mathrm{PC2}_{(w^{c1}\mathrm{p})}\right\|_1, \mathbf{PC2}^{c1}_{\mathrm{np}} = \left\|\mathrm{PC2}_{(w^{c1}\mathrm{np})}\right\|_1, \\
&\vdots \\
\mathbf{PC14}^{c1}_{\mathrm{p}} &= \left\|\mathrm{PC14}_{(w^{c1}\mathrm{p})}\right\|_1, \mathbf{PC14}^{c1}_{\mathrm{np}} = \left\|\mathrm{PC14}_{(w^{c1}\mathrm{np})}\right\|_1, \\
\mathbf{L0}^{c1}_{\mathrm{p}} &= \left\|x_{0(w^{c1}\mathrm{p})}\right\|_1, \mathbf{L0}^{c1}_{\mathrm{np}} = \left\|x_{0(w^{c1}\mathrm{np})}\right\|_1, \\
\mathbf{L1}^{c1}_{\mathrm{p}} &= \left\|x_{1(w^{c1}\mathrm{p})}\right\|_1, \mathbf{L1}^{c1}_{\mathrm{np}} = \left\|x_{1(w^{c1}\mathrm{np})}\right\|_1; \\
&\textbf{and} \\
\mathbf{PC1}^{c2}_{\mathrm{p}} &= \left\|\mathrm{PC1}_{(w^{c2}\mathrm{p})}\right\|_1, \mathbf{PC1}^{c2}_{\mathrm{np}} = \left\|\mathrm{PC1}_{(w^{c2}\mathrm{np})}\right\|_1, \\
\mathbf{PC2}^{c2}_{\mathrm{p}} &= \left\|\mathrm{PC2}_{(w^{c2}\mathrm{p})}\right\|_1, \mathbf{PC2}^{c2}_{\mathrm{np}} = \left\|\mathrm{PC2}_{(w^{c2}\mathrm{np})}\right\|_1, \\
&\vdots \\
\mathbf{PC14}^{c2}_{\mathrm{p}} &= \left\|\mathrm{PC14}_{(w^{c2}\mathrm{p})}\right\|_1, \mathbf{PC14}^{c2}_{\mathrm{np}} = \left\|\mathrm{PC14}_{(w^{c2}\mathrm{np})}\right\|_1, \\
\mathbf{L0}^{c2}_{\mathrm{p}} &= \left\|x_{0(w^{c2}\mathrm{p})}\right\|_1, \mathbf{L0}^{c2}_{\mathrm{np}} = \left\|x_{0(w^{c2}\mathrm{np})}\right\|_1, \\
\mathbf{L1}^{c2}_{\mathrm{p}} &= \left\|x_{1(w^{c2}\mathrm{p})}\right\|_1, \mathbf{L1}^{c2}_{\mathrm{np}} = \left\|x_{1(w^{c2}\mathrm{np})}\right\|_1; \\
&\textbf{and} \\
\mathbf{PC1}^{c3}_{\mathrm{p}} &= \left\|\mathrm{PC1}_{(w^{c3}\mathrm{p})}\right\|_1, \mathbf{PC1}^{c3}_{\mathrm{np}} = \left\|\mathrm{PC1}_{(w^{c3}\mathrm{np})}\right\|_1, \\
\mathbf{PC2}^{c3}_{\mathrm{p}} &= \left\|\mathrm{PC2}_{(w^{c3}\mathrm{p})}\right\|_1, \mathbf{PC2}^{c3}_{\mathrm{np}} = \left\|\mathrm{PC2}_{(w^{c3}\mathrm{np})}\right\|_1, \\
&\vdots \\
\mathbf{PC14}^{c3}_{\mathrm{p}} &= \left\|\mathrm{PC14}_{(w^{c3}\mathrm{p})}\right\|_1, \mathbf{PC14}^{c3}_{\mathrm{np}} = \left\|\mathrm{PC14}_{(w^{c3}\mathrm{np})}\right\|_1, \\
\mathbf{L0}^{c3}_{\mathrm{p}} &= \left\|x_{0(w^{c3}\mathrm{p})}\right\|_1, \mathbf{L0}^{c3}_{\mathrm{np}} = \left\|x_{0(w^{c3}\mathrm{np})}\right\|_1, \\
\mathbf{L1}^{c3}_{\mathrm{p}} &= \left\|x_{1(w^{c3}\mathrm{p})}\right\|_1, \mathbf{L1}^{c3}_{\mathrm{np}} = \left\|x_{1(w^{c3}\mathrm{np})}\right\|_1,
\end{aligned}
\tag{6.5}
$$

where $w^{c1}\mathrm{p}$, $w^{c2}\mathrm{p}$ and $w^{c3}\mathrm{p}$ represent the number of slide windows allocated to the clusters $c1$ $c2$ and $c3$ respectively in the paralysed side of the j-th patient at k-th visit. Similarly, $w^{c1}\mathrm{np}$, $w^{c2}\mathrm{np}$ and $w^{c3}\mathrm{np}$ represent the number of slide windows allocated to the clusters $c1$ $c2$ and $c3$ respectively in the non-paralysed side of the j-th patient at the k-th visit.

Based on **Equation** (6.5), the new feature **NPP** in cluster 1 of the j-th patient at the k-th visit is defined as follows:

$$
\begin{aligned}
\mathbf{NPP}^{c1}_{\mathrm{PC1}} &= (\mathbf{PC1}^{c1}_{\mathrm{np}} - \mathbf{PC1}^{c1}_{\mathrm{p}})/(w^{c1}\mathrm{np} + w^{c1}\mathrm{p}),\\
\mathbf{NPP}^{c1}_{\mathrm{PC2}} &= (\mathbf{PC2}^{c1}_{\mathrm{np}} - \mathbf{PC2}^{c1}_{\mathrm{p}})/(w^{c1}\mathrm{np} + w^{c1}\mathrm{p}),\\
&\vdots\\
\mathbf{NPP}^{c1}_{\mathrm{PC14}} &= (\mathbf{PC14}^{c1}_{\mathrm{np}} - \mathbf{PC14}^{c1}_{\mathrm{p}})/(w^{c1}\mathrm{np} + w^{c1}\mathrm{p}),\\
\mathbf{NPP}^{c1}_{\mathrm{L0}} &= (\mathbf{L0}^{c1}_{\mathrm{np}} - \mathbf{L0}^{c1}_{\mathrm{p}})/(w^{c1}\mathrm{np} + w^{c1}\mathrm{p}),\\
\mathbf{NPP}^{c1}_{\mathrm{L1}} &= (\mathbf{L1}^{c1}_{\mathrm{np}} - \mathbf{L1}^{c1}_{\mathrm{p}})/(w^{c1}\mathrm{np} + w^{c1}\mathrm{p}).
\end{aligned}
\tag{6.6}
$$

Similarly, the new feature **NPP** in the clusters 2 and 3 of the j-th patient at the k-th visit can also be defined:

$$
\begin{aligned}
\mathbf{NPP}^{c2}_{\mathrm{PC1}} &= (\mathbf{PC1}^{c2}_{\mathrm{np}} - \mathbf{PC1}^{c2}_{\mathrm{p}})/(w^{c2}\mathrm{np} + w^{c2}\mathrm{p}),\\
\mathbf{NPP}^{c2}_{\mathrm{PC2}} &= (\mathbf{PC2}^{c2}_{\mathrm{np}} - \mathbf{PC2}^{c2}_{\mathrm{p}})/(w^{c2}\mathrm{np} + w^{c2}\mathrm{p}),\\
&\vdots\\
\mathbf{NPP}^{c2}_{\mathrm{PC14}} &= (\mathbf{PC14}^{c2}_{\mathrm{np}} - \mathbf{PC14}^{c2}_{\mathrm{p}})/(w^{c2}\mathrm{np} + w^{c2}\mathrm{p}),\\
\mathbf{NPP}^{c2}_{\mathrm{L0}} &= (\mathbf{L0}^{c2}_{\mathrm{np}} - \mathbf{L0}^{c2}_{\mathrm{p}})/(w^{c2}\mathrm{np} + w^{c2}\mathrm{p}),\\
\mathbf{NPP}^{c2}_{\mathrm{L1}} &= (\mathbf{L1}^{c2}_{\mathrm{np}} - \mathbf{L1}^{c2}_{\mathrm{p}})/(w^{c2}\mathrm{np} + w^{c2}\mathrm{p}).
\end{aligned}
\tag{6.7}
$$

and

$$
\begin{aligned}
\mathbf{NPP}^{c3}_{\mathrm{PC1}} &= (\mathbf{PC1}^{c3}_{\mathrm{np}} - \mathbf{PC1}^{c3}_{\mathrm{p}})/(w^{c3}\mathrm{np} + w^{c3}\mathrm{p}),\\
\mathbf{NPP}^{c3}_{\mathrm{PC2}} &= (\mathbf{PC2}^{c3}_{\mathrm{np}} - \mathbf{PC2}^{c3}_{\mathrm{p}})/(w^{c3}\mathrm{np} + w^{c3}\mathrm{p}),\\
&\vdots\\
\mathbf{NPP}^{c3}_{\mathrm{PC14}} &= (\mathbf{PC14}^{c3}_{\mathrm{np}} - \mathbf{PC14}^{c3}_{\mathrm{p}})/(w^{c3}\mathrm{np} + w^{c3}\mathrm{p}),\\
\mathbf{NPP}^{c3}_{\mathrm{L0}} &= (\mathbf{L0}^{c3}_{\mathrm{np}} - \mathbf{L0}^{c3}_{\mathrm{p}})/(w^{c3}\mathrm{np} + w^{c3}\mathrm{p}),\\
\mathbf{NPP}^{c3}_{\mathrm{L1}} &= (\mathbf{L1}^{c3}_{\mathrm{np}} - \mathbf{L1}^{c3}_{\mathrm{p}})/(w^{c3}\mathrm{np} + w^{c3}\mathrm{p}).
\end{aligned}
\tag{6.8}
$$

As we have discussed before, if the slide windows are clustered into the same cluster, they will have the same pattern; the 48 new features represent the difference between the paralysed side and non-paralysed sides of each patient. To be more specific, for the same cluster, usually, if a patient uses his/her paralysed side frequently, the number of slide windows of paralysed side will be close to that of non-paralysed side, it implies that the patient has a good recovery level (i.e., a high CAHAI score). In contrast, if the number of slide windows of paralysed side is smaller than that of non-paralysed side, which means that the patient will have a low recovery level (i.e., a low CAHAI score).

