



## BE (IoTE) Degree Program

### Stage 4 Project Group Thesis Receipt

Project Title: Intelligent multi-modal AI data fusion for wearable sensors  
Supervisor: Dr Deepu John

| Student Name(s) | UCD Student Number | UCD Student Number |
|-----------------|--------------------|--------------------|
| Ke Qi Wei       | 18206387           |                    |
| Xie Zhi Peng    | 18206103           |                    |
| Wang Xi Ting    | 18206392           |                    |
| Zhang Tian Long | 18206089           |                    |
|                 |                    |                    |
|                 |                    |                    |

**Plagiarism:** the unacknowledged inclusion of another person's writings or ideas or formally presented work (including essays, examinations, projects, laboratory presentations). The penalties associated with plagiarism designed to impose sanctions seriousness of University's commitment to academic integrity. Ensure that you have read the University's **Briefing for Students on Academic Integrity and Plagiarism** and the UCD **Statement, Plagiarism Policy and Procedures**, (<http://www.ucd.ie/registrar/>)

#### Declaration of Authorship

I/we declare that all of the following are true:

- 1) I/we fully understand the definition of plagiarism.
- 2) I/we have not plagiarised any part of this project and it is my original work.
- 3) All material in this report is my/our own work except where there is clear acknowledgement and appropriate reference to the work of others.

Signed.....柯其蔚..... Date ..... 2022.5.20.....  
Signed.....谢智鹏..... Date ..... 2022.5.20.....  
Signed.....王曦婷..... Date ..... 2022.5.20.....  
Signed.....张天龙..... Date ..... 2022.5.20.....  
Signed..... Date .....  
Signed..... Date .....  
Signed..... Date .....

---

Office Use Only

|                                                |
|------------------------------------------------|
| Date and Time Received:<br><br><br><br><br>N/A |
|------------------------------------------------|

|                                     |
|-------------------------------------|
| Received by: Brightspace Submission |
|-------------------------------------|

Report Tracking

|  |
|--|
|  |
|  |
|  |
|  |

# Final Year Project

---

## Intelligent multi-modal AI data fusion for wearable sensors

Qiwei Ke, Tianlong Zhang, Xiting Wang ,Zhipeng Xie

---

Student ID: 18206387, 18206089, 18206392, 18206103

---

A thesis submitted in part fulfillment of the degree of

**BSc in Internet of Things (IoT)**

**Supervisor:** Dr Deepu John



Beijing-Dublin International College  
University College Dublin

May 19, 2022

---

# Table of Contents

---

|          |                                                        |    |
|----------|--------------------------------------------------------|----|
| <b>1</b> | <b>Project Specification</b>                           | 6  |
| <b>2</b> | <b>Introduction</b>                                    | 7  |
| <b>3</b> | <b>Literature Review</b>                               | 8  |
| 3.1      | Feature Extraction                                     | 8  |
| 3.2      | Signal Quality Indicators                              | 9  |
| 3.3      | Machine Learning And Data Fusion                       | 9  |
| <b>4</b> | <b>Work Division</b>                                   | 10 |
| <b>5</b> | <b>Feature Extraction</b>                              | 11 |
| 5.1      | Introduction                                           | 11 |
| 5.2      | Database Selection                                     | 11 |
| 5.3      | Feature Extraction implementation A                    | 12 |
| 5.4      | Feature Extraction Implementation B                    | 20 |
| 5.5      | Conclusion                                             | 28 |
| <b>6</b> | <b>SQI calculation</b>                                 | 29 |
| 6.1      | Introduction                                           | 29 |
| 6.2      | Methodology                                            | 29 |
| 6.3      | Discussion                                             | 37 |
| <b>7</b> | <b>Data Fusion Using Machine Learning</b>              | 41 |
| 7.1      | Introduction                                           | 41 |
| 7.2      | Traditional method of sensor signal fusion             | 41 |
| 7.3      | Proposal solutions                                     | 42 |
| 7.4      | Convolution neural network(CNN)                        | 42 |
| 7.5      | Fusion method used in the project – SVR                | 45 |
| 7.6      | Data pre-processing                                    | 46 |
| 7.7      | The first fusion output                                | 47 |
| 7.8      | First attempt to improve – Wavelet threshold denoising | 47 |

---

|           |                                                                    |    |
|-----------|--------------------------------------------------------------------|----|
| 7.9       | Second attempt to improve – Using the traditional method . . . . . | 48 |
| 7.10      | Reflection on the feature extraction . . . . .                     | 49 |
| 7.11      | Final attempt . . . . .                                            | 49 |
| 7.12      | Unpredictable problems . . . . .                                   | 50 |
| 7.13      | Conclusion . . . . .                                               | 51 |
| <b>8</b>  | <b>Breath Rate Detection And Data Processing</b> . . . . .         | 52 |
| 8.1       | Introduction . . . . .                                             | 52 |
| 8.2       | Breath Number Detection . . . . .                                  | 52 |
| 8.3       | Data Format Conversion . . . . .                                   | 53 |
| <b>9</b>  | <b>Future Work</b> . . . . .                                       | 54 |
| <b>10</b> | <b>Conclusion</b> . . . . .                                        | 55 |

---

# List of Figures

---

|      |                                                                                                                   |    |
|------|-------------------------------------------------------------------------------------------------------------------|----|
| 1.1  | Project Background . . . . .                                                                                      | 6  |
| 2.1  | Project flow chart . . . . .                                                                                      | 7  |
| 5.1  | CapnoBase Exemplary signal [19] . . . . .                                                                         | 11 |
| 5.2  | Fiducial points for AM [20] . . . . .                                                                             | 12 |
| 5.3  | PPG and ECG AM . . . . .                                                                                          | 13 |
| 5.4  | Fiducial points for BW [20] . . . . .                                                                             | 13 |
| 5.5  | PPG and ECG BW . . . . .                                                                                          | 14 |
| 5.6  | Fiducial points for FM [20] . . . . .                                                                             | 14 |
| 5.7  | PPG and ECG FM . . . . .                                                                                          | 14 |
| 5.8  | Zoomed version of ECG signal with data marks . . . . .                                                            | 15 |
| 5.9  | Zoomed version of PPG signal . . . . .                                                                            | 15 |
| 5.10 | Exemplary normalized signal . . . . .                                                                             | 16 |
| 5.11 | Different interpolated signals . . . . .                                                                          | 17 |
| 5.12 | Exemplary ECG signal with noisy start . . . . .                                                                   | 18 |
| 5.13 | Flow chart of the proposed system . . . . .                                                                       | 20 |
| 5.14 | Extraction AM signal . . . . .                                                                                    | 21 |
| 5.15 | Get HRV and PRV signal from AM signal . . . . .                                                                   | 21 |
| 5.16 | Noise HRV/PRV signal(top),IMF1(second of top),IMF2(third of top),IMF3(fourth of top),IMF4(bottom) . . . . .       | 23 |
| 5.17 | Noise ECG-AM/PPG-AM signal(top),IMF1(second of top),IMF2(third of top),IMF3(fourth of top),IMF4(bottom) . . . . . | 24 |
| 5.18 | Reference Respiration Signal . . . . .                                                                            | 24 |
| 5.19 | HRV(IMF2+IMF3) and PRV(IMF2+IMF3) signal at 480s . . . . .                                                        | 26 |
| 5.20 | ECG-AM(IMF2+IMF3) and PPG-AM(IMF2+IMF3) signal at 480s . . . . .                                                  | 26 |
| 5.21 | Mixed-ECG and Mixed-PPG signal. The upper one is Mixed-ECG, the bottom one is Mixed-PPG . . . . .                 | 27 |

---

|      |                                                                                                                    |    |
|------|--------------------------------------------------------------------------------------------------------------------|----|
| 6.1  | Calculation process for one segment of ECG/PPG . . . . .                                                           | 30 |
| 6.2  | Process for template generation for one segment . . . . .                                                          | 30 |
| 6.3  | Optimization process of detection . . . . .                                                                        | 31 |
| 6.4  | Process for template generation for one segment . . . . .                                                          | 31 |
| 6.5  | Average SQI before normalization with different SNR . . . . .                                                      | 32 |
| 6.6  | The flowchart of one proposed algorithm for data fusion[5] . . . . .                                               | 33 |
| 6.7  | SQI for ECG and PPG signals with noise partly added . . . . .                                                      | 33 |
| 6.8  | Comparisons of the quality of extracted signal labelled with different SQI . . . . .                               | 34 |
| 6.9  | SQI with different SNR . . . . .                                                                                   | 34 |
| 6.10 | The flowchart for optimization calculation . . . . .                                                               | 36 |
| 6.11 | An example for the problem in R-peak detection . . . . .                                                           | 38 |
| 6.12 | Expected curve for SNR and SQI(method 2) . . . . .                                                                 | 38 |
| 6.13 | Choosing ECG template . . . . .                                                                                    | 39 |
| 6.14 | Choosing PPG template . . . . .                                                                                    | 39 |
| 6.15 | Scatter plot for SQI analysis . . . . .                                                                            | 40 |
| 7.1  | The flowchart of fusion . . . . .                                                                                  | 41 |
| 7.2  | The principle of convolution layer in 2-D model . . . . .                                                          | 43 |
| 7.3  | Padding operation . . . . .                                                                                        | 43 |
| 7.4  | The principle of two pooling layer . . . . .                                                                       | 44 |
| 7.5  | SVR principle . . . . .                                                                                            | 45 |
| 7.6  | An example quadratic function . . . . .                                                                            | 46 |
| 7.7  | SVR principle – from 2-D to 3-D . . . . .                                                                          | 46 |
| 7.8  | Code to fill mean value . . . . .                                                                                  | 46 |
| 7.9  | Code to fill -1 . . . . .                                                                                          | 46 |
| 7.10 | Wavelet decomposing, where X is input, ca is low frequency information, cd is high frequency information . . . . . | 48 |
| 7.11 | Two curves stands for features of ECG and PPG, most of the peaks are staggered . . . . .                           | 49 |
| 7.12 | The scatter diagram for SQI-1 . . . . .                                                                            | 50 |
| 7.13 | The scatter diagram for SQI-3 . . . . .                                                                            | 50 |
| 8.1  | An example of three point detect . . . . .                                                                         | 53 |

---

# List of Tables

---

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.1  | Interpolation method description . . . . .                                                                                                                                                                                                                                                                                                                                                                                                    | 16 |
| 5.2  | Mean absolute error without interpolation . . . . .                                                                                                                                                                                                                                                                                                                                                                                           | 18 |
| 5.3  | Mean absolute error with interpolation . . . . .                                                                                                                                                                                                                                                                                                                                                                                              | 19 |
| 5.4  | Mean absolute error for individual signal or approach . . . . .                                                                                                                                                                                                                                                                                                                                                                               | 19 |
| 5.5  | AE after fusion of different IMF layers of the PPG-AM signal . . . . .                                                                                                                                                                                                                                                                                                                                                                        | 25 |
| 5.6  | AE after fusion of different IMF layers of the ECG-AM signal . . . . .                                                                                                                                                                                                                                                                                                                                                                        | 25 |
| 5.7  | AE after fusion of different IMF layers of the HRV signal . . . . .                                                                                                                                                                                                                                                                                                                                                                           | 25 |
| 5.8  | AE after fusion of different IMF layers of the PRV signal . . . . .                                                                                                                                                                                                                                                                                                                                                                           | 25 |
| 5.9  | Comparison of AE before and after optimisation . . . . .                                                                                                                                                                                                                                                                                                                                                                                      | 27 |
| 5.10 | Performance comparison between two implementations . . . . .                                                                                                                                                                                                                                                                                                                                                                                  | 28 |
| 7.1  | The fusion output of AM, FM and BW without SQI using approximate referencevalue                                                                                                                                                                                                                                                                                                                                                               | 47 |
| 7.2  | The self-fusion output using average reference value . . . . .                                                                                                                                                                                                                                                                                                                                                                                | 47 |
| 7.3  | The fusion based on Xie's extraction using different weighted coefficients, where SQI-1 is calculated based on the original raw data using the method in section 6.2.1; RQI-1 is calculated based on the extraction features using the method in section 6.2.2; RQI-2 is the updated version of RQI-1; SQI-2 is calculated based on the extraction features using the method in section 6.2.3; SQI-3 is the updated version of SQI-2. . . . . | 50 |

# Chapter 1: Project Specification

The data collected from IoT wearable sensors is generally quite noisy and has motion artefacts due to the ambulatory nature of biological signals. The noisy data can result in poor feature extraction and hence could cause unnecessary misunderstanding for users. A solution to this challenge is the fusion of multi-modal signals using numerous sensors. The goal of this project is to create a multi-sensor data fusion framework along with signal quality assessment. The fusion based approach can help to keep the overall trend of the signals by using multiple signal sources. This will dramatically reduce the impact of motion artefacts and noises compared with the non-fusion methods.