6.4.3 The physical explanation among the clustering features

Until now, we have 48 new features in **Equations** (6.6), (6.7) and (6.8), based on the 3 clusters through the **GMM**, all the clustering feature sets are very similar. Now we take one set of the clustering feature ($\mathbf{NPP}^z_{PC7}$, $z = 1, 2, 3$ which represents the cluster label) for example, the new features in cluster 1 are the most distinct ones and those features in cluster 1 also correspond to the "big" movements in the slide windows (see also **Figure** 6.1). Thus, the features defined in **Equation** (6.6) indicate the ratio of performing those "big" movements between the paralysed and non-paralysed hands. For **Figure** 6.2, the value of cluster 2 is quite small, representing a "moderate" movement or activity pattern. Finally, **Figure** 6.3 shows cluster 3 mainly corresponding to the "still" or carrying out "mild" activities since the value is very close to zero. The corresponding features defined in **Equations** (6.7) and (6.8) are the ratios between two hands doing "moderate" or "mild" activities.

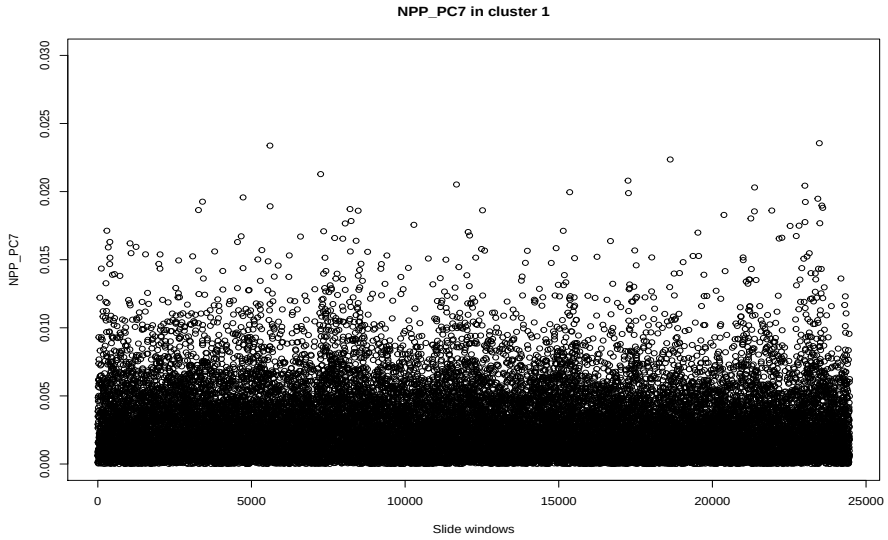


Figure 6.1: The example of clustering feature NPP_{PC7} in cluster 1.

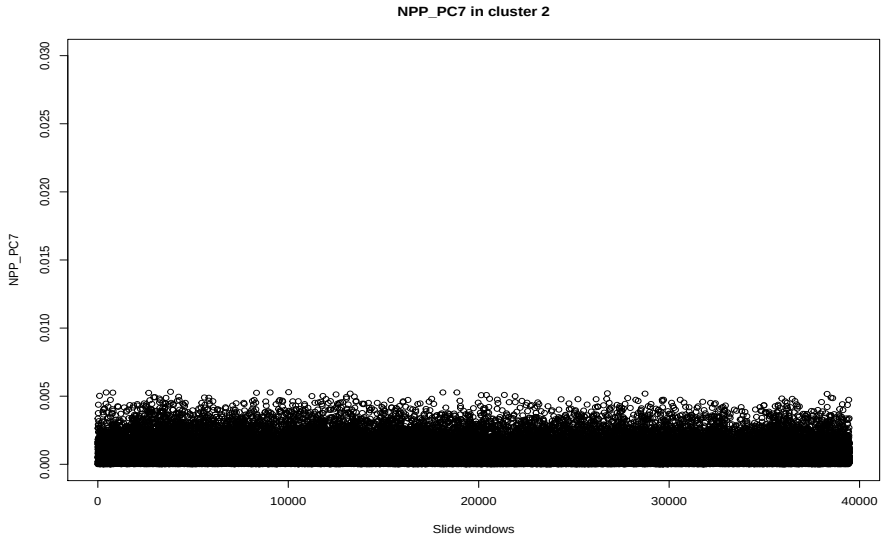


Figure 6.2: The example of clustering feature NPP_{PC7} in cluster 2.

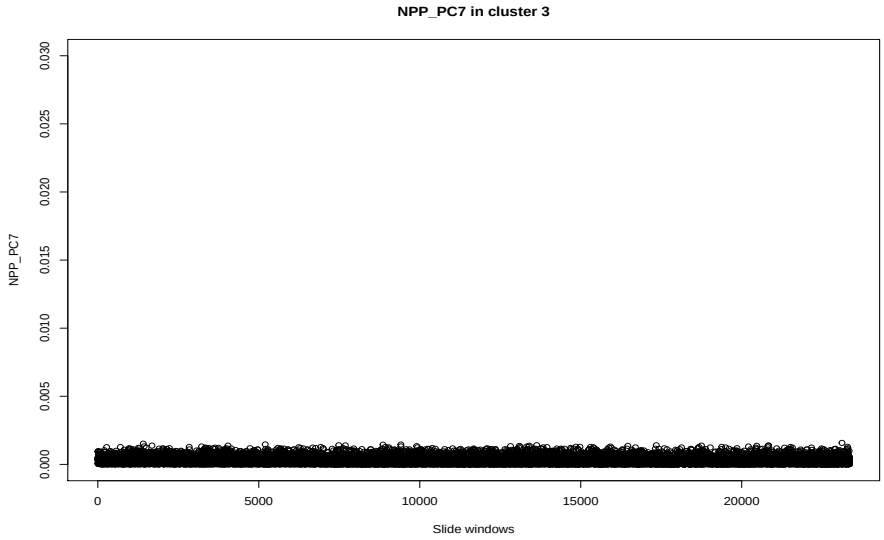


Figure 6.3: The example of clustering feature NPP_{PC7} in cluster 3.

6.5 Predictive model

After we obtain the 48 clustering features in **Section** 6.4.2, these features and the CAHAI scores from the initial visit (the medical history information) are treated as the covariates to predict the CAHAI score for the rest visits (i.e., visit 2 - visit 8). Similar to **Chapter** 3, there may exist some redundant or irrelevant features, and we use the popular strategy LASSO (see **Appendix** 3.5, **Chapter** 3) to select the useful candidate features

in our predictive model. In this chapter, we use the longitudinal mixed-effects model with Gaussian process prior (LMGP) as the predictive model which was discussed in **Section** 3.2.2, **Chapter** 3.

6.5.1 Evaluation of the predictive model

As we discussed in **Section** 2.2, **Chapter** 2, the patients were split into two groups: the acute group (26 subjects) and the chronic group (33 subjects). The two groups' information is also shown in **Figure** 2.3, **Chapter** 2 and **Figure** 3.13, **Chapter** 3. The LMGP with squared exponential kernel was used for conducted by both groups separately. Note that after the candidate features are selected by LASSO, we use all the candidate features selected by LASSO in the fixed-effects part and random-effects part consistently ($\boldsymbol{x}_{i,j} = \boldsymbol{\phi}_{i,j}$ in **Equation** (3.4), **Section** 3.2.2, **Chapter** 3) in the LMGP. Specifically for each group, leave-one-patient-out cross validation(LOPO-CV) (see **Section** 3.3, **Chapter** 3) was applied. The mean RMSE values were reported in **Figure** 6.4 for actue and chronic groups respectively. The mean RMSE's with leave-one-patient-out cross validation prediction are 5.25 and 2.96 for acute and chronic groups respectively.

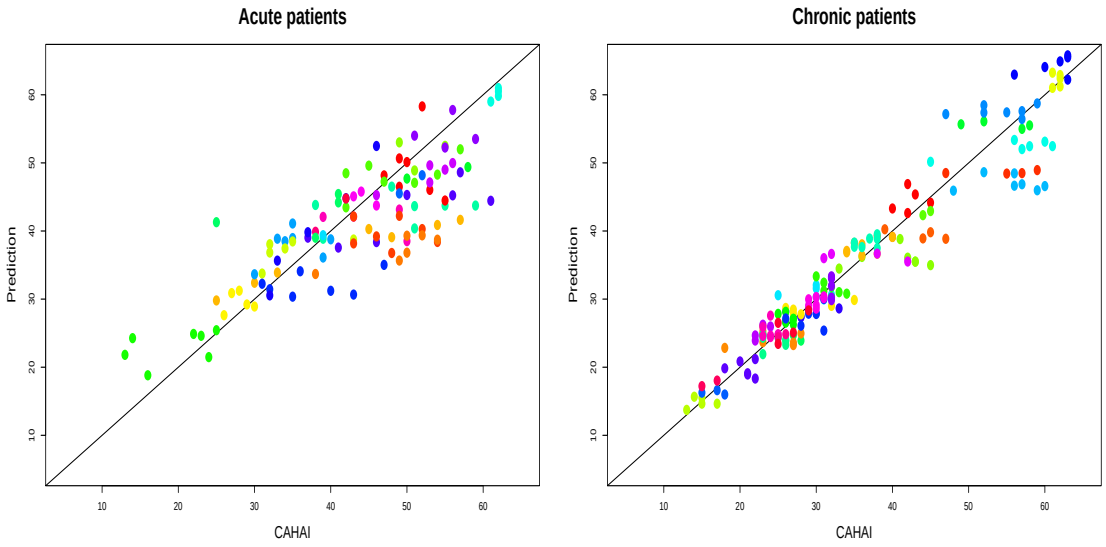


Figure 6.4: Prediction vs clinical assessed CAHAI by using leave-one-patient-out cross-validation. Left panel: RMSE is 5.25 in acute patients. Right panel: RMSE is 2.96 in chronic patients. Note that: the differnet colours present the different patients.