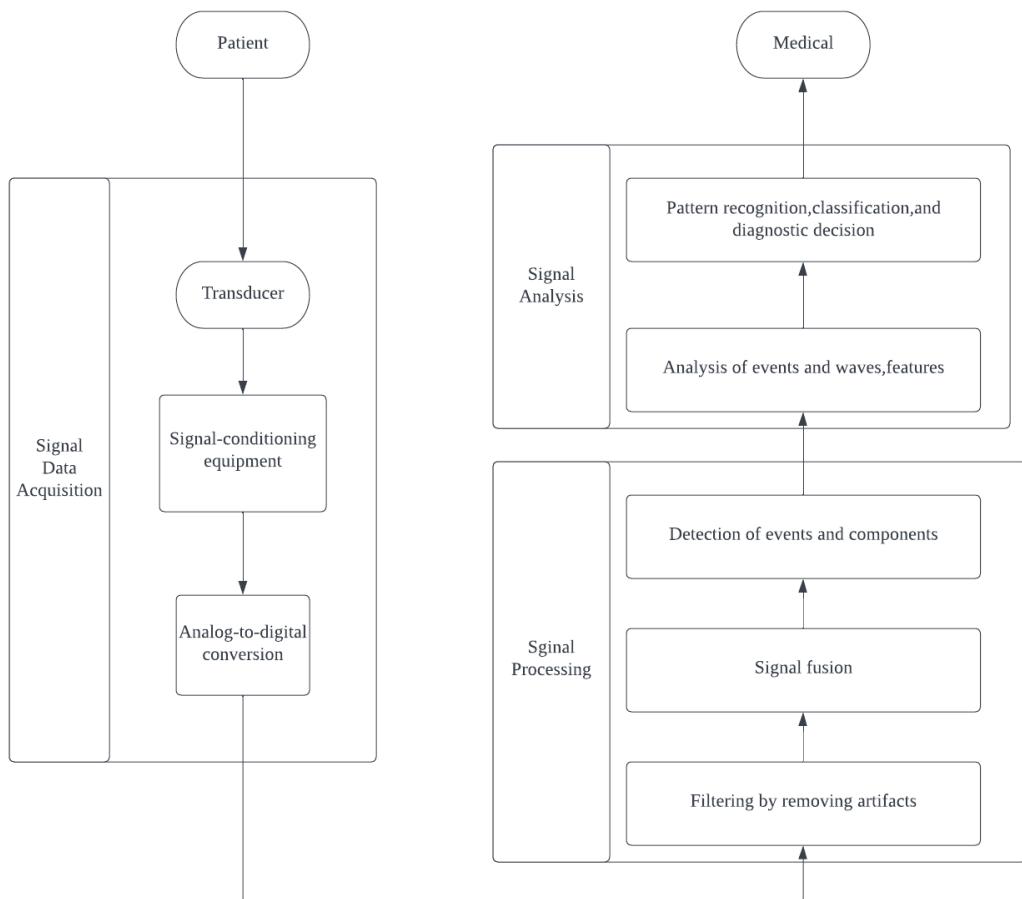


Figure 1.1: Project Background

# Chapter 2: Introduction

This report is organized into 10 chapters. Chapter 3 provides a review on the cutting-edged techniques for respiratory rate estimation. 3 pivotal steps (feature extraction, signal quality index calculation, data fusion) regarding this topic will be given separately. Chapter 4 gives an overview about work division and corresponding responsibility. Chapter 5 describes the detailed process to extract features from PPG and ECG. Chapter 6 illustrates how the signal quality indicators are derived. In Chapter 7, we implement one machine learning algorithm called Linear Support Vector Regression to fuse the data. Chapter 8 states some auxiliary work in service of the first three sections, which include an algorithm for detecting the number of breaths within a fragment, as well as data format conversion and interpolation tasks. In Chapter 9, we put forward potential optimization based on current limitations, while the overall summary is concluded in Chapter 10.

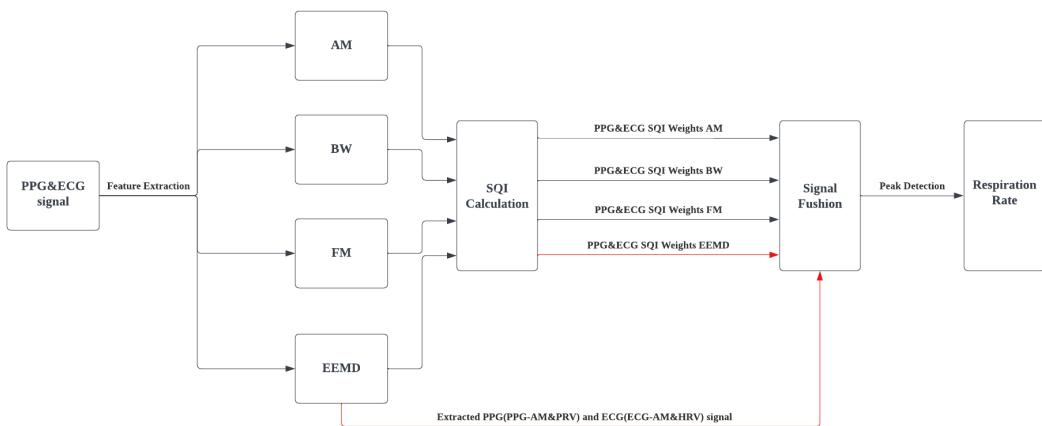


Figure 2.1: Project flow chart

The overall flow of the project is starts with feature extraction, then use extracted signal to calculate SQI value.

---

# Chapter 3: Literature Review

---

## 3.1 Feature Extraction

The first attempt to extract respiratory rate from ECG and PPG was recorded around 1990 [1]. In the past several years, multiple approaches have been developed to extract fiducial points from both PPG and ECG. More specifically, all the methods can be categorized into two main streams, namely feature-based technique and filter-based technique [2]. In [3, 4], the authors explained that the formal method contained beat-by-beat feature extraction in the time domain while the latter method concentrated on minimizing non-respiratory frequency components by filtering the original signal. As for the choice of ECG and PPG, different implementations tend to have diverse results. In [5], the authors claimed that there was no obvious contrast between ECG and PPG in the statistic. However in [1], the performance of ECG outperformed the PPG. The authors tested 253 algorithms and only 91 algorithms had better output based on PPG. From the results, they reasonably argued that it may result from different physiological mechanisms. In contrast, Meredith et al. declared that ECG was only suitable for young and healthy subjects while PPG had more possibilities to return generalized results after thorough investigations [6]. Additionally, Johansson estimated the respiratory rate from PPG solely based on the systolic waveform, diastolic waveform, respiratory sinus arrhythmia (RSA), pulse amplitude and respiratory induced intensity variations (RIIVs) [7]. Similarly, Pimentel et al. pioneered to create the Gaussian process regression framework and tested it on different modulations of PPG [3]. Importantly, they put forward that previous measurements on point estimation were of great uncertainty since they could not be quantified. Thus, their framework revealed the possibility of important physiological parameters to reduce uncertainty. Moreover, Karlen successfully adopted the Incremental- Merge Segmentation (IMS) method based on PPG which combined Iterative-End-Point-Fit and Incremental algorithms. They emphasized that this algorithm was efficient in computation and noise-robust [8]. For the ECG-based method, Mirmohamadsadeghi et al. attempted to utilize the weighted multi-signal oscillator-based band-pass filtering (W-OSC) to localize the dominant frequency component from RSA and R-peak amplitudes (RPA) waveforms[9]. Their approaches mitigated not only the load of adjusting settings due to different subjects but also the side-effects of abnormal beats or bad quality segments.

At the beginning of the project we used AM, BW and FM to extract features from the signal, but the final fusion results were far from the reference values and after a review, we suspected that the problem was in the feature extraction. After trying to optimise all three methods, the results were still poor and we decided to try a new feature extraction method. Christina Orphanidou's proposed technique employs Ensemble Empirical Mode Decomposition in order to identify the respiration "mode" from the noise-corrupted Heart Rate Variability/Pulse Rate Variability and Amplitude Modulation signals extracted from ECG and PPG signals.[5] ZHAOHUA WU & NORDEN E. HUANG propose Ensemble Empirical Mode Decomposition (EEMD) in 2009. This new approach consists of sifting an ensemble of white noise-added signals (data) and treats the mean as the final true result. As EEMD is a time-space analysis method, the added white noise is averaged out with a sufficient number of trials; the only persistent part that survives the averaging process is the component of the signal (original data).[10] After we see K. Venu Madhav et al. make use of Empirical Mode Decomposition (EMD) extracted respiratory information from ECG, BP and PPG signals efficiently[11], we consider that with the superiority of the EEMD method we can get a better result by use EEMD Decomposition.

---

## 3.2 Signal Quality Indicators

Calculations on signal quality indicators are often applied in the monitoring of heart rates. A generic signal quality indicator (SQI) has been proposed by Arlene John , Barry Cardiff et al. that can be used to detect the signal quality of any periodic or quasi-periodic signal and is particularly suitable for wearable devices[12]. This SQI is based on the waveform shape of the signal of interest and attempts to assess how similar the signal is to the expected waveform shape. Also this computational approach is applied to a novel multimodal data fusion technique using the discrete wavelet transform (DWT) proposed by Arlene John and Stephen J. Redmond, as well as to applications that fuse electrocardiogram (ECG) and optical densitometer (PPG) signals to improve beat detection accuracy for mobility monitoring using Internet of Things (IoT) sensors[13].Christina Orphanidou, Timothy Bonnici et al. have also proposed a signal quality index (SQI) to provide realtime assessment of the suitability of ECG and PPG signals for deriving reliable heart rates (HRs)[14]. This approach requires the expert judgement of different signal segments to discard some of the signals, and then the analysis of the detected R-peak/PPG pulse-peak to classify the different signals as 'good' or 'bad'. Yu-Chia Yang, Win-Ken Beh et al. present an ECG-aided signal quality assessment system for PPG, which mainly analysed the factors causing unstable signal quality of PPG signals and used to XGBoost model in machine learningto have the best performance on distinguishing signal quality[15]. 2017, a new method for extracting the respiratory rate from ECG and PPG obtained via wearable sensors is presented by Christina Orphanidou[5]. The method mentioned in the fusion phase for fusion by selecting the extraction with a small coefficient of variation also provides an idea for the calculation of SQI.