6.5.2 Model comparision

The model comparison is reported in **Table** 6.4 which evaluate by using leave-one-patient-out cross validation. Similar to **Chapter** 3, we treat the method of Tang et al. (2020) as the baseline for the stroke rehabilitation assessment as well. Moreover, we

also compare the results from the proposed stroke-rehab-driven features in **Chapter** 3. Furthermore, we go on to compare LMGP with support vector regression (SVR) and random forest regression(RF) for acute/chronic patients groups based on the new clustering features.

Models	RMSE (Acute)	RMSE (Chronic)
PCA + GMM + LMGP (Tang et al., 2020)	15.98	12.76
Stroke-rehab-driven + LMGP (**Chapter** 3)	5.75	3.12
Data reduction system + GMM + SVR (linear)	6.79	3.05
Data reduction system + GMM + SVR (rbf)	7.98	3.80
Data reduction system + GMM + RF	7.55	4.42
Data reduction system + GMM + LMGP	5.25	2.96

Table 6.4: Model comparison, note that: SVR (linear) and SVR (rbf) stands for support vector regression by using linear kernel and radial basis respectively; RF stands for the random forest regression.

From **Table** 6.4, we see that the new stroke rehabilitation assessment system (the data reduction system + GMM + LMGP) in this chapter gave the best performance. As we discussed in **Section** 3.3.2, the method of Tang et al. (2020) suffers from the inadequate annotations (unknown movements behind the data) and just use 1% of the training data, which leads to the unsatisfactory results in the LOPO-CV settings. For our new stroke rehabilitation assessment system, based on the wavelet-domain, the useful information is extracted from the pattern of each small **DWT** tree structure through our data reduction system. In other words, the raw signal is transformed to other types of format (14PCs, x_0 and x_1) which contains the information from both time and frequency-domains. Hence, we can see new stroke rehabilitation assessment system based on the clustering features with a much lower RMSE than the state-of-the-art data-driven approach (Tang et al., 2020). Compared to the stroke rehabilitation assessment system in **Chapter** 3, the error present in this chapter is about 0.5 lower than that before especially in the acute group. Bear in mind, we just apply 7.6% of the original size of **DWT** coefficients from free-living data, and the errors still drop by about 0.5 than when we apply the original size of **DWT** coefficients in **Chapter** 3. Based on the data reduction system and **GMM**, comparing with the performance of LMGP and other methods (linear SVR, non-linear baselines for SVR(rbf) and RF), it also shows that the LMGP can further model the longitudinal mixed-effects, making the system adaptive to different subjects/time-slots, with the lowest errors.

Moreover, this chapter provides a proper way to analyse the details within the whole **DWT** tree struture. After the data reduction system, instead of giving an analysis from coefficient to coefficient through both time and frequency-domains, we therefore give an

analysis for the information extracted from pattern to pattern in different small **DWT** tree structures. Most importantly, based on the small **DWT** tree structure which can be found in **Table** 6.1, we only keep no more than 10% of the orignial **DWT** coefficients after data reduction system. Since the patterns were designed by the size of small **DWT** tree structure (how many **DWT** coefficients are included), we will investigate the results in LMGP with the different sizes of small **DWT** tree structures in the next section.

6.6 The sensitivity of selecting different sizes of small DWT tree strucutres

From the previous sections in this chapter, based on the whole **DWT** tree structure from the free-living data, we first try to select the small **DWT** tree structure with 2 **DWT** coefficients at level 0 which can be found in **Table** 6.1. Using the approaches of slide window and multivariate **fPCA**, we aim to refine the useful information from the pattern of each slide window. In other words, the pattern can be designed by the length of slide window, if the 'proper' pattern is obtained, it will play an essential role in our final LMGP model. Hence, we investigate the sensitivity of the choice of how many **DWT** coefficients should be included in each small **DWT** tree structure. The five types of small tree structures which contain different numbers of **DWT** coefficients can be found in **Table** 6.5.

	$\mathbf{W}_1$	$\mathbf{W}_2$	$\mathbf{W}_3$
Scale	Scale 1 (Level 5)	Scale 2 (Level 4)	Scale 3 (Level 3)
Length type 1	32	16	8
Length type 2	64	32	16
Length type 3	96	48	24
Length type 4	160	80	40
Length type 5	288	144	72
	$\mathbf{W}_4$	$\mathbf{W}_5$	$\mathbf{W}_6$
Scale	Scale 4 (Level 2)	Scale 5 (Level 1)	Scale 6 (Level 0)
Length type 1	4	2	1
Length type 2	8	4	2
Length type 3	12	6	3
Length type 4	20	10	5
Length type 5	36	18	9

Table 6.5: The five types of small **DWT** tree structure.

From the **Table** 6.5, the small **DWT** tree structure from type 1 to type 5 corresponds

to the original length from 64, 128, 192, 320 and 576 respectively for free-living data (about 1 minute, 2 minutes, 3 minutes, 5 minutes and 9 miniutes for the free-living data). Moreover, each small **DWT** tree structure from type 1 to type 5 contains a total of 63, 126, 189, 315 and 567 **DWT** coefficients respectively from level 0 to level 5. As we discussed in **Section** 6.3.1, after the proper **DWT** tree structure is obtained through the **NIG-MT**, we keep about 60% sample size from the original **DWT** coefficients from free-living data. Then the slide window and multivariate **fPCA** approches are applied by using different types of small **DWT** tree structure. The small **DWT** tree structure from type 1 to type 5 transfroms to 16 variables (14 PCs, x_0 and x_1) in each slide window, which means that we just keep about 16/63, 16/126, 16/189, 16/315 and 16/567 of the sample size in the remaining 60% sample size of the orignal **DWT** coefficients. Hence, for our data reduction system (**NIG-MT**, slide window and multivariate **fPCA**) through the 5 types of small **DWT** tree structure, we keep about 15.238%, 7.619%, 5.079%, 3.048% and 1.693% of original **DWT** coefficients from the free-living data. The errors among 5 types of small **DWT** tree structure in LMGP by using LOPO-CV are reported in **Table** 6.6.

RMSE	type 1	type 2	type 3	type 4	type 5
Acute group	5.19	5.25	6.09	5.12	5.63
Chronic group	2.83	2.96	3.04	3.05	3.03

Table 6.6: The RMSEs correspond to the 5 types of small **DWT** tree structures in both acute and chronic groups.

From **Table** 6.6, the variation of errors among these 5 different small **DWT** tree structures are not sensitive in both acute and chronic groups. Although the type 4 and type 1 of small **DWT** tree structures have the lowest error in acute and chronic group respectively, the difference is ignorable.

6.7 Limitation

In this chapter, the acceptable and 'relatively good' results with very small sample size are obtained via the data reduction system, **GMM** and LMGP. However, the limitation is to use the 1-second wise **VM** data (after transformation, see **Section** 2.2.3, **Chapter** 2). Even though the sample size can be reduced when using 1-second wise **VM** data, it results in the loss of detailed information to some extent for the original accelerometer data. Hence, even when we choose the smallest small **DWT** tree structure (the level 0 just contains 1 **DWT** coefficient), the pattern still corresponds to 64 seconds (about 1 minute) in each slide window. In other words, the stroke patients can finish multiple movements during this 1 minute's period, so we just simply set the 3 clusters in the **GMM**; this is the reason why the physical explanation of clustering features just roughly allocating to "big

movements", "moderate activity" and "still or mild activites" in **Section** 6.4.3. Instead of using the **VM** data (after transfrom), we will investigate the raw **VM** data (100 points for each second) in future; it will provide the patterns with more narrow slide window (e.g., 1 second or 2 seconds) based on small **DWT** tree structure, then the setting of clusters can be large, e.g., more than 10, so that more explanation of specific activities (e.g., running, walking and so on) can be given. Similarly, going back to the variation of RMSEs among different small **DWT** tree structure in **Section** 6.6, if applying the **VM** data (not the 1-second wise format) in future, we may find more powerful and stable patterns from the narrow slide window in the time-domain, and we suppose that it will reduce the RMSEs further.

6.8 Conclusion

Unlike the designed data, the free-living accelerometer data from stroke patients has a much more complex structure. It is a challenging problem for statisticians and researchers in other areas. This chapter starts from the data reduction and, after applying **DWT**, the accelerometer data is transformed to the **DWT** coefficients. Instead of focusing on the scalar features (summary statistics information) for the wavlet-domain, we develop a data reduction system (NIG-MT, slide window and multivariate **fPCA**) based on the patterns within **DWT** coefficients in the time and frequency-domains. In other words, through the **NIG-MT** based on the **DWT** tree structure, the zero **DWT** coefficients and their children nodes have been shrunk at the level 0. Then the **DWT** coefficients can be separated into different proper small **DWT** tree structures which correspond to the slide windows in the accelerometer data. After applying the multivariate **fPCA**, the useful information from the pattern of small **DWT** tree structure can be extracted and further transformed to the smaller data set and, after applying the **GMM**, 48 new features in 3 clusters have been obtained. Then, we use the new features to measure the recovery levels among acute and chronic patients and the results are encouraging. We also find that the data reduction system has a good performance when reducing the sample size and extracting the useful information from the **DWT** coefficients in the wavelet domain. In our further study, the limitation as we discussed in **Section** 6.7, is based on the raw **VM** data, we will investigate whether we need to use a mixture model with more components and also investigate how each mixture component is associated with different types of "activity". The details will be discussed in the next chapter.

Chapter 7

Conclusion and further work

In this chapter, the overall conclusion will be given in **Section** 7.1 , and further work will be outlined in **Section** 7.2.

7.1 Conclusions and contributions

Using the discrete wavelet transform (**DWT**), the one-dimensional data was transformed to the two-dimensional **DWT** coefficients, which enables a more refined data analysis based on the wavelet-domain. Once the data was transformed to the **DWT** coefficients, a **DWT** tree structure could be found. It presented two properties: **(i)** The adjacent coefficients from the horizontal direction corresponded to time-domain, while the adjacent coefficients from vertical direction related to the frequency-domain. Especially for the data which contains human's movement, given an analysis in **DWT** tree structure, we not only investigated the data's nature with changing time, but also captured the detailed features with the changing frequency of the data. **(ii)** The value of two children **DWT** coefficients relate to its parent **DWT** coefficient at the previous decomposition level. If the parent node was far from zero, its two children nodes were tend to non-zero as well. It implied that the **DWT** tree structure in the adjacent decomposition level represented a strong correlation from the top to the bottom.

Based on the **DWT** tree structure, the Normal Inverse-Gamma Markov tree (**NIG-MT**) model provided a special posterior marginal probability $Pr(S_{j,k} = 1|\mathcal{D})$ in a hidden state. It gave an indication as to how to decide which **DWT** coefficients tended to be "noise" or "still" and could be shrunk them to zero. After the shrinking process, **DWT** coefficients were shrunk efficiently and properly in both the time and frequency-domains. Furthermore, the whole **DWT** tree structure from the entire data could be separated into different small **DWT** tree structures. From the aspect of time-domain, the small **DWT** tree structure corresponded to specific pattern in the accelerometer data. When

used in the analysis of the patterns within the accelerometer data, it provided more information concernning the movement data, especially with the challenging analysis of the free-living accelerometer data. In addition, for each small **DWT** tree structure, the multivariate functional principal component analysis (**fPCA**) could be applied to both time and frequency-domains as well. As we discussed previously, the **DWT** coefficients displayed a speical tree structure with the large data size, this makes direct analysis more difficult. The multivariate **fPCA** combined the **DWT** coefficients at several decomposition levels and extracting the useful information. In practice, the **DWT** tree structure at bottom decomposition levels was transformed to PCs since it contains a large number of **DWT** coefficients.

Through the Gaussian Mixture Models (**GMM**), the information from each small **DWT** tree structure was allocated to several clusters which presented more special cluster information. Finally, three classification models (GPC, SVM and RF) were applied in small **DWT** tree structure's information, to classify the different activities based on the designed data; it also shows that the methods of slide window and multivariate **fPCA** can be used to extract the useful information for the pattern based on the **DWT** coefficients in the acclerometer data.

In this thesis, there are two main features (wavelet features and clustering features) tailored for stroke patients: **(i)** Compared to the previous commonly used wavelet features, we define two kinds of new scalar features ($\boldsymbol{PNP}^1$ and $\boldsymbol{PNP}^2$; see **Section** 3.2.1) from **DWT** coefficients, both of which show good performance in both the linear fixed-effected model and the longitudinal mixed-effects model with a GP prior. **(ii)** The clustering features $\boldsymbol{NPP}$ (from small **DWT** tree structures by using **NIG-MT**, slide window, multivariate **fPCA** and **GMM**; see **Section** 6.4.2) show good performance as well in our longitudinal mixed-effects model for CAHAI prediction.

A data reduction sysytem with **NIG-MT**, slide window and multivariate **fPCA** has been developed, extracts the useful information for the **DWT** tree structure from the accelerometer data; it also reduces the sample size of **DWT** coefficients dramatically through the wavelet-domain. After applying the data reduction system, the extraction of useful information from wavelet-domain plays an essential role for the stroke rehabilitation assessment in **Chapter** 6.

The discussion in this thesis is limited to use the 1 second-wise **VM** data in **Chapter** 6, but there should be no significant difficulty in extending the methods to raw **VM** data and to address other related problems. The methods, based on the **DWT** tree structure, the **NIG-MT** model, the slide window, and the multivariate **fPCA** together with **GMM**, provide an approach for data reduction and regression modelling for big data with complex structure.

7.2 Further work

The extension of further work in this thesis can be separated into four parts:

(i) Chapter 5, instead of using **fPCA**, could develop goal-oriented data reduction method, e.g., functional partial least squares regression.

(ii) As we discussed in **Chapter** 6, the mixture model (Bishop, 2007) will be further investigated; this includes more than 10 mixture components based on a large data set including more patients. Afterwards, instead of giving an explanation of mixture components with "big movements", "moderate activity" and "still or mild activites", we will pay more attention to the number of specific types of "activeness" related to each mixture component. This may be used to identify most of the activities in daily life, a "hot" topic in data science.

(iii) Clustering model. This is a more challenging problem but is much more useful in practice. We can then assign the patients automatically to different clusters. The method can be extended to address many similar but difficult problems.

(iv) Going back to the 2-dimensional **DWT** tree structure, it is very similar to the 2-D image in some ways. Chan et al. (2008) present a multiscale algorithm to estimate the disparity between a pair of images; we can further investigate the slices of **DWT** tree structure in an "image" way to seek new features for patients' data.

R codes

The R codes presented in this part relate to the **NIG-MT** in **Chapter** 4. Note that the codes are not being optimised.

The R codes for NIG-MT (source code)

```
#function for DWTbayesian
##function for posteroir transition matrix
transMatrix_jk <- function(j,k){
transM <- rep()
jj <- j
if(jj==1){
transM <- matrix(c(rho_00_jlevel(1)*(phi_1k_s0[k]/ksai_1k_s0[k]),
   rho_10_jlevel(1)*(phi_1k_s0[k]/ksai_1k_s1[k]),rho_01_jlevel(1)*(
   phi_1k_s1[k]/ksai_1k_s0[k]),rho_11_jlevel(1)*(phi_1k_s1[k]/
   ksai_1k_s1[k])),2)
}else{if(jj==2){
transM <- matrix(c(rho_00_jlevel(2)*(phi_2k_s0[k]/ksai_2k_s0[k]),
   rho_10_jlevel(2)*(phi_2k_s0[k]/ksai_2k_s1[k]),rho_01_jlevel(2)*(
   phi_2k_s1[k]/ksai_2k_s0[k]),rho_11_jlevel(2)*(phi_2k_s1[k]/
   ksai_2k_s1[k])),2)
}else{if(jj==3){transM <- matrix(c(rho_00_jlevel(3)*(phi_3k_s0[k]/
   ksai_3k_s0[k]),rho_10_jlevel(3)*(phi_3k_s0[k]/ksai_3k_s1[k]),
   rho_01_jlevel(3)*(phi_3k_s1[k]/ksai_3k_s0[k]),rho_11_jlevel(3)*(
   phi_3k_s1[k]/ksai_3k_s1[k])),2)}
else{if(jj==4){transM <- matrix(c(rho_00_jlevel(4)*(phi_4k_s0[k]/
   ksai_4k_s0[k]),rho_10_jlevel(4)*(phi_4k_s0[k]/ksai_4k_s1[k]),
   rho_01_jlevel(4)*(phi_4k_s1[k]/ksai_4k_s0[k]),rho_11_jlevel(4)*(
   phi_4k_s1[k]/ksai_4k_s1[k])),2)}
else{transM <- matrix(c(rho_00_jlevel(5)*(phi_5k_s0[k]/ksai_5k_s0[k
   ]),rho_10_jlevel(5)*(phi_5k_s0[k]/ksai_5k_s1[k]),rho_01_jlevel
   (5)*(phi_5k_s1[k]/ksai_5k_s0[k]),rho_11_jlevel(5)*(phi_5k_s1[k]/
   ksai_5k_s1[k])),2)
}
```

```
}
}
}
return(transM)
}

#log-likelihood function
log_likelihood <- function(hyper_7){
v<-hyper_7[1]
sigma_0<-hyper_7[2]
alpha<-hyper_7[3]
tau<- hyper_7[4]
eta<-hyper_7[5]
gamma_h<-hyper_7[6]
rho_00_1<-hyper_7[7]


m_0k_0 <- rep(0,nrow(D_0k_ALL))
m_0k_1 <- rep(0,nrow(D_0k_ALL))
for( i in 1:nrow(D_0k_ALL)){
tau_1 <- (2^((-alpha)*0))*tau
m_0k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_0k_ALL[i,]^2))^(-v-0.5-1)
m_0k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_0k_ALL[i,]^2-1*(D_0k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}


m_1k_0 <- rep(0,nrow(D_1k_ALL))
m_1k_1 <- rep(0,nrow(D_1k_ALL))
for( i in 1:nrow(D_1k_ALL)){
tau_1 <- (2^((-alpha)*1))*tau
m_1k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_1k_ALL[i,]^2))^(-v-0.5-1)
m_1k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_1k_ALL[i,]^2-1*(D_1k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}

m_2k_0 <- rep(0,nrow(D_2k_ALL))
```

```
m_2k_1 <- rep(0,nrow(D_2k_ALL))
for( i in 1:nrow(D_2k_ALL)){
tau_1 <- (2^((-alpha)*2))*tau
m_2k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_2k_ALL[i,]^2))^(-v-0.5-1)
m_2k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_2k_ALL[i,]^2-1*(D_2k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}

m_3k_0 <- rep(0,nrow(D_3k_ALL))
m_3k_1 <- rep(0,nrow(D_3k_ALL))
for( i in 1:nrow(D_3k_ALL)){
tau_1 <- (2^((-alpha)*3))*tau
m_3k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_3k_ALL[i,]^2))^(-v-0.5-1)
m_3k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_3k_ALL[i,]^2-1*(D_3k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}

m_4k_0 <- rep(0,nrow(D_4k_ALL))
m_4k_1 <- rep(0,nrow(D_4k_ALL))
for( i in 1:nrow(D_4k_ALL)){
tau_1 <- (2^((-alpha)*4))*tau
m_4k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_4k_ALL[i,]^2))^(-v-0.5-1)
m_4k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_4k_ALL[i,]^2-1*(D_4k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}

m_5k_0 <- rep(0,nrow(D_5k_ALL))
m_5k_1 <- rep(0,nrow(D_5k_ALL))
for( i in 1:nrow(D_5k_ALL)){
tau_1 <- (2^((-alpha)*5))*tau
m_5k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_5k_ALL[i,]^2))^(-v-0.5-1)
m_5k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
```

```
    ^2+0.5*(D_5k_ALL[i,]^2-1*(D_5k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}

#################
phi_5k_s0 <- m_5k_0
phi_5k_s1 <- m_5k_1

ksai_5k_s0 <- rep(0,nrow(D_5k_ALL))
ksai_5k_s1 <- rep(0,nrow(D_5k_ALL))
for (i in 1:nrow(D_5k_ALL)){
ksai_5k_s0[i] <- max(1-eta*(2^(-5)),0)*phi_5k_s0[i] + min(eta
    *(2^(-5)),1)*phi_5k_s1[i]
ksai_5k_s1[i] <- (1-gamma_h)*phi_5k_s0[i] + gamma_h*phi_5k_s1[i]
}
##################
phi_4k_s0 <- rep(0,nrow(D_4k_ALL))
phi_4k_s1 <- rep(0,nrow(D_4k_ALL))
for (i in 1:nrow(D_4k_ALL) ){
phi_4k_s0[i] <-  m_4k_0[i]*ksai_5k_s0[2*i-1]*ksai_5k_s0[2*i]
phi_4k_s1[i] <-  m_4k_1[i]*ksai_5k_s1[2*i-1]*ksai_5k_s1[2*i]
}
##################
ksai_4k_s0 <- rep(0,nrow(D_4k_ALL))
ksai_4k_s1 <- rep(0,nrow(D_4k_ALL))
for (i in 1:nrow(D_4k_ALL)){
ksai_4k_s0[i] <- max(1-eta*(2^(-4)),0)*phi_4k_s0[i] + min(eta
    *(2^(-4)),1)*phi_4k_s1[i]
ksai_4k_s1[i] <- (1-gamma_h)*phi_4k_s0[i] + gamma_h*phi_4k_s1[i]
}
################
phi_3k_s0 <- rep(0,nrow(D_3k_ALL))
phi_3k_s1 <- rep(0,nrow(D_3k_ALL))
for (i in 1:nrow(D_3k_ALL) ){
phi_3k_s0[i] <- m_3k_0[i]*ksai_4k_s0[2*i-1]*ksai_4k_s0[2*i]
phi_3k_s1[i] <- m_3k_1[i]*ksai_4k_s1[2*i-1]*ksai_4k_s1[2*i]
}
###############
ksai_3k_s0 <- rep(0,nrow(D_3k_ALL))
ksai_3k_s1 <- rep(0,nrow(D_3k_ALL))
for (i in 1:nrow(D_3k_ALL)){
ksai_3k_s0[i] <- max(1-eta*(2^(-3)),0)*phi_3k_s0[i] + min(eta
    *(2^(-3)),1)*phi_3k_s1[i]
```

```
ksai_3k_s1[i] <- (1-gamma_h)*phi_3k_s0[i] + gamma_h*phi_3k_s1[i]
}
##############
phi_2k_s0 <- rep(0,nrow(D_2k_ALL))
phi_2k_s1 <- rep(0,nrow(D_2k_ALL))
for (i in 1:nrow(D_2k_ALL) ){
phi_2k_s0[i] <-  m_2k_0[i]*ksai_3k_s0[2*i-1]*ksai_3k_s0[2*i]
phi_2k_s1[i] <-  m_2k_1[i]*ksai_3k_s1[2*i-1]*ksai_3k_s1[2*i]
}
############
ksai_2k_s0 <- rep(0,nrow(D_2k_ALL))
ksai_2k_s1 <- rep(0,nrow(D_2k_ALL))
for (i in 1:nrow(D_2k_ALL)){
ksai_2k_s0[i] <- max(1-eta*(2^(-2)),0)*phi_2k_s0[i] + min(eta
   *(2^(-2)),1)*phi_2k_s1[i]
ksai_2k_s1[i] <- (1-gamma_h)*phi_2k_s0[i] + gamma_h*phi_2k_s1[i]
}
################
phi_1k_s0 <- rep(0,nrow(D_1k_ALL))
phi_1k_s1 <- rep(0,nrow(D_1k_ALL))
for (i in 1:nrow(D_1k_ALL) ){
phi_1k_s0[i] <-  m_1k_0[i]*ksai_2k_s0[2*i-1]*ksai_2k_s0[2*i]
phi_1k_s1[i] <-  m_1k_1[i]*ksai_2k_s1[2*i-1]*ksai_2k_s1[2*i]
}
###############
ksai_1k_s0 <- rep(0,nrow(D_1k_ALL))
ksai_1k_s1 <- rep(0,nrow(D_1k_ALL))
for (i in 1:nrow(D_1k_ALL)){
ksai_1k_s0[i] <- max(1-eta*(2^(-1)),0)*phi_1k_s0[i] + min(eta
   *(2^(-1)),1)*phi_1k_s1[i]
ksai_1k_s1[i] <- (1-gamma_h)*phi_1k_s0[i] + gamma_h*phi_1k_s1[i]
}
#################
phi_0k_s0 <- rep(0,nrow(D_0k_ALL))
phi_0k_s1 <- rep(0,nrow(D_0k_ALL))
for (i in 1:nrow(D_0k_ALL) ){
phi_0k_s0[i] <- m_0k_0[i]*ksai_1k_s0[2*i-1]*ksai_1k_s0[2*i]
phi_0k_s1[i] <- m_0k_1[i]*ksai_1k_s1[2*i-1]*ksai_1k_s1[2*i]
}
############
ksai_0k_s0 <- rep(0,nrow(D_0k_ALL))
for (i in 1:nrow(D_0k_ALL)){
ksai_0k_s0[i] <- (1-rho_00_1)*phi_0k_s0[i] + rho_00_1*phi_0k_s1[i]
```

```
}

LogLikeli<- -sum(log(ksai_0k_s0))
return(LogLikeli)
}

rho_00_jlevel <- function(j){
rho_00 <- max(1-eta*(2^(-j)),0)
return(rho_00)
}
rho_01_jlevel <- function(j){
rho_01 <-  min(eta*(2^(-j)),1)
return(rho_01)
}
rho_10_jlevel <- function(j){
rho_10 <-  1-gamma_h
return(rho_10)
}
rho_11_jlevel <- function(j){
rho_11 <-  gamma_h
return(rho_11)
}
```