## 3.3 Machine Learning And Data Fusion

At first, I read the article from Tribeni Prasad Banerjee [16] who introduce some methods to do the fusion including algebraic functions, Kalman filtering, Bayesian estimators, adaptive observers and weighted average. Then Arlene John talks in more detail about weighted average which is also the provided method at first. Then Shahina Begum [17] mentions a decision-level fusion based on features extracted from different methods. Qinghua Gu [18] propose an rf-SVM (Regression Forecast – Support Vector Machine) to do the data fusion. Shahaboddin Shamshirband [17] uses the SVR and makes the radial basis function (RBF) and polynomial function as SVR kernel functions to do the fusion. Their proposals give me guidance about where to start to learn and implement our method.

---

## Chapter 4: Work Division

---

**Qiwei Ke** reviewed previous approaches on how to extract fiducial points from PPG and ECG and he decided to implement the feature-based extraction. Meanwhile, he chose one of the representative databases called CapnoBase as the data source. After that, he successfully obtained the characteristics with the assistance of amplitude modulation, baseline wander and frequency modulation. Followed by appropriate normalization and interpolation process, the derived data was ready for upcoming data fusion.

**Zhipeng Xie** learned how to extract and calculate respiration rate from heart rate signals by the Ensemble Empirical Mode Decomposition method and used this method in the feature extraction of selected signals. Review some peak detection and breath detection methods and implement a three-point detection approach to calculate the number of breaths within a segment. Learned how to manipulate mat files in python, and implemented an automated procedure to modify the specified data in a MAT-file and output it to a CSV file in the required format, so that the data can better serve the step of data fusion.

**Xiting Wang** reviewed various literature on SQL calculation methods and selected four methods to try out. Making innovation and optimisation according to the integration. Implementing different methods of obtaining respiration rate reference values and compare the running output to improve the final result. Worked closely with the students responsible for the other sections to make full use of the original and extracted signals, combining MATLAB and Python supervised learning models to analyse the ECG and PPG signals and assign signal quality indices for each segment.

**Tianlong Zhang** reviewed the previous article about sensor data fusion, and try to find out the combination between the traditional methods and machine learning to do the data fusion. Then I learned the knowledge about data fusion which is based on machine learning. using Python to implement the code and learning some knowledge about the CSV file and Python rules. Then trying different feature extractions and SQLs to do the fusion to find out the best combination.

---

# Chapter 5: Feature Extraction

---

## 5.1 Introduction

During the preparation, we have gone through multiple essays regrading the feature extraction. Based on these, our initial attempt was to process the signal with amplitude modulation, baseline wander and frequency modulation. After testing, the results were unsatisfying and we hardly optimize it through further experiment. So we reversed the original idea and tired a completely different approach called EEMD decomposition as our alternative scheme.

## 5.2 Database Selection

Regarding the database, I decide to use the CapnoBase dataset to conduct the survey. CapnoBase dataset is one of the well-known databases for respiratory rate estimation [19]. All signals are obtained from medical processes like elective surgery. More specifically, ECG, PPG and other referential signals which have strong relations with respiratory rate are collected from patients to assist the researchers. A piece of samples is shown in figure 5.1. Moreover, because of its popularity, many other researchers use this dataset to test their algorithms and optimize their results which can be used as our benchmark to justify whether our implementation is practical.

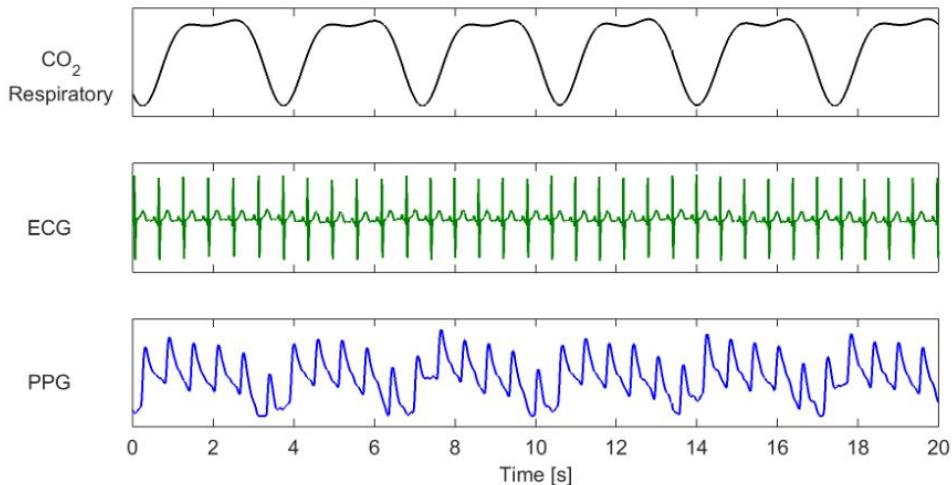


Figure 5.1: CapnoBase Exemplary signal [19]

## 5.3 Feature Extraction implementation A

### 5.3.1 Data Processing And Feature Extraction

In this section, the detailed process of data processing will be given. Among all, feature extraction and data filtering are the key steps to generate acceptable results. Note that the original signals in the dataset contain 42 pieces of information and the duration of each signal is around 480 seconds. To enrich the training samples for the upcoming machine learning section, 32 seconds fragments clipped from original data are used in the following procedure.

Instead of utilizing filter-based techniques to extract the fiducial points, multiple feature-based technologies in digital signal processing are applied to get diverse outputs, namely Amplitude Modulation (AM), Baseline Wander (BW) and Frequency Modulation (FM). Based on the modulated signal, further comparison can be made to demonstrate the performance. After introducing the basic concept of feature extraction for each modulation technique, I would like to discuss the challenges and related countermeasures during peak and trough detection.

#### Amplitude Modulation

According to the essay, features of amplitude modulation can be derived from localizing the peaks and recording their corresponding amplitudes [20].

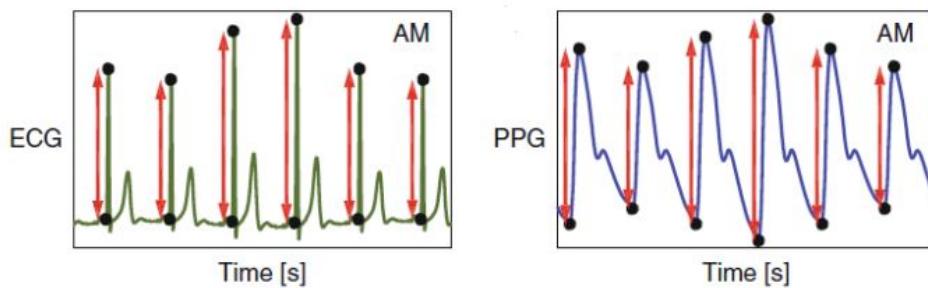


Figure 5.2: Fiducial points for AM [20]

To better illustrate the above process, PPG and ECG AM signals extracted from the same period are given in figure 5.3. In this example, 53 feature points for ECG and 50 feature points for PPG are distributed in 32 seconds. Significantly, the number of feature points from each signal fragment is not fixed and it is highly dependent on the current respiratory rate.

#### Baseline Wander

As for the features of baseline wander, two necessary steps need to be carried out. Firstly, the values for both peaks and troughs should be detected. Secondly, we take the average of adjoining peaks and values to acquire the expected outcomes [20].

To better illustrate the above process, PPG and ECG BW signals extracted from the same period are given in figure 5.5. In this example, 53 feature points for ECG and 50 feature points for PPG are distributed in 32 seconds.

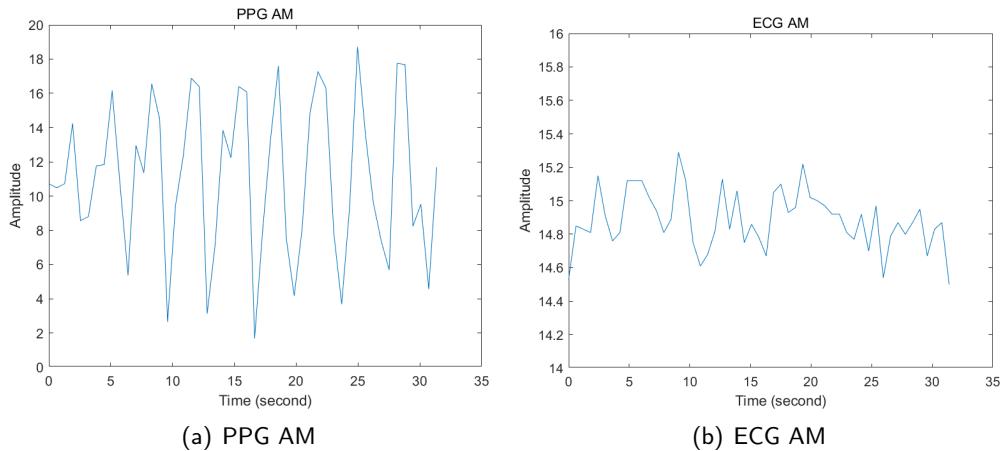


Figure 5.3: PPG and ECG AM

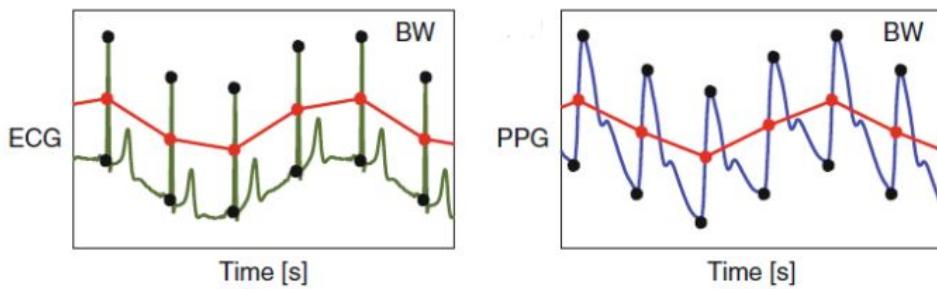


Figure 5.4: Fiducial points for BW [20]

## Frequency Modulation

When it comes to frequency modulation, the time difference between two adjacent peaks indicates the characteristic [20].

To better illustrate the above process, PPG and ECG FM signals extracted from the same period are given in figure 5.7. In this example, 52 feature points for ECG and 49 feature points for PPG are distributed in 32 seconds.

## Peak And Trough Detection

Regarding the peak and trough detection, several similar dilemmas for both the ECG signal and PPG signal are encountered. On the one hand, due to the nature of the signal, every individual signal highly differs from other signals. The unique physical characteristics of each patient lead to various amplitudes of signals which causes an extra workload for data processing. On the other hand, because of the accident like equipment failure, some data may have invalid intervals (The amplitude of the signal for the first few seconds is zero) which results in unacceptable feature collection. Aside from common difficulties, other problems will be discussed in the following section.