The R codes for NIG-MT

```
rm(list=ls())
source(file ="C:\\R\\function_DWTbayesian20181009.R")
cd1 <-read.csv(file ="c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd1.csv",header = F)
cd2<-read.csv(file = "c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd2.csv",header = F)
cd3<-read.csv(file = "c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd3.csv",header = F)
cd4<-read.csv(file = "c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd4.csv",header = F)
cd5<-read.csv(file = "c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd5.csv",header = F)
cd6<-read.csv(file = "c:\\dwtcoefficients\\Pside\\acute\\
   la012_wk1_L.csvPcd6.csv",header = F)

signal_raw <- read.csv(file = "c:\\all1s\\Pside\\la012_wk1_L.csv",
   header = F)
```

```
signal_raw <- signal_raw$V4
signal_raw <- append(signal_raw,rep(0,2^18-length(signal_raw)))
cd1 <- cd1$V1
2^17-length(cd1)
D_5k <- append(cd1,rep(0,2^17-length(cd1)))
cd2 <- cd2$V1
D_4k <- append(cd2,rep(0,2^16-length(cd2)))
cd3 <- cd3$V1
D_3k <- append(cd3,rep(0,2^15-length(cd3)))
cd4 <- cd4$V1
D_2k <- append(cd4,rep(0,2^14-length(cd4)))
cd5 <- cd5$V1
D_1k <- append(cd5,rep(0,2^13-length(cd5)))
cd6 <- cd6$V1
D_0k <- append(cd6,rep(0,2^12-length(cd6)))

JJ <- 18
ped <- length(signal_raw)/2^JJ
for (ii in 1:1){
signal_ped <- signal_raw[((ii-1)*2^JJ+1):(ii*2^JJ)]
D_5k_ped <- D_5k[((ii-1)*2^(JJ-1)+1):(ii*2^(JJ-1))]
D_4k_ped <- D_4k[((ii-1)*2^(JJ-2)+1):(ii*2^(JJ-2))]
D_3k_ped <- D_3k[((ii-1)*2^(JJ-3)+1):(ii*2^(JJ-3))]
D_2k_ped <- D_2k[((ii-1)*2^(JJ-4)+1):(ii*2^(JJ-4))]
D_1k_ped <- D_1k[((ii-1)*2^(JJ-5)+1):(ii*2^(JJ-5))]
D_0k_ped <- D_0k[((ii-1)*2^(JJ-6)+1):(ii*2^(JJ-6))]
}

D_5k_ALL <- data.frame(D_5k_ped)
D_4k_ALL <- data.frame(D_4k_ped)
D_3k_ALL <- data.frame(D_3k_ped)
D_2k_ALL <- data.frame(D_2k_ped)
D_1k_ALL <- data.frame(D_1k_ped)
D_0k_ALL <- data.frame(D_0k_ped)

system.time({opt_hyper1 <-nlminb(c(1, 0.01, 0.5, 10, 1, 0.8, 0.5),
   log_likelihood,lower=c(0.1,0.001,0.05,0.1,0.05,0.1,0.1),upper=c
   (10,0.2,5,20,20,0.9,0.9))})

hyper_theta <- opt_hyper1$par
v=hyper_theta[1]  #hyper for inv-gamma distribution
sigma_0=hyper_theta[2] # hyper for inv-gamma distribution
alpha <- hyper_theta[3]  #
```

```
tau <- hyper_theta[4]  # control the total big or small in
   posterior of Z_jk
eta=hyper_theta[5] #hyper for transition matrix
gamma_h= hyper_theta[6] # hyper for transition matrix
########initial stale prob (rho_0k_0, rho_0k_1)
rho_00_1 = hyper_theta[7]
rho_00_0 = 1- rho_00_1
#################################
#####bottom-up pyramid algorithem
#################################
########
m_0k_0 <- rep(0,nrow(D_0k_ALL))
m_0k_1 <- rep(0,nrow(D_0k_ALL))
for( i in 1:nrow(D_0k_ALL)){
tau_1 <- (2^((-alpha)*0))*tau
m_0k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(v*sigma_0^2+0.5*(D_0k_ALL[i,]^2))^(-v-0.5-1)
m_0k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
   ^2+0.5*(D_0k_ALL[i,]^2-1*(D_0k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
   -0.5-1)
}
#########
m_1k_0 <- rep(0,nrow(D_1k_ALL))
m_1k_1 <- rep(0,nrow(D_1k_ALL))
for( i in 1:nrow(D_1k_ALL)){
tau_1 <- (2^((-alpha)*1))*tau
m_1k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(v*sigma_0^2+0.5*(D_1k_ALL[i,]^2))^(-v-0.5-1)
m_1k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
   ^2+0.5*(D_1k_ALL[i,]^2-1*(D_1k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
   -0.5-1)
}
##########
m_2k_0 <- rep(0,nrow(D_2k_ALL))
m_2k_1 <- rep(0,nrow(D_2k_ALL))
for( i in 1:nrow(D_2k_ALL)){
tau_1 <- (2^((-alpha)*2))*tau
m_2k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(v*sigma_0^2+0.5*(D_2k_ALL[i,]^2))^(-v-0.5-1)
m_2k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
   gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
```

```
    ^2+0.5*(D_2k_ALL[i,]^2-1*(D_2k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}
##########
m_3k_0 <- rep(0,nrow(D_3k_ALL))
m_3k_1 <- rep(0,nrow(D_3k_ALL))
for( i in 1:nrow(D_3k_ALL)){
tau_1 <- (2^((-alpha)*3))*tau
m_3k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_3k_ALL[i,]^2))^(-v-0.5-1)
m_3k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_3k_ALL[i,]^2-1*(D_3k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}
###########
m_4k_0 <- rep(0,nrow(D_4k_ALL))
m_4k_1 <- rep(0,nrow(D_4k_ALL))
for( i in 1:nrow(D_4k_ALL)){
tau_1 <- (2^((-alpha)*4))*tau
m_4k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_4k_ALL[i,]^2))^(-v-0.5-1)
m_4k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_4k_ALL[i,]^2-1*(D_4k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}
###########
m_5k_0 <- rep(0,nrow(D_5k_ALL))
m_5k_1 <- rep(0,nrow(D_5k_ALL))
for( i in 1:nrow(D_5k_ALL)){
tau_1 <- (2^((-alpha)*5))*tau
m_5k_0[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(v*sigma_0^2+0.5*(D_5k_ALL[i,]^2))^(-v-0.5-1)
m_5k_1[i] <- (((v*sigma_0^2)^(v+1)*gamma(v+1/2+1))/((2*pi)^0.5*
    gamma(v+1)))*(tau_1^(-1)/(1+tau_1^(-1)))^(0.5)*(v*sigma_0
    ^2+0.5*(D_5k_ALL[i,]^2-1*(D_5k_ALL[i,]^2/(1+tau_1^(-1)))))^(-v
    -0.5-1)
}
################
phi_5k_s0 <- m_5k_0
phi_5k_s1 <- m_5k_1
```

```
ksai_5k_s0 <- rep(0,nrow(D_5k_ALL))
ksai_5k_s1 <- rep(0,nrow(D_5k_ALL))
for (i in 1:nrow(D_5k_ALL)){
ksai_5k_s0[i] <- max(1-eta*(2^(-5)),0)*phi_5k_s0[i] + min(eta
   *(2^(-5)),1)*phi_5k_s1[i]
ksai_5k_s1[i] <- (1-gamma_h)*phi_5k_s0[i] + gamma_h*phi_5k_s1[i]
}
###################
phi_4k_s0 <- rep(0,nrow(D_4k_ALL))
phi_4k_s1 <- rep(0,nrow(D_4k_ALL))
for (i in 1:nrow(D_4k_ALL) ){
phi_4k_s0[i] <-  m_4k_0[i]*ksai_5k_s0[2*i-1]*ksai_5k_s0[2*i]
phi_4k_s1[i] <-  m_4k_1[i]*ksai_5k_s1[2*i-1]*ksai_5k_s1[2*i]
}
###################
ksai_4k_s0 <- rep(0,nrow(D_4k_ALL))
ksai_4k_s1 <- rep(0,nrow(D_4k_ALL))
for (i in 1:nrow(D_4k_ALL)){
ksai_4k_s0[i] <- max(1-eta*(2^(-4)),0)*phi_4k_s0[i] + min(eta
   *(2^(-4)),1)*phi_4k_s1[i]
ksai_4k_s1[i] <- (1-gamma_h)*phi_4k_s0[i] + gamma_h*phi_4k_s1[i]
}
################
phi_3k_s0 <- rep(0,nrow(D_3k_ALL))
phi_3k_s1 <- rep(0,nrow(D_3k_ALL))
for (i in 1:nrow(D_3k_ALL) ){
phi_3k_s0[i] <- m_3k_0[i]*ksai_4k_s0[2*i-1]*ksai_4k_s0[2*i]
phi_3k_s1[i] <- m_3k_1[i]*ksai_4k_s1[2*i-1]*ksai_4k_s1[2*i]
}
###############
ksai_3k_s0 <- rep(0,nrow(D_3k_ALL))
ksai_3k_s1 <- rep(0,nrow(D_3k_ALL))
for (i in 1:nrow(D_3k_ALL)){
ksai_3k_s0[i] <- max(1-eta*(2^(-3)),0)*phi_3k_s0[i] + min(eta
   *(2^(-3)),1)*phi_3k_s1[i]
ksai_3k_s1[i] <- (1-gamma_h)*phi_3k_s0[i] + gamma_h*phi_3k_s1[i]
}
#############
phi_2k_s0 <- rep(0,nrow(D_2k_ALL))
phi_2k_s1 <- rep(0,nrow(D_2k_ALL))
for (i in 1:nrow(D_2k_ALL) ){
phi_2k_s0[i] <-  m_2k_0[i]*ksai_3k_s0[2*i-1]*ksai_3k_s0[2*i]
phi_2k_s1[i] <-  m_2k_1[i]*ksai_3k_s1[2*i-1]*ksai_3k_s1[2*i]
```

```
}
############
ksai_2k_s0 <- rep(0,nrow(D_2k_ALL))
ksai_2k_s1 <- rep(0,nrow(D_2k_ALL))
for (i in 1:nrow(D_2k_ALL)){
ksai_2k_s0[i] <- max(1-eta*(2^(-2)),0)*phi_2k_s0[i] + min(eta
   *(2^(-2)),1)*phi_2k_s1[i]
ksai_2k_s1[i] <- (1-gamma_h)*phi_2k_s0[i] + gamma_h*phi_2k_s1[i]
}
################
phi_1k_s0 <- rep(0,nrow(D_1k_ALL))
phi_1k_s1 <- rep(0,nrow(D_1k_ALL))
for (i in 1:nrow(D_1k_ALL) ){
phi_1k_s0[i] <-  m_1k_0[i]*ksai_2k_s0[2*i-1]*ksai_2k_s0[2*i]
phi_1k_s1[i] <-  m_1k_1[i]*ksai_2k_s1[2*i-1]*ksai_2k_s1[2*i]
}
###############
ksai_1k_s0 <- rep(0,nrow(D_1k_ALL))
ksai_1k_s1 <- rep(0,nrow(D_1k_ALL))
for (i in 1:nrow(D_1k_ALL)){
ksai_1k_s0[i] <- max(1-eta*(2^(-1)),0)*phi_1k_s0[i] + min(eta
   *(2^(-1)),1)*phi_1k_s1[i]
ksai_1k_s1[i] <- (1-gamma_h)*phi_1k_s0[i] + gamma_h*phi_1k_s1[i]
}
##################
phi_0k_s0 <- rep(0,nrow(D_0k_ALL))
phi_0k_s1 <- rep(0,nrow(D_0k_ALL))
for (i in 1:nrow(D_0k_ALL) ){
phi_0k_s0[i] <- m_0k_0[i]*ksai_1k_s0[2*i-1]*ksai_1k_s0[2*i]
phi_0k_s1[i] <- m_0k_1[i]*ksai_1k_s1[2*i-1]*ksai_1k_s1[2*i]
}
############
ksai_0k_s0 <- rep(0,nrow(D_0k_ALL))
for (i in 1:nrow(D_0k_ALL)){
ksai_0k_s0[i] <- (1-rho_00_1)*phi_0k_s0[i] + rho_00_1*phi_0k_s1[i]
}

##########################################
######posterior of marginal prob s_jk=1
##########################################
#level 0
s_0k_0 <- rep(0,nrow(D_0k_ALL))
s_1k_1 <- rep(0,nrow(D_0k_ALL))
```

```
for (i in 1:nrow(D_0k_ALL)){
s_0k_0[i] <- rho_00_0*(phi_0k_s0[i]/ksai_0k_s0[i])
s_1k_1[i] <- rho_00_1*(phi_0k_s1[i]/ksai_0k_s0[i])
}
inistate_Pos_1 <- s_1k_1
inistate_Pos <- as.matrix(cbind(s_0k_0,s_1k_1))
#level 1
stateProb_l1 <- NULL
for (i in 1:nrow(inistate_Pos)){
TEMPstateProb_l1 <- rbind(inistate_Pos[i,]%*%transMatrix_jk(1,2*i
   -1),inistate_Pos[i,]%*%transMatrix_jk(1,2*i))
stateProb_l1 <- rbind(stateProb_l1, TEMPstateProb_l1)
}
stateProb_level1_1 <- stateProb_l1[,2]
#level 2
stateProb_l2 <- NULL
for (i in 1:nrow(stateProb_l1)){
TEMPstateProb_l2 <- rbind(stateProb_l1[i,]%*%transMatrix_jk(2,2*i
   -1),stateProb_l1[i,]%*%transMatrix_jk(2,2*i))
stateProb_l2 <- rbind(stateProb_l2, TEMPstateProb_l2)
}
stateProb_level2_1 <- stateProb_l2[,2]
#level 3
stateProb_l3 <- NULL
for (i in 1:nrow(stateProb_l2)){
TEMPstateProb_l3 <- rbind(stateProb_l2[i,]%*%transMatrix_jk(3,2*i
   -1),stateProb_l2[i,]%*%transMatrix_jk(3,2*i))
stateProb_l3 <- rbind(stateProb_l3, TEMPstateProb_l3)
}
stateProb_level3_1 <- stateProb_l3[,2]
#level 4
stateProb_l4 <- NULL
for (i in 1:nrow(stateProb_l3)){
TEMPstateProb_l4 <- rbind(stateProb_l3[i,]%*%transMatrix_jk(4,2*i
   -1),stateProb_l3[i,]%*%transMatrix_jk(4,2*i))
stateProb_l4 <- rbind(stateProb_l4, TEMPstateProb_l4)
}
stateProb_level4_1 <- stateProb_l4[,2]
#level 5
stateProb_l5 <- NULL
for (i in 1:nrow(stateProb_l4)){
TEMPstateProb_l5 <- rbind(stateProb_l4[i,]%*%transMatrix_jk(5,2*i
   -1),stateProb_l4[i,]%*%transMatrix_jk(5,2*i))
```

```
stateProb_l5 <- rbind(stateProb_l5, TEMPstateProb_l5)
}
stateProb_level5_1 <- stateProb_l5[,2]


#level 0
D_0k_sharking09 <- D_0k_ped
D_0k_sharking08 <- D_0k_ped
D_0k_sharking07 <- D_0k_ped
D_0k_sharking06 <- D_0k_ped
D_0k_sharking05 <- D_0k_ped
D_0k_sharking04 <- D_0k_ped
D_0k_sharking03 <- D_0k_ped
D_0k_sharking02 <- D_0k_ped
for (i in 1:length(D_0k_ped)){
if(inistate_Pos_1[i]< 0.9 ){
D_0k_sharking09[i] <- 0
}
if(inistate_Pos_1[i]< 0.8 ){
D_0k_sharking08[i] <- 0
}
if(inistate_Pos_1[i]< 0.7 ){
D_0k_sharking07[i] <- 0
}
if(inistate_Pos_1[i]< 0.6 ){
D_0k_sharking06[i] <- 0
}
if(inistate_Pos_1[i]< 0.5 ){
D_0k_sharking05[i] <- 0
}
if(inistate_Pos_1[i]< 0.4 ){
D_0k_sharking04[i] <- 0
}
if(inistate_Pos_1[i]< 0.3 ){
D_0k_sharking03[i] <- 0
}
if(inistate_Pos_1[i]< 0.2 ){
D_0k_sharking02[i] <- 0
}
}
#level1
D_1k_sharking09 <- D_1k_ped
D_1k_sharking08 <- D_1k_ped
```

```
D_1k_sharking07 <- D_1k_ped
D_1k_sharking06 <- D_1k_ped
D_1k_sharking05 <- D_1k_ped
D_1k_sharking04 <- D_1k_ped
D_1k_sharking03 <- D_1k_ped
D_1k_sharking02 <- D_1k_ped
for (i in 1:length(D_1k_ped)){
if(stateProb_level1_1[i]< 0.9 ){
D_1k_sharking09[i] <- 0
}
if(stateProb_level1_1[i]< 0.8 ){
D_1k_sharking08[i] <- 0
}
if(stateProb_level1_1[i]< 0.7 ){
D_1k_sharking07[i] <- 0
}
if(stateProb_level1_1[i]< 0.6 ){
D_1k_sharking06[i] <- 0
}
if(stateProb_level1_1[i]< 0.5 ){
D_1k_sharking05[i] <- 0
}
if(stateProb_level1_1[i]< 0.4 ){
D_1k_sharking04[i] <- 0
}
if(stateProb_level1_1[i]< 0.3 ){
D_1k_sharking03[i] <- 0
}
if(stateProb_level1_1[i]< 0.2 ){
D_1k_sharking02[i] <- 0
}
}
#level 2
D_2k_sharking09 <- D_2k_ped
D_2k_sharking08 <- D_2k_ped
D_2k_sharking07 <- D_2k_ped
D_2k_sharking06 <- D_2k_ped
D_2k_sharking05 <- D_2k_ped
D_2k_sharking04 <- D_2k_ped
D_2k_sharking03 <- D_2k_ped
D_2k_sharking02 <- D_2k_ped
for (i in 1:length(D_2k_ped)){
if(stateProb_level2_1[i]< 0.9 ){
```

```
D_2k_sharking09[i] <- 0
}
if(stateProb_level2_1[i]< 0.8 ){
D_2k_sharking08[i] <- 0
}
if(stateProb_level2_1[i]< 0.7 ){
D_2k_sharking07[i] <- 0
}
if(stateProb_level2_1[i]< 0.6 ){
D_2k_sharking06[i] <- 0
}
if(stateProb_level2_1[i]< 0.5 ){
D_2k_sharking05[i] <- 0
}
if(stateProb_level2_1[i]< 0.4 ){
D_2k_sharking04[i] <- 0
}
if(stateProb_level2_1[i]< 0.3 ){
D_2k_sharking03[i] <- 0
}
if(stateProb_level2_1[i]< 0.2 ){
D_2k_sharking02[i] <- 0
}
}
#level 3
D_3k_sharking09 <- D_3k_ped
D_3k_sharking08 <- D_3k_ped
D_3k_sharking07 <- D_3k_ped
D_3k_sharking06 <- D_3k_ped
D_3k_sharking05 <- D_3k_ped
D_3k_sharking04 <- D_3k_ped
D_3k_sharking03 <- D_3k_ped
D_3k_sharking02 <- D_3k_ped
for (i in 1:length(D_3k_ped)){
if(stateProb_level3_1[i]< 0.9 ){
D_3k_sharking09[i] <- 0
}
if(stateProb_level3_1[i]< 0.8 ){
D_3k_sharking08[i] <- 0
}
if(stateProb_level3_1[i]< 0.7 ){
D_3k_sharking07[i] <- 0
}
```

```
if(stateProb_level3_1[i]< 0.6 ){
D_3k_sharking06[i] <- 0
}
if(stateProb_level3_1[i]< 0.5 ){
D_3k_sharking05[i] <- 0
}
if(stateProb_level3_1[i]< 0.4 ){
D_3k_sharking04[i] <- 0
}
if(stateProb_level3_1[i]< 0.3 ){
D_3k_sharking03[i] <- 0
}
if(stateProb_level3_1[i]< 0.2 ){
D_3k_sharking02[i] <- 0
}
}
#level 4
D_4k_sharking09 <- D_4k_ped
D_4k_sharking08 <- D_4k_ped
D_4k_sharking07 <- D_4k_ped
D_4k_sharking06 <- D_4k_ped
D_4k_sharking05 <- D_4k_ped
D_4k_sharking04 <- D_4k_ped
D_4k_sharking03 <- D_4k_ped
D_4k_sharking02 <- D_4k_ped
for (i in 1:length(D_4k_ped)){
if(stateProb_level4_1[i]< 0.9 ){
D_4k_sharking09[i] <- 0
}
if(stateProb_level4_1[i]< 0.8 ){
D_4k_sharking08[i] <- 0
}
if(stateProb_level4_1[i]< 0.7 ){
D_4k_sharking07[i] <- 0
}
if(stateProb_level4_1[i]< 0.6 ){
D_4k_sharking06[i] <- 0
}
if(stateProb_level4_1[i]< 0.5 ){
D_4k_sharking05[i] <- 0
}
if(stateProb_level4_1[i]< 0.4 ){
D_4k_sharking04[i] <- 0
```

```
}
if(stateProb_level4_1[i]< 0.3 ){
D_4k_sharking03[i] <- 0
}
if(stateProb_level4_1[i]< 0.2 ){
D_4k_sharking02[i] <- 0
}
}
#level 5
D_5k_sharking09 <- D_5k_ped
D_5k_sharking08 <- D_5k_ped
D_5k_sharking07 <- D_5k_ped
D_5k_sharking06 <- D_5k_ped
D_5k_sharking05 <- D_5k_ped
D_5k_sharking04 <- D_5k_ped
D_5k_sharking03 <- D_5k_ped
D_5k_sharking02 <- D_5k_ped
for (i in 1:length(D_5k_ped)){
if(stateProb_level5_1[i]< 0.9 ){
D_5k_sharking09[i] <- 0
}
if(stateProb_level5_1[i]< 0.8 ){
D_5k_sharking08[i] <- 0
}
if(stateProb_level5_1[i]< 0.7 ){
D_5k_sharking07[i] <- 0
}
if(stateProb_level5_1[i]< 0.6 ){
D_5k_sharking06[i] <- 0
}
if(stateProb_level5_1[i]< 0.5 ){
D_5k_sharking05[i] <- 0
}
if(stateProb_level5_1[i]< 0.4 ){
D_5k_sharking04[i] <- 0
}
if(stateProb_level5_1[i]< 0.3 ){
D_5k_sharking03[i] <- 0
}
if(stateProb_level5_1[i]< 0.2 ){
D_5k_sharking02[i] <- 0
}}
```