- **ECG Signal**

A zoomed version of the ECG signal is given in figure 5.8 for reference purposes. From the graph, certain characteristics can be concluded to reduce our workload. Firstly, fluctuation of the amplitude is within a small scale so peaks can be captured precisely with the help of the peak detection algorithm. In the beginning, in order to tackle the issue of diverse amplitudes

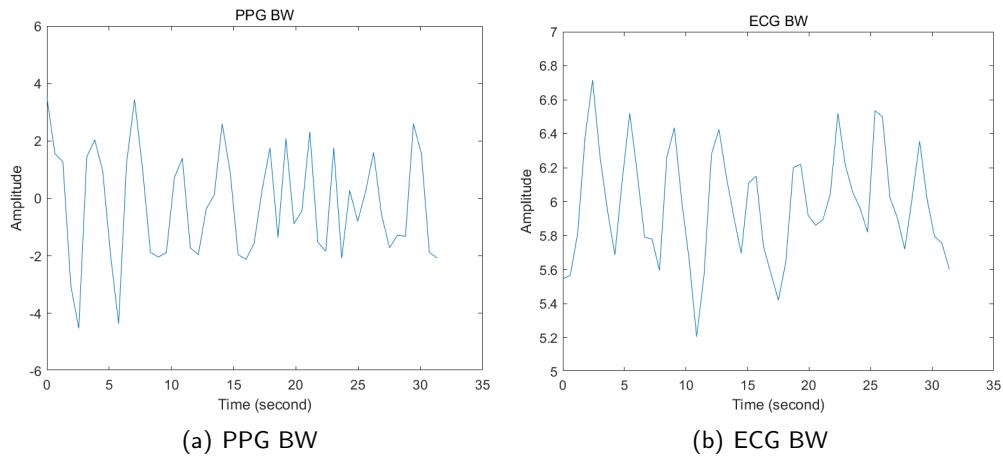


Figure 5.5: PPG and ECG BW

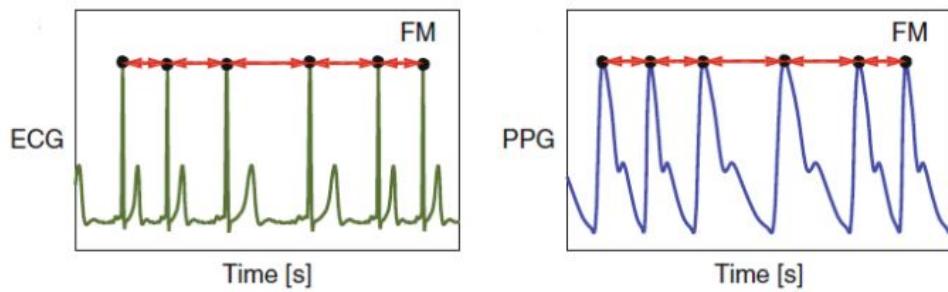


Figure 5.6: Fiducial points for FM [20]

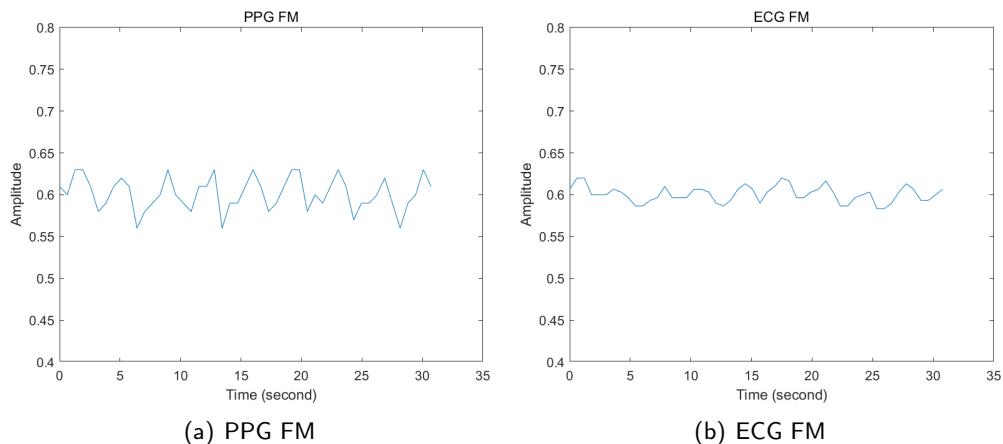


Figure 5.7: PPG and ECG FM

mentioned above, signals in the dataset are categorized according to their rough amplitudes and pre-determined thresholds can output better results. However, after some discussions with my tutor, I realize that we can not manually adjust the threshold for every signal in real-life scenarios. Consequently, I decide to obtain the approximate amplitude threshold by analyzing the beginning of the signal. Meanwhile, to deal with the potential invalid interval mentioned above, I set the window to 5 seconds. Secondly, for the trough detection, they appear 4 to 6 samples earlier than the peaks with a high probability based on observation. As a matter of fact, limitations should be admitted that this algorithm may have poor performance compared to those complicated ones. Besides, because of the size of the truncated window, certain troughs may be clipped from the current window which results in a mismatch of related peaks. As a

consequence, I have to skip this peak to keep the data consistent.

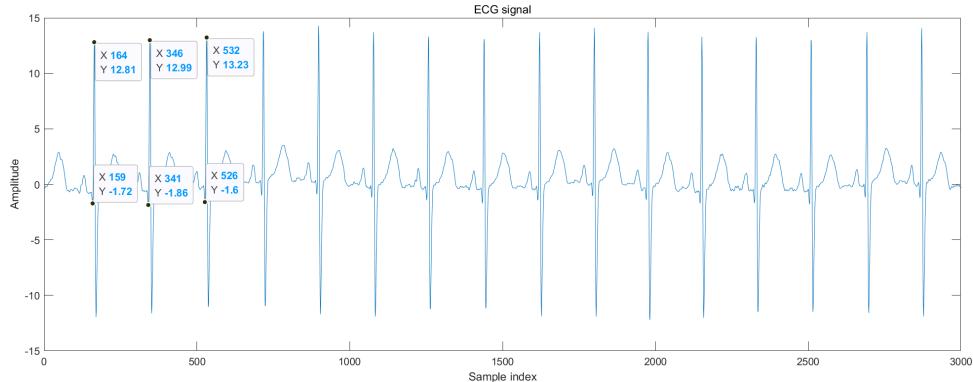


Figure 5.8: Zoomed version of ECG signal with data marks

- **PPG Signal**

A zoomed version of the PPG signal is given in figure 5.9 for reference purposes. Unlike the ECG signal, fluctuation of the amplitude for the PPG signal is relatively obvious which means that setting amplitude thresholds for peak detection is inappropriate. To conquer this obstacle, I choose to implement the peak detection with a pre-determined minimal peak distance. After careful consideration, this number is set to 120 to get the best results. Similar to the peak detection, the method for ECG trough detection turns out to have worse outcomes when I try to locate the trough of the PPG signal. Consequently, for the sake of reaching expectation as well as optimizing the re-usability of our function, I reverse the value of the entire PPG signal so that the peak detection algorithm can also calculate the trough.

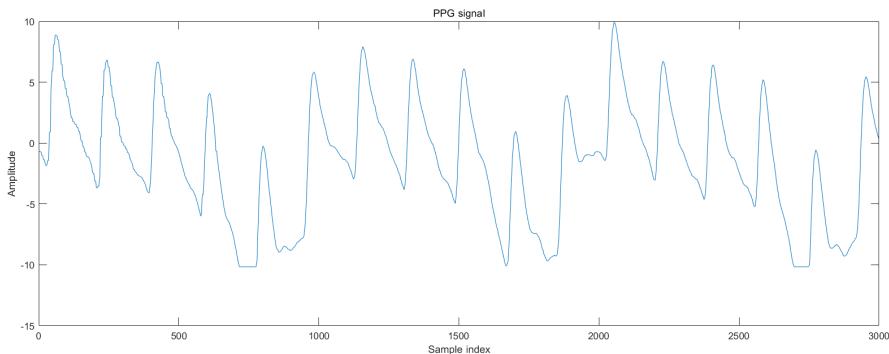


Figure 5.9: Zoomed version of PPG signal

### 5.3.2 Data Normalization

Data normalization is a mathematical process that helps scale the function domain from its original domain to the domain ranging from 0 to 1 based on its maximum and minimum value. Its formula is provided below for better understanding. From the data shown above, it is clear that the processed feature points are distributed in different areas. For instance, the range for the PPG BW signal is about from -6 to 4 while the range for the ECG BW signal is about from 5 to 7. If we directly use the data without normalization, the output of the regression model from the latter machine learning algorithm may have poor convergence. An exemplary result is listed below (figure 5.10) for reference.

$$\text{MinMaxScale} : \frac{x - \min(x)}{\max(x) - \min(x)} \quad (5.1)$$

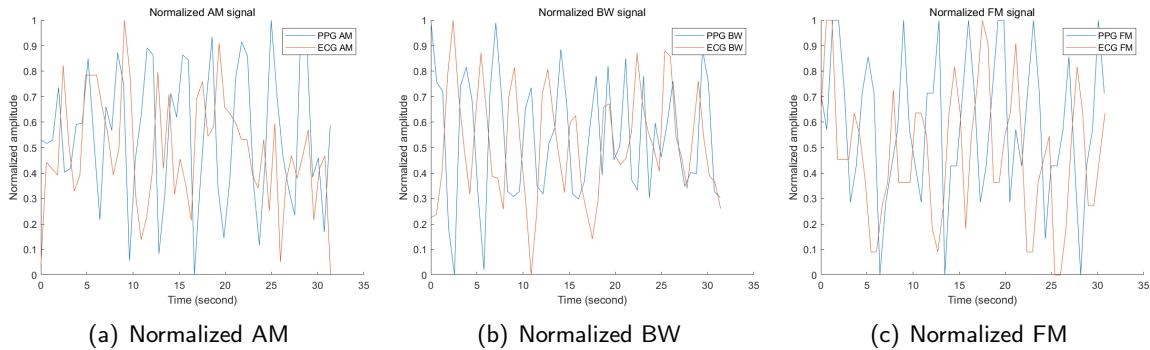


Figure 5.10: Exemplary normalized signal

### 5.3.3 Data Interpolation

Data interpolation is a mathematical process that can generate new data samples by estimation based on known samples. In this project, we aim to overcome the problems of sample shortage and make the curves smoother. In Matlab software, multiple interpolation methods are developed. Table 5.1 is shown below for a brief summary of each method (excerpted from Matlab interp1 documentation).

| Interpolation Method   | Functionality                                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Linear interpolation   | The interpolated value at a query point is based on linear interpolation of the values at neighboring grid points in each respective dimension.  |
| Nearest interpolation  | The interpolated value at a query point is the value at the nearest sample grid point.                                                           |
| Next interpolation     | The interpolated value at a query point is the value at the next sample grid point.                                                              |
| Previous interpolation | The interpolated value at a query point is the value at the previous sample grid point.                                                          |
| Pchip interpolation    | The interpolated value at a query point is based on a shape-preserving piecewise cubic interpolation of the values at neighboring grid points.   |
| Cubic interpolation    | Cubic convolution                                                                                                                                |
| Makima interpolation   | The interpolated value at a query point is based on a piecewise function of polynomials with a degree at most three.                             |
| Spline interpolation   | The interpolated value at a query point is based on a cubic interpolation of the values at neighboring grid points in each respective dimension. |

Table 5.1: Interpolation method description

Ideally, the expected curve should be smooth enough and the difference between the original data and the interpolated data should be as small as possible. To clarify the viewpoints, all the implementation performances are listed below in figure 5.11 for comparison. Note that the v5cubic method is identical to the cubic method in the Matlab application, so we only show the graph of the cubic method.

From the graph, we can easily derive the conclusion that Linear interpolation is not an appropriate method since there is no difference between the original signal and interpolated signal. Similar to nearest interpolation, next interpolation and previous interpolation, the principles behind these three methods result in discontinuous curves. In addition, the value dramatically varies at the turning point which leads to poor accuracy. Among the remaining interpolation, spline interpolation encounters the comparable issue that the fitting curve significantly deviates from the original curve. After taking the computational cost into the consideration, I choose the cubic method to implement interpolation.

### 5.3.4 Data Filtering

In this section, more details about how to solve the problems of the noisy signal will be demonstrated. In the previous introduction, 5 second ECG window is used to analyze the characteristics of the signal. Nevertheless, it is highly possible to yield unsatisfying results if the window is corrupted with noise.