Bibliography

1999 Stroke: Neurologic and functional recovery the copenhagen stroke study. Physical Medicine and Rehabilitation Clinics of North America **10** (4), 887 – 906, a New Century Approach to Stroke Management and Rehabilitation.

Abramovich, F., Sapatinas, T. & Silverman, B. W. 1998 Wavelet thresholding via a Bayesian approach. Journal of the Royal Statistical Society. Series B (Statistical Methodology) **60** (4), 725–749.

Axivity Ltd 2013 AX3, 3-Axis Logging Accelerometer. `http://axivity.com/v2/products/AX3/AX3-1.7-Datasheet.pdf`, [Online; accessed July-2013].

Ayachi, F. S., Nguyen, H. P., Lavigne-Pelletier, C., Goubault, E., Boissy, P. & Duval, C. 2016 Wavelet-based algorithm for auto-detection of daily living activities of older adults captured by multiple inertial measurement units (imus). Physiological measurement **37** (3), 442?461.

Bai, Y., Guan, Y. & Ng, W.-F. 2020 Fatigue assessment using ecg and actigraphy sensors. In Proceedings of the 24rd International Symposium on Wearable Computers. New York, NY, USA: Association for Computing Machinery.

Baker, J. W. 2007 Quantitative Classification of Near-Fault Ground Motions Using Wavelet Analysis. Bulletin of the Seismological Society of America **97** (5), 1486–1501.

Barreca, S., Stratford, P., Lambert, C., Masters, L. & Streiner, D. 2005a Test-Retest Reliability, Validity, and Sensitivity of the Chedoke Arm and Hand Activity Inventory: a New Measure of Upper-Limb Function for Survivors of Stroke. Arch Phys Med Rehabil **86**, 1616–1622.

Barreca, S., Stratford, P., Masters, L., Lambert, C., Griffiths, J. & McBay, C. 2006 Validation of Three Shortened Versions of the Chedoke Arm and Hand Activity Inventory. Physiother. Can. **58**, 1–9.

Barreca, S. R., Stratford, P. W., Lambert, C. L., Masters, L. M. & Streiner, D. L. 2005b Test-retest reliability, validity, and sensitivity of the chedoke arm and

hand activity inventory: a new measure of upper-limb function for survivors of stroke. Archives of physical medicine and rehabilitation **86 8**, 1616–22.