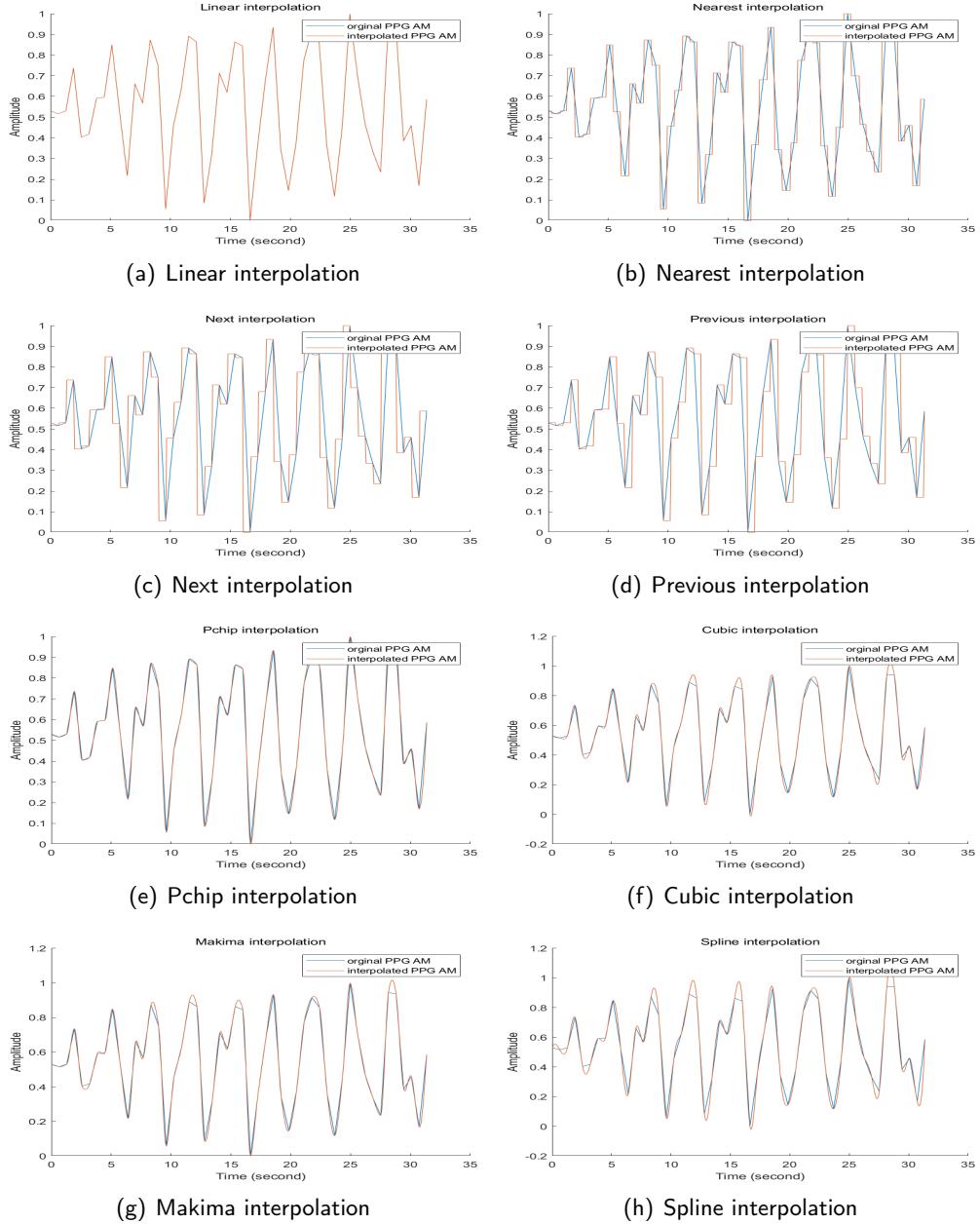


Figure 5.11: Different interpolated signals

An instance is given in figure 5.12. Alternatively, I will use the PPG signal instead if abnormal extraction exists. More precisely, if the difference in the number of the peaks obtained from ECG and PPG exceeds 4, it will be regarded as an incorrect circumstance. Finally, if both the ECG and PPG are noisy at the beginning, then we should filter it out by introducing the signal quality index (details will be shown in part 2).

### 5.3.5 Result Analysis Without SQI Involvement

Table 5.2 and table 5.3 highlight the best results and give us a clear insight into the algorithm. Mean absolute error (MAE) calculates the arithmetic average of the absolute errors between the expected output and estimated output. A mathematical formula is given below for better understanding. The less the value, the better the results. For test cases without interpolation, it is clear that ECG with frequency modulation outperforms the rest modulation with the average MAE reaching 5.0 among

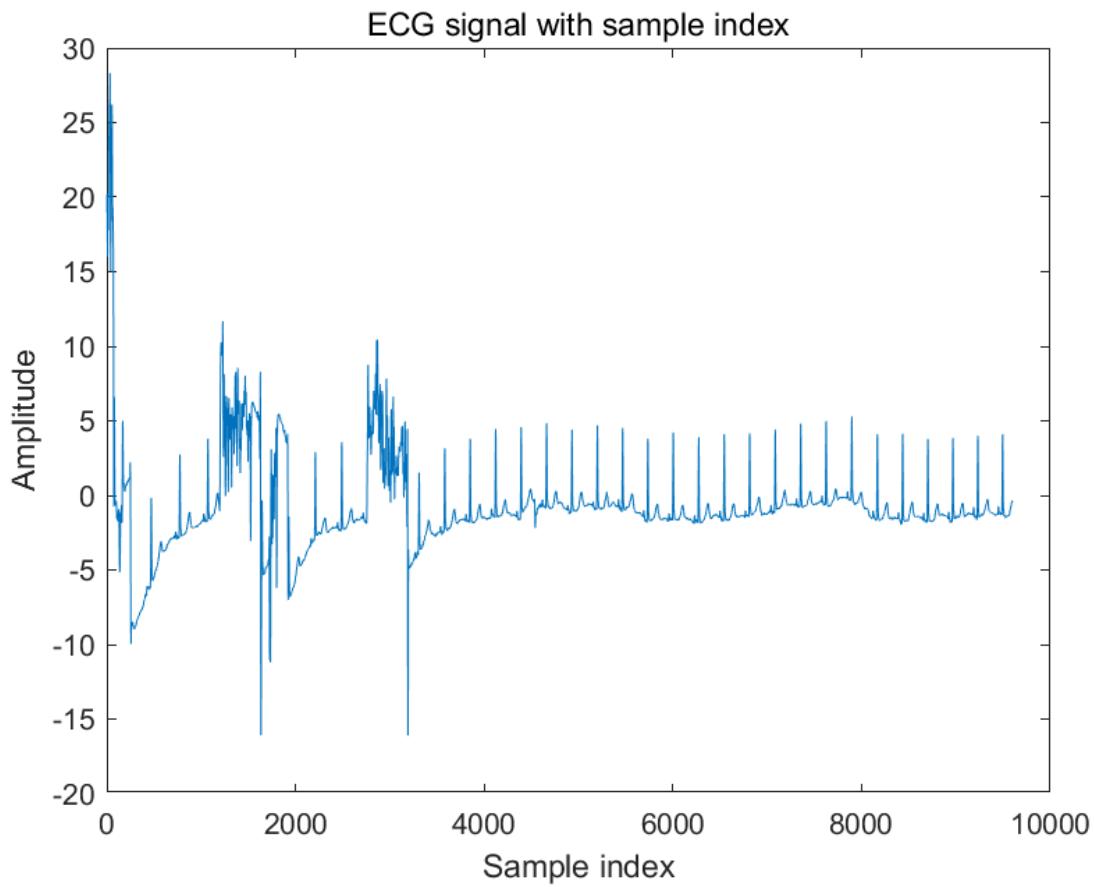


Figure 5.12: Exemplary ECG signal with noisy start

all 42 signals while ECG with amplitude modulation performs the worst with average MAE arriving at 9.1. Surprisingly, test cases with interpolation turn out to be even worse. PPG with baseline wander outmatches the rest modulation with the average MAE only reaches 8.1 among all 42 signals while PPG with frequency modulation gives the worst outputs with an average MAE arriving at 12.5.

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (5.2)$$

| Signal With Method | 25% AE | 50% AE | 75% AE | Mean MAE |
|--------------------|--------|--------|--------|----------|
| ECG + BW           | 3.9    | 7.3    | 10.7   | 8.1      |
| PPG + BW           | 1.3    | 5.0    | 8.4    | 6.0      |
| ECG + AM           | 4.6    | 8.7    | 12.5   | 9.1      |
| PPG + AM           | 3.4    | 7.0    | 10.5   | 8.0      |
| ECG + FM           | 1.3    | 3.3    | 6.7    | 5.0      |
| PPG + FM           | 3.2    | 6.5    | 10.6   | 8.0      |

Table 5.2: Mean absolute error without interpolation

Table 5.4 concludes the best choice of signal and modulation approaches. PPG is slightly more accurate than ECG in both interpolation and non-interpolation cases, with MAE reaching 10.43 and 7.33 respectively. When it comes to the modulation method, FM presents the best output among all three approaches which shows a considerable edge in non-interpolation circumstances. Nonetheless, BW outstrips others in interpolation circumstances, with MAE arriving at 9.15.

| Signal With Method | 25% AE | 50% AE | 75% AE | Mean MAE |
|--------------------|--------|--------|--------|----------|
| ECG + BW           | 6.0    | 9.7    | 13.2   | 10.2     |
| PPG + BW           | 3.0    | 7.3    | 10.7   | 8.1      |
| ECG + AM           | 6.8    | 11.1   | 14.7   | 11.4     |
| PPG + AM           | 6.2    | 9.6    | 13.5   | 10.7     |
| ECG + FM           | 4.4    | 8.9    | 13.1   | 10.2     |
| PPG + FM           | 7.0    | 10.8   | 15.4   | 12.5     |

Table 5.3: Mean absolute error with interpolation

| Signal Or Method | MAE Without Interpolation | MAE With Interpolation |
|------------------|---------------------------|------------------------|
| ECG              | 7.40                      | 10.60                  |
| PPG              | 7.33                      | 10.43                  |
| AM               | 8.55                      | 11.05                  |
| BW               | 7.05                      | 9.15                   |
| FM               | 6.5                       | 11.35                  |

Table 5.4: Mean absolute error for individual signal or approach

### 5.3.6 Discussion

Although I successfully implement the feature extraction and get the expected results, potential improvements may be achieved in the following aspects. On the one hand, we can optimize our trough detection algorithm to boost the accuracy of PPG. In the project, I mainly utilize the Matlab built-in functions to obtain the characteristics. It operates precisely during the detection but sometimes few troughs may be left undetected. Proposing own solutions towards trough detection might further decrease the MAE. On the other hand, due to the nature of feature-based extraction, very few samples can be derived during the process. In contrast, filter-based extraction concentrates on all available information provided by the signal so it can produce a respiratory signal with a higher sample rate. Therefore, denser data are possible to get more accurate results. Besides, the contrast between the non-interpolation between interpolation should be insignificant in theory. More research efforts should be focused on why cubic interpolation introduces high error. In addition, I can test my algorithm robustness by introducing other appropriate databases like BIDMC Dataset [21, 22] or MIMIC II Dataset [3] which contain additional useful reference signals and extra samples.

### 5.3.7 Section Conclusion

In conclusion, I extract the features by applying 3 feature-based methods, namely amplitude modulation, baseline wander and frequency modulation. Followed by data normalization, it aims to scale the data properly to help the upcoming machine learning process. After normalization, I implement the cubic interpolation to make the curve smoother. More progress can be carried out by improving the trough detection algorithm. Also, testing the filter-based extraction instead of feature-based extraction seems to be another practical avenue to meliorate the MAE.

## 5.4 Feature Extraction Implementation B

### 5.4.1 The Process For Feature Extraction Via EEMD Method

After trying feature extraction methods like the one described in former part we found that the average error results were not very good, so I wanted to try another approach to do this part in order to get a better result. At this part employs I use Ensemble Empirical Mode Decomposition in order to identify the respiration rate from the noise-corrupted Heart Rate Variability/Pulse Rate Variability and Amplitude Modulation signals extracted from ECG and PPG signals.

Note : This part use the same database as Part 1-1. As the database contains 42 groups of signals, all signals shown in this section are processed from group 0009\_8min.mat for presentation purposes.

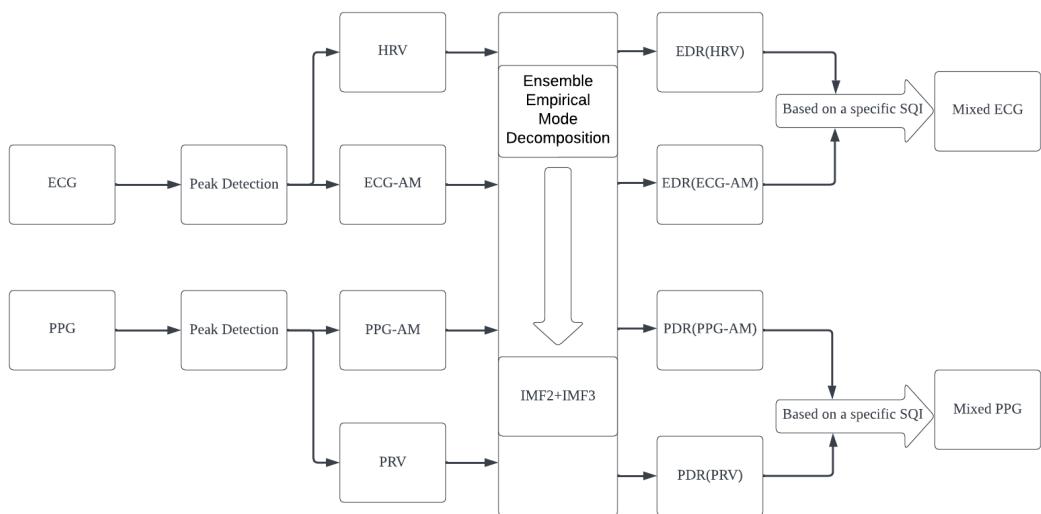


Figure 5.13: Flow chart of the proposed system

After peak detection got ECG-AM and PPG-AM signals. According to the former two signals, we can calculate HRV and PRV signals. After Ensemble Empirical Mode Decomposition is used, two IMFs are extracted from each signal and get the four respiration signals obtained from the HRV/PRV and ECG-AM/PPG-AM signals (EDR-HRV and EDR-ECG\_AM from the ECG and PDR-PRV and PDR-PPG\_AM from the PPG). ECG-Derived Respiration (EDR) and PPG-Derived Respiration (PDR) signal. Then mix the two EDR and PDR signals separately based on specific section SQI values get mixed ECG and mixed PPG signal.

### 5.4.2 Data Processing And Feature Extraction

In this section, I will explain how my ECG-AM and PPG-AM signals were extracted, how I derived the PRV and HRV signals from these two signals, and how I performed the EEMD decomposition of these four signals and selected the appropriate signal for superposition to obtain the target signal.

#### ECG-AM And PPG-AM Signals Extraction

The approach to get ECG-AM and PPG-AM signals is not same with traditional amplitude modulation. Trends in baseline wander (BW) and peak amplitude (AM) are retrieved as two separate

waveforms, the former by averaging adjoining peaks and values, and the latter by either de-trending first or taking the peak-trough distance instead of the raw amplitude. For each signal, I treat both trends as a single AM-derived waveform (shown in Figure 5.14) in my program.[5]

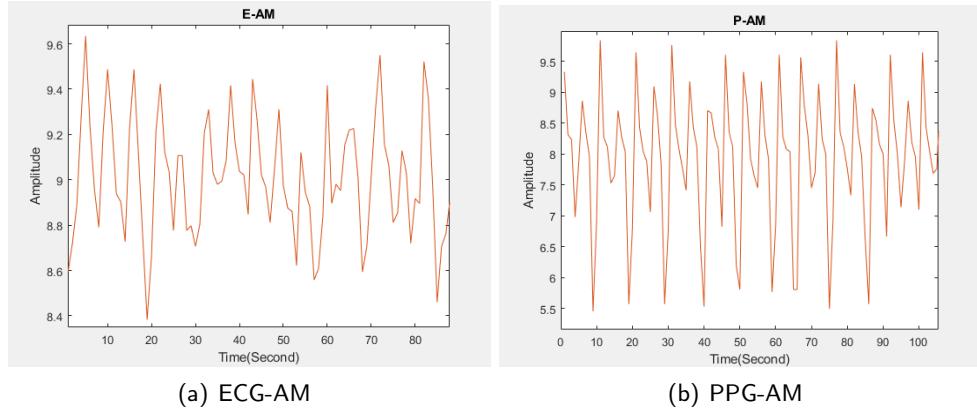


Figure 5.14: Extraction AM signal

### HRV And PRV Signals Extraction

Typically respiration affects ECG and PPG measurements with respect to both heart rate variability (HRV) and beats morphology. The heart rate variability(HRV) wave is generated by the temporal variation between the continuous peak R wave in the ECG and the peak pulse wave in the PPG. Therefore I calculate the HRV and PRV signal by the peaks gap and sampling rate as shown in following equation(a is the horizontal coordinate of the peaks). HRV/PRV signals typically contain a low-frequency component (0.03-0.15 Hz) and a high-frequency component (0.15-0.4 Hz) and the high-frequency component is found to contain respiratory frequencies.[23]

$$HRV_n = \frac{(a_{n+1} - a_n)}{f_s} \quad (5.3)$$

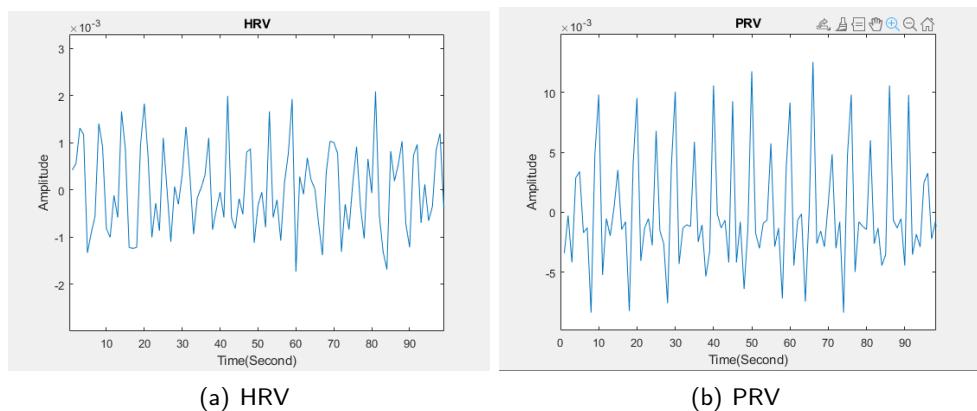


Figure 5.15: Get HRV and PRV signal from AM signal

### Ensemble Empirical Mode Decomposition

The Empirical Mode Decomposition (EMD) is an adaptive time-frequency analysis method that has proven to be very adaptable in a variety of applications. EEMD is an improvement on the Empirical Mode Decomposition (EMD). It's been widely employed in applications that call for signal extraction

---

from data generated by noisy nonlinear and non-stationary processes.[24] The EEMD decomposition is very similar with EMD decomposition except we should add white noise before EMD decomposition.

The EMD algorithm posits that a signal  $x(t)$  may be decomposed into a collection of  $n$  IMFs,  $c_j$ , where  $r_n$  is the signal residual after the extraction of  $n$  IMFs.[5] The extraction steps for the different IMFs are as follows:

$$x(t) = \sum_{j=1}^n c_j + r_n \quad (5.4)$$

1. Identifies the signal's local maxima and minima, and builds the upper envelope by linking the signal's local maxima with 3D splines, and the lower envelope by connecting the signal's local minima.
2. The local average of the upper and lower envelopes is calculated and then subtracted from the original signal  $x(t)$  to obtain the first component  $h$ .
3. The aforementioned processes are continued for signal  $h$  until the envelope meets certain conditions and becomes symmetric with respect to the zero mean. The final  $h$  is then designated as  $c_j$ , the highest frequency component of the data and the remainder of the IMF is the result of repeating step 2 until no further components can be retrieved.

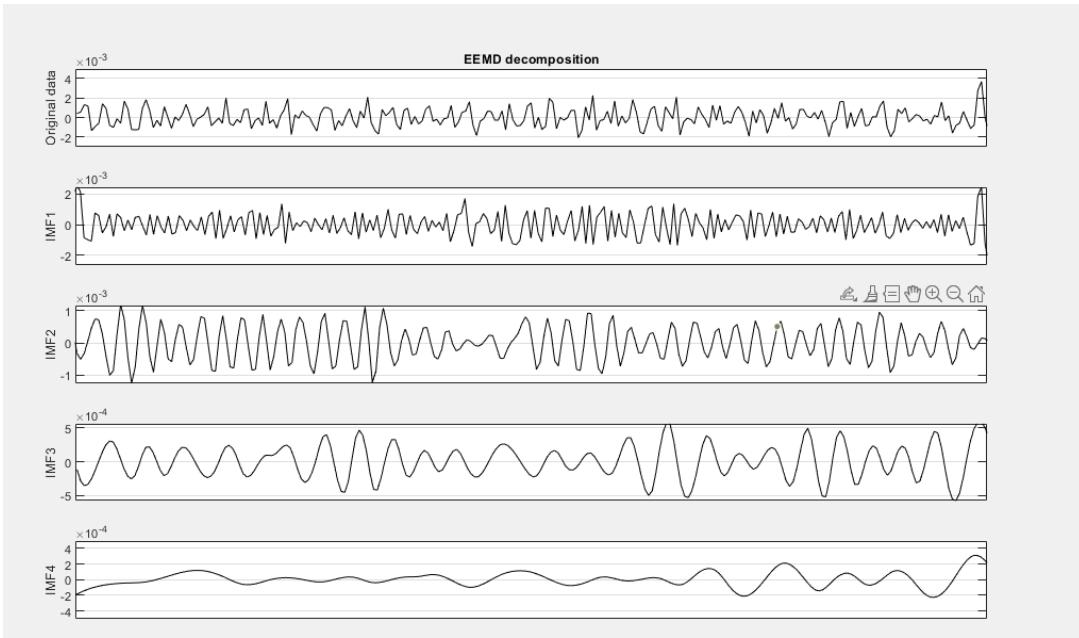
EMD algorithm can separate a real-world signal into meaningful components but this technique has been found to suffer from 'mode mixing', where multiple different scales exist within a single IMF, making it difficult to identify specific physical processes within a single IMF.[10] By introducing varying white noise  $w(t)$  to the original signal  $x$ , EEMD applies the principles of Noise Assisted Data Analysis (NADA) to the EMD algorithm  $x(t)$  to tackle this issue.