Bishop, C. M. 2007 Pattern Recognition and Machine Learning (Information Science and Statistics), 1st edn. Springer.

Bobin, M., Bimbard, F., Boukallel, M., Anastassova, M. & Ammi, M. 2019 Spectrum: Smart ecosystem for stroke patient?s upper limbs monitoring. Smart Health **13**, 100066.

Bouten, C., Koekkoek, K., Verduin, M., Kodde, R. & Janssen, J. 1997 A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity. IEEE transactions on bio-medical engineering **44**, 136–47.

Chan, W. L., Choi, H. & Baraniuk, R. G. 2008 Coherent multiscale image processing using dual-tree quaternion wavelets. IEEE Transactions on Image Processing **17** (7), 1069–1082.

Chipman, H. A., Kolacxyk, E. D. & McCulloch, R. E. 1996 Signal de-noising using adaptive Bayesian wavelet shrinkage. In Proceedings of Third International Symposium on Time-Frequency and Time-Scale Analysis (TFTS-96), pp. 225–228.

Chipman, H. A., Kolaczyk, E. D. & McCulloch, R. E. 1997 Adaptive Bayesian wavelet shrinkage. Journal of the American Statistical Association **92** (440), 1413–1421.

Clyde, M. & George, E. I. 2000 Flexible empirical Bayes estimation for wavelets. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **62** (4), 681–698.

CLYDE, M., PARMIGIANI, G. & VIDAKOVIC, B. 1998 Multiple shrinkage and subset selection in wavelets. Biometrika **85** (2), 391–401.

Daubechies, I. 2006 Orthonormal bases of compactly supported wavelets. Commun. Pure. Appl. Math pp. 909–996.

Doherty, A., Jackson, D., Hammerla, N., Plötz, T., Olivier, P., Granat, M. H., White, T., van Hees, V. T., Trenell, M. I., Owen, C. G., Preece, S. J., Gillions, R., Sheard, S., Peakman, T., Brage, S. & Wareham, N. J. 2017 Large scale population assessment of physical activity using wrist worn accelerometers: The uk biobank study. PLOS ONE **12** (2), 1–14.

Dolatabadi, E., Zhi, Y., Ye, B., Coahran, M., Lupinacci, G., Mihailidis, A., Wang, R. & Taati, B. 2017 The toronto rehab stroke pose dataset to detect compensation during stroke rehabilitation therapy. pp. 375–381.

Donald B. Percival, A. T. W. 2000 Wavelet methods for time series analysis, 1st edn. Cambridge University Press.

Donnan, G., Fisher, M., Macleod, M. & Davis, S. 2008 Stroke. Lancet **371** (2), 1612–1623.

Donoho, D. L. & Johnstone, I. M. 1994 Ideal spatial adaptation by wavelet shrinkage. Biometrika **81** (3), 425–455.

Ferrari, P., Friedenreich, C. & Matthews, C. 2007 The role of measurement error in estimating levels of physical activity. American journal of epidemiology **166**, 832–40.

Fraley, C. & Raftery, A. E. 2002 Model-based clustering, discriminant analysis, and density estimation. Journal of the American Statistical Association **97** (458), 611–631.

Fraley, C. & Raftery, A. E. 2007 Bayesian regularization for normal mixture estimation and model-based clustering. Journal of Classification **24** (2), 155–181.

Fraley, C., Raftery, A. E., Murphy, T. B. & Scrucca, L. 2012 mclust version 4 for r: Normal mixture modeling for model-based clustering, classification, and density estimation.

Ganesh, A. C., Renganathan, B. S., Rajakumaran, C., Preejith, S. P., Shubham, K., Jayaraj, J. & Mohanasankar, S. 2018 Post-stroke rehabilitation monitoring using wireless surface electromyography: A case study. In 2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA), pp. 1–6.

Gao, Y., Long, Y., Guan, Y., Basu, A., Baggaley, J. & Ploetz, T. 2019 Towards reliable, automated general movement assessment for perinatal stroke screening in infants using wearable accelerometers. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **3** (1).

Guan, Y. & Plötz, T. 2017 Ensembles of deep lstm learners for activity recognition using wearables. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1** (2).

Gurchiek, R., Choquette, R., Beynnon, B., Slauterbeck, J., Tourville, T., Toth, M. & McGinnis, R. 2019 Open-source remote gait analysis: A post-surgery patient monitoring application. Scientific reports **9**, 17966.

Hammerla, N. Y., Fisher, J. M., Andras, P., Rochester, L., Walker, R. & Plotz, T. 2015 Pd disease state assessment in naturalistic environments using deep learning. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, p. 1742?1748. AAAI Press.

Ibrahim, A. A., K?derle, A., Ga?ner, H., Klucken, J., Eskofier, B. M. & Kluge, F. 2020 Inertial sensor-based gait parameters reflect patient-reported fatigue in multiple sclerosis. Journal of neuroengineering and rehabilitation **17** (1), 165.

Jung, H., Park, J., Jeong, J., Ryu, T., Kim, Y. & Lee, S. I. 2018 A wearable monitoring system for at-home stroke rehabilitation exercises: A preliminary study. In 2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), pp. 13–16.

Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H. & Celler, B. G. 2006 Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. IEEE Transactions on Information Technology in Biomedicine **10** (1), 156–167.

Little, B., Alshabrawy, O., Stow, D., Ferrier, I., McNaney, R., Jackson, D., Ladha, K., Ladha, C., Ploetz, T., Bacardit, J., Olivier, P., Gallagher, P. & O'Brien, J. 2020 Deep learning-based automated speech detection as a marker of social functioning in late-life depression. Psychological Medicine pp. 1–10.

Ma, L. & Soriano, J. 2017 Efficient functional ANOVA through wavelet-domain Markov groves. Journal of the American Statistical Association **0** (ja), 0–0.

Mallat, S. G. 1989 A theory for multiresolution signal decomposition: the wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (7), 674–693.

Nguyen, A., Moore, D. & McCowan, I. 2007 Unsupervised clustering of free-living human activities using ambulatory accelerometry. In 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4895–4898.

Plotz, T. & Guan, Y. 2018 Deep learning for human activity recognition in mobile computing. Computer **51** (5), 50–59.

Plötz, T., Hammerla, N. Y., Rozga, A., Reavis, A., Call, N. & Abowd, G. D. 2012 Automatic assessment of problem behavior in individuals with developmental disabilities. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, p. 391?400. New York, NY, USA: Association for Computing Machinery.

Preece, S. J., Goulermas, J. Y., Kenney, L. P. J. & Howard, D. 2009 A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE Transactions on Biomedical Engineering **56** (3), 871–879.

Ramsay, J. O. & Silverman, B. 2005 Functional Data Analysis, 2nd edn. Springer.

Ratcliffe, A., Zhai, B., Guan, Y., Jackson, D., SWARM & Sneyd, J. 2020 Patient-centred measurement of recovery from day-case surgery using wrist worn accelerometers: a pilot and feasibility study. Anaesthesia .

Rehman, R. z. u., Din, S., Guan, Y., Yarnall, A., Shi, J. & Rochester, L. 2019 Selecting clinically relevant gait characteristics for classification of early parkinson's disease: A comprehensive machine learning approach. Scientific Reports **9**.

Sekine, M., Tamura, T., Ogawa, M., Togawa, T. & Fukui, Y. 1998 Classification of acceleration waveform in a continuous walking record. In Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), , vol. 3, pp. 1523–1526 vol.3.

Shi, J., Cheng, Y., Serradilla, J., Morgan, G., Lambden, C., Ford, G., Price, C., Rodgers, H., Cassidy, T., Rochester, L. & Eyre, J. 2013 Evaluating Functional Ability of Upper Limbs after Stroke Using Video Game Data. In International Conference on Brain and Health Informatics (ed. K. Imamura, S. Usui, T. Shirao, T. Kasamatsu, L. Schwabe & N. Zhong), Lecture Notes in Artificial Intelligence, vol. 8211, pp. 181–192. Springer.

Shi, J. & Choi, T. 2011 Gaussian Process Regression Analysis for Functional Data, 1st edn. Chapman and Hall/CRC.

Shi, J., Wang, B., Will, E. & West, R. 2012 Mixed-effects Gaussian process functional regression models with application to dose response curve prediction. Statistics in Medicine **31** (26), 3165–3177.

Siirtola, P., Laurinen, P., Haapalainen, E., Roning, J. & Kinnunen, H. 2009 Clustering-based activity classification with a wrist-worn accelerometer using basic features. In 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp. 95–100.

Supratak, A., Dong, H., Wu, C. & Guo, Y. 2017 Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg. IEEE Transactions on Neural Systems and Rehabilitation Engineering **PP**.

Tang, L., Halloran, S., Shi, J. Q., Guan, Y., Cao, C. & Eyre, J. 2020 Evaluating upper limb function after stroke using the free-living accelerometer data. Statistical Methods in Medical Research .

Tibshirani, R. 1996 Regression shrinkage and selection via the LASSO. Journal of the Royal Statistical Society, Series B **58** (1), 267–288.

Van Kuppevelt, D., Heywood, J., Hamer, M., Sabia, S., Fitzsimons, E. & van Hees, V. 2018 Segmenting accelerometer data from daily life with unsupervised machine learning. bioRxiv .

Wintermark, M., Sesay, M., Barbier, E., Borbaly, K., Dillon, W. P., Eastwood, J. D., Glenn, T. C., Grandin, C. B., Pedraza, S., Soustiel, J.-F., Nariai, T., Zaharchuk, G., Cailla, J.-M., Dousset, V. & Yonas, H. 2005 Comparative overview of brain perfusion imaging techniques. Stroke **36** (9), e83–e99.

Yin, S. & Huang, Z. 2015 Performance monitoring for vehicle suspension system via fuzzy positivistic c-means clustering based on accelerometer measurements. IEEE/ASME Transactions on Mechatronics **20** (5), 2613–2620.

Zhai, B., Perez-Pozuelo, I., Clifton, E. A. D., Palotti, J. & Guan, Y. 2020 Making sense of sleep: Multimodal sleep stage classification in a large, diverse population using movement and cardiac sensing. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **4** (2).