$$x_i(t) = x(t) + w_i(t) \quad (5.5)$$

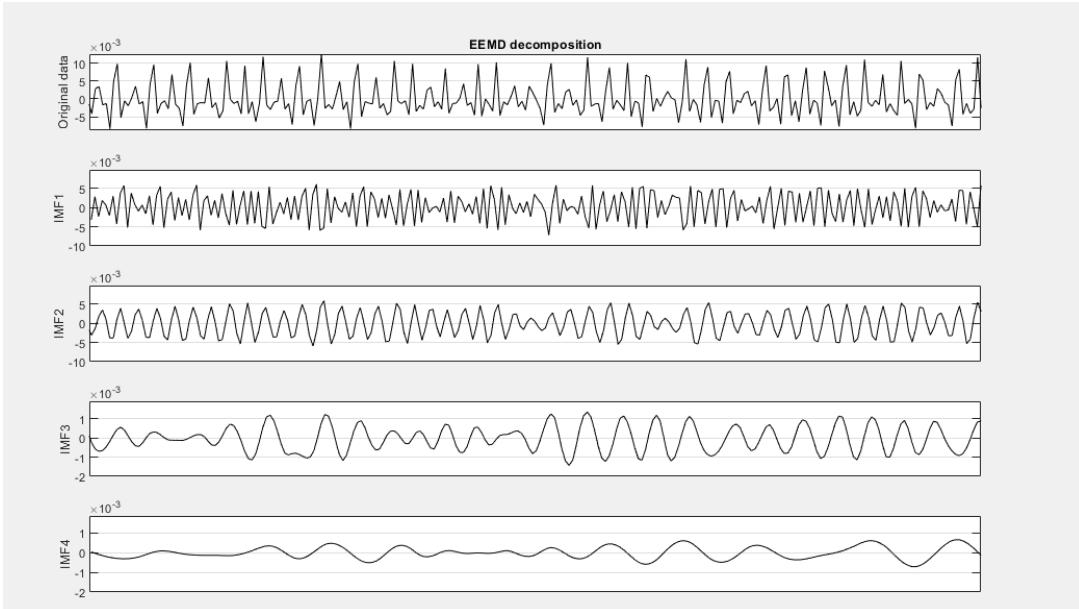
EEMD is repeated the EMD analysis process described in steps 1-3 above for each observation, and then takes the ensemble average of the corresponding IMFs as the final result.

### Application Of EEMD To Extract ECG-AM, PPG-AM, HRV And PRV-derived Respiratory Signals

Figure 5.16 shows an example of EEMD decomposition for an HRV and PRV signal and Figure 5.17 the EEMD decomposition for an ECG-AM and PPG-AM signal. Due to the limited size of the image, only a portion of the signal is captured here for image clarity.

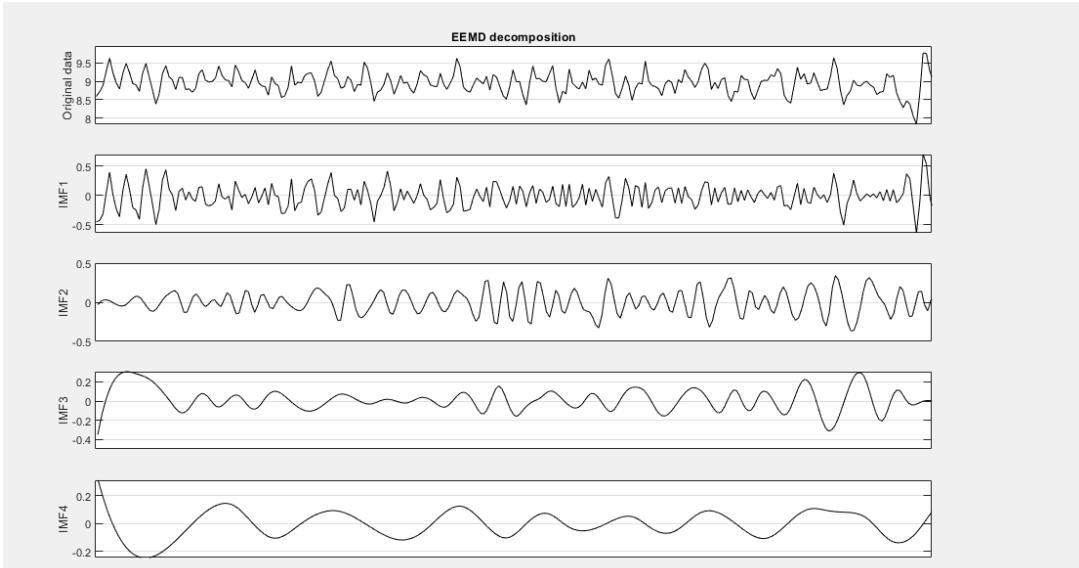


(a) EEMD HRV

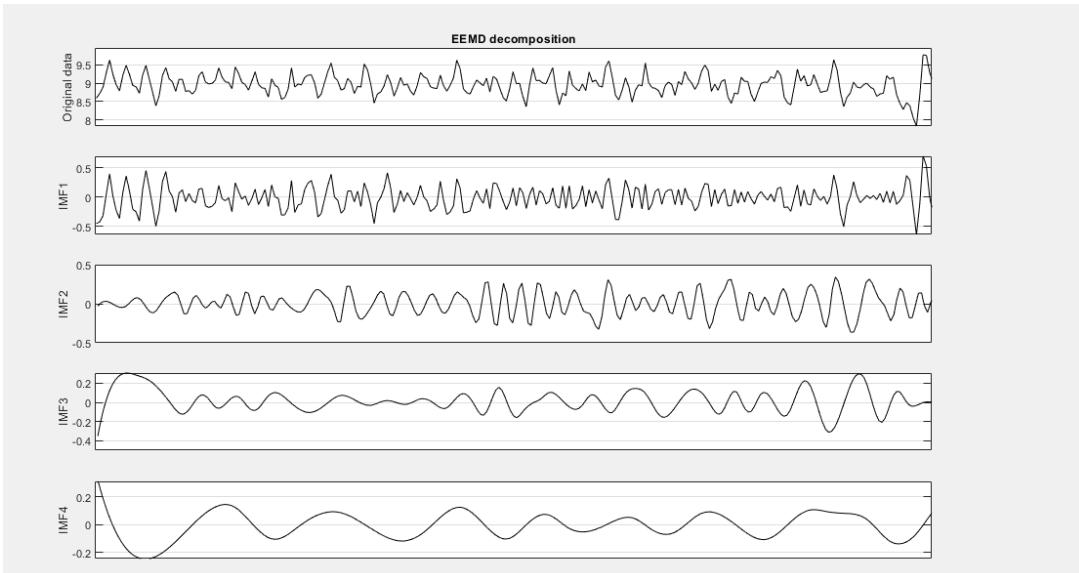


(b) EEMD PRV

Figure 5.16: Noise HRV/PRV signal (top), IMF1 (second of top), IMF2 (third of top), IMF3 (fourth of top), IMF4 (bottom)



(a) EEMD ECG-AM



(b) EEMD PPG-AM

Figure 5.17: Noise ECG-AM/PPG-AM signal (top), IMF1 (second of top), IMF2 (third of top), IMF3 (fourth of top), IMF4 (bottom)

Comparison of the decomposed IMF signals with the reference signal shows that IMF2 is nearly synchronized with the reference respiratory rate signal. In order to obtain a more accurate signal, I tried to fuse two signals from IMF1 IMF2 IMF3 and perform peak detection to calculate the respiration rate at different periods. The calculated respiration rate was compared with the reference respiration rate to find the AE value and the signal with the lowest AE.

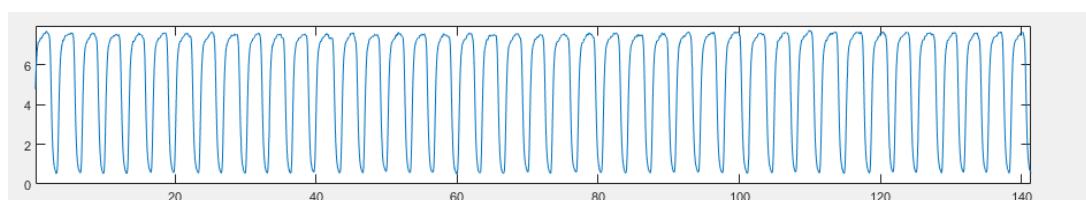


Figure 5.18: Reference Respiration Signal

I fuse IMF1 IMF2 and IMF3 separately and then use the peak detection approach to calculate the respiration rate. Choosing 25% 50% and 75% point after the sort of the result array. I use these three points' AE value and MAE value to estimate the performance of the mixed signal. According to the result shown in table 5.5, 5.6, 5.7, 5.8, we can know that IMF2 and IMF3 mixed-signal have the lowest AE value at PPG-AM HRV and PRV signals. At the ECG-AM signal, IMF2 has a lower AE value. Overall IMF2 and IMF3 mixed-signal has a more consistent and beneficial performance than IMF2. Therefore we use IMF2 and IMF3 to extract the target signal.

| PPG-AM/AE | 25%-point | 50%-point | 75%-point | mean value |
|-----------|-----------|-----------|-----------|------------|
| IMF1+IMF2 | 3.284     | 5.990     | 8.222     | 6.385      |
| IMF1+IMF3 | 4.044     | 8.396     | 10.420    | 8.115      |
| IMF2+IMF3 | 1.034     | 2.494     | 8.300     | 5.132      |
| IMF2      | 1.000     | 2.881     | 8.063     | 5.213      |

Table 5.5: AE after fusion of different IMF layers of the PPG-AM signal

| ECG-AM/AE | 25%-point | 50%-point | 75%point | mean value |
|-----------|-----------|-----------|----------|------------|
| IMF1+IMF2 | 4.917     | 7.081     | 9.116    | 7.152      |
| IMF1+IMF3 | 5.977     | 8.533     | 11.176   | 8.889      |
| IMF2+IMF3 | 0.841     | 2.673     | 5.850    | 4.521      |
| IMF2      | 0.786     | 2.460     | 5.967    | 4.454      |

Table 5.6: AE after fusion of different IMF layers of the ECG-AM signal

| HRV/AE    | 25%-point | 50%-point | 75%point | mean value |
|-----------|-----------|-----------|----------|------------|
| IMF1+IMF2 | 7.022     | 10.857    | 13.346   | 10.805     |
| IMF1+IMF3 | 9.887     | 11.653    | 14.845   | 12.134     |
| IMF2+IMF3 | 0.916     | 2.223     | 5.195    | 4.107      |
| IMF2      | 0.769     | 2.280     | 5.129    | 4.201      |

Table 5.7: AE after fusion of different IMF layers of the HRV signal

| HRV/AE    | 25%-point | 50%-point | 75%point | mean value |
|-----------|-----------|-----------|----------|------------|
| IMF1+IMF2 | 7.997     | 10.106    | 12.467   | 10.390     |
| IMF1+IMF3 | 9.281     | 11.228    | 14.183   | 11.697     |
| IMF2+IMF3 | 0.739     | 1.779     | 6.226    | 4.025      |
| IMF2      | 0.786     | 1.857     | 6.007    | 4.042      |

Table 5.8: AE after fusion of different IMF layers of the PRV signal

The following figures are the signal obtained by mixing the signals of the IMF2 and IMF3 layers:

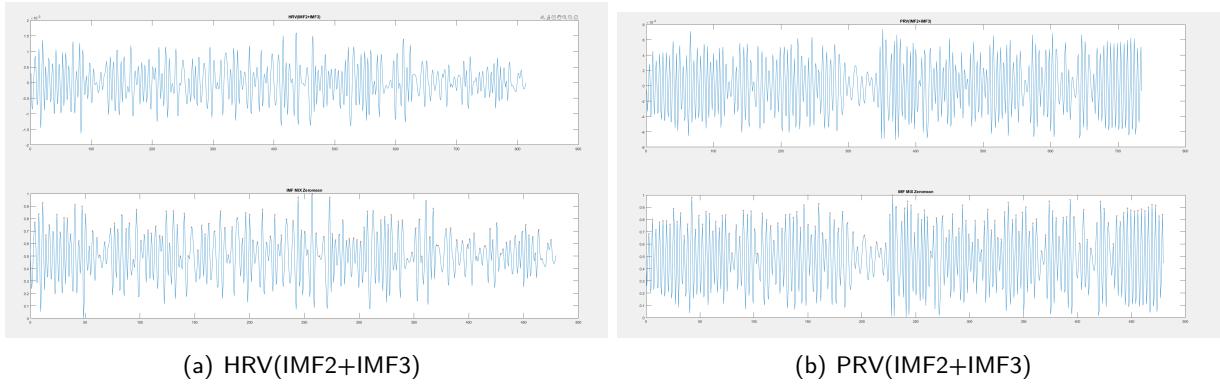


Figure 5.19: HRV(IMF2+IMF3) and PRV(IMF2+IMF3) signal at 480s

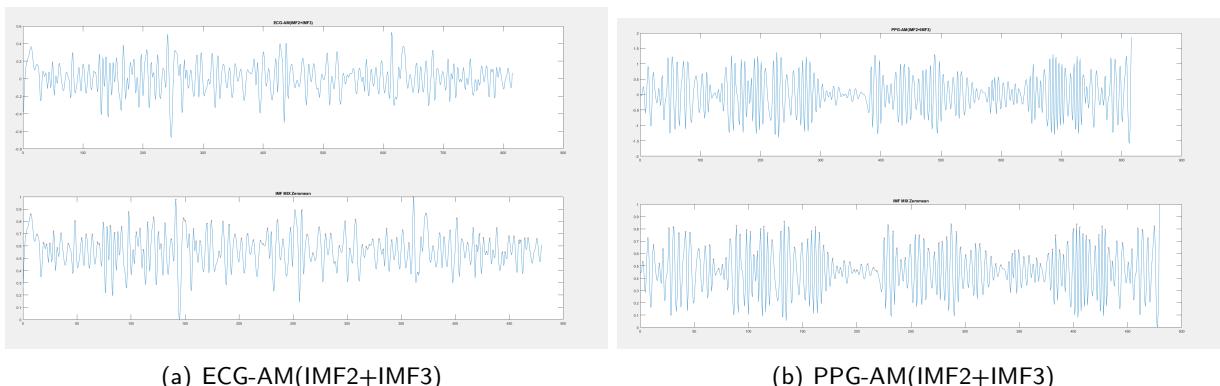


Figure 5.20: ECG-AM(IMF2+IMF3) and PPG-AM(IMF2+IMF3) signal at 480s

The difference between the top and bottom images is that the top image has the number of points as the horizontal coordinate and the bottom image has the seconds as the horizontal coordinate.

### 5.4.3 Data Normalization And Signal Slicing

To facilitate the subsequent SQI calculation and signal fusion we normalize the extracted signals between 0 and 1. At here I am using the built-in function `mapminmax` in matlab to do normalize.

This benchmark dataset contains 42 cases of 8-min recordings. We plan to split the 480s data into 15 32s segments and then find the respiration rate of each segment and then use the segment respiration rate and the reference value to find the MAE. Therefore the extracted signal and reference value should be split into 15 segments. For the extracted signal mentioned before, I divided the length of the signal by 15 and rounded it up to get the length of each segment, except the last one. The last segment will hold all the points left over from the previous equalisation. This will result in the last signal fragment holding more or fewer points than the other fragments, fortunately, during my testing this was not a serious problem in this dataset and the error was within an acceptable range. For the reference values I have extracted the set of values with the closest measurement times.

#### 5.4.4 Signal Optimization With SQI And Result Analysis

At this step I optimize the signals that we got from the former step by each section's SQI value. The specific details are weighting ECG-AM and HRV signals by SQI values to get better-mixed ECG signals. Repeating the same steps with PPG-AM and PRV signals to get a mixed PPG signal. Mixed signal shown in Figure 5.21 and Each signal's performance as below:

| Signal/AE | 25%-point | 50%-point | 75%point | mean value |
|-----------|-----------|-----------|----------|------------|
| HRV       | 1.47      | 2.58      | 5.75     | 4.42       |
| ECG-AM    | 1.62      | 2.88      | 6.53     | 4.90       |
| ECG-MIX   | 1.43      | 2.04      | 4.84     | 4.21       |
| PRV       | 1.06      | 2.198     | 6.50     | 4.19       |
| PPG-AM    | 1.50      | 2.76      | 8.38     | 5.06       |
| PPG-MIX   | 1.02      | 1.45      | 6.25     | 4.03       |

Table 5.9: Comparison of AE before and after optimisation

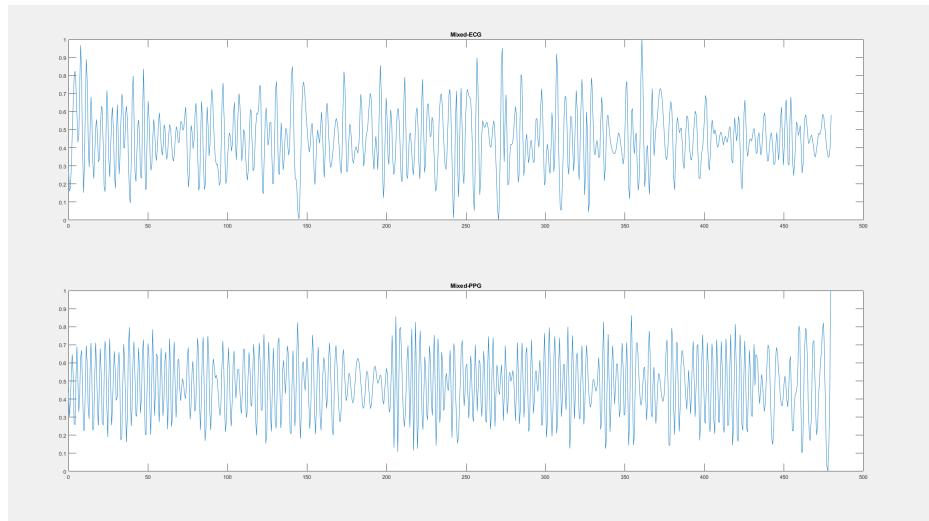


Figure 5.21: Mixed-ECG and Mixed-PPG signal. The upper one is Mixed-ECG, the bottom one is Mixed-PPG

According to the supper table, we can find that the mixed ECG signal has a lower MAE value than HRV and ECG-AM signals both at the three detection points and at the mean value. The mixed PPG signal also has better performance than PRV and PPG-AM signals. Therefore the optimization is successful.

#### 5.4.5 Discussion

Based on the above results we can find that the respiration rate signal extracted using the EEMD method has a good accuracy rate. On this basis, the signal can be further optimised to obtain higher accuracy. See Future Work chapter 9 for specific optimisation methods.

---

#### 5.4.6 Section Conclusion

In conclusion, I extract ECG-AM and PPG-AM signals from the original ECG and PPG signal and calculate HRV and PRV signals by AM signals. Then use the EEMD method to decompose extracted signals and choose suitable IMF layers to synthesise respiration signals. At last according to some optimizations, all respiration signals have a reasonable MAE value. Therefore this method is feasible and has a high accuracy rate.

### 5.5 Conclusion

In this section, we successfully extracted features with the assistance of amplitude modulation, baseline wander, frequency modulation and EEMD decomposition. We have selected the best performing signals from the two different methods for comparison. Through comparison shown in table 5.10, we find that EEMD decomposition outperforms the rest implementations without the involvement of the signal quality index. Therefore, we decide to use the extracted signals derived from EEMD decomposition in the following sections.

| Best Performance Signal | 25%-point(AE) | 50%-point(AE) | 75%-point(AE) | Mean Value of MAE |
|-------------------------|---------------|---------------|---------------|-------------------|
| Implementation A        | 1.300         | 3.300         | 6.700         | 5.000             |
| Implementation B        | 0.739         | 1.779         | 6.226         | 4.025             |

Table 5.10: Performance comparison between two implementations

---

# Chapter 6: SQI calculation

---

## 6.1 Introduction

This section is mainly focus on the calculation of the SQI. Signal Quality Indicators (SQIs) can be used as a confidence metric to assess the quality of data we obtained and reduce absolute error (AE) in the estimated respiration rate.

Recent literature on data fusion has highlighted the importance of selecting only the clean part of the signal for fusion[25]. However, in our project the final fusion part uses deep learning techniques, so the demand for data volume cannot be ignored. We decided to use all 42 sets of data and cut each set into 15 segments to meet the fusion requirements. Rather than just classifying the signals as 'good' or 'bad', we had to analyse the ECG and PPG signals for each segment and finally assign a precise value to each of the two signals to represent its weight in the fusion phase. In this section I used three methods to calculate the SQI from the original signal, the extracted signal, and the original and extracted signals together, which I will describe in detail in the report.

## 6.2 Methodology

### 6.2.1 Method 1:A calculation method based on original signals

This method uses adaptive template-matching. It is a Template-matching approach for signal quality assessment of the ECG and PPG based on that proposed by Christina Orphanidou, Timothy Bonnici et al. in 2015[14]. Regardless of the QRS complex or PPG pulse the actual morphology of the waveform in a given ECG or PPG sample, template-matching searches for the regularity of a segment, which is an indicator of reliability.

The steps in my calculation are as follows:

1. Find the PPG-peaks/QRS complex for each piece of ECG/PPG signal
2. The average interval is calculated.
3. Individual R-peak/pulse-peak can be extracted by taking a window, the width of which is the median beat-to-beat interval, centered on each detected pulse-peak/R-peak.
4. QRS template/PPG pulse-wave template is obtained by taking the mean of all extracted signal of ECG/PPG from step 3 in that segment of sample.
5. The correlation coefficient of each individual QRS/PPG pulse-wave with the average template is then calculated.
6. The average correlation coefficient is finally obtained by averaging all correlation coefficients over the sample.
7. Normalize CC(correlation coefficients) for ECG and PPG to get the SQIs for that segment.

The overall calculation process can be represented as the following flow chart:

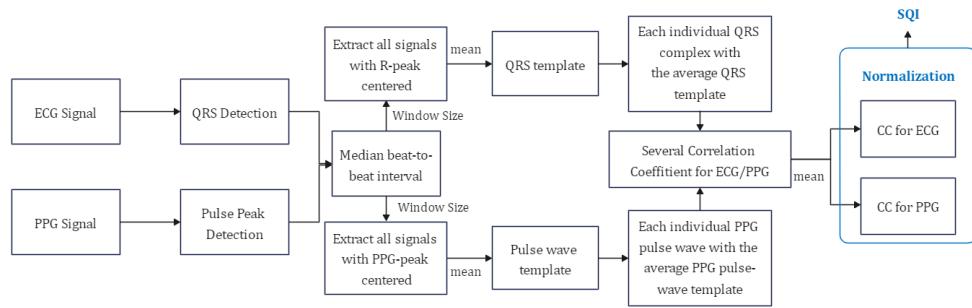


Figure 6.1: Calculation process for one segment of ECG/PPG

According to the process I have described, we can perform peak detection of the ECG and PPG signals for each segment, the production and application of templates, and finally obtain the SQI values of ECG and PPG on each segment by calculating the correlation coefficients. The basic shapes of the ECG and PPG templates are roughly as shown below.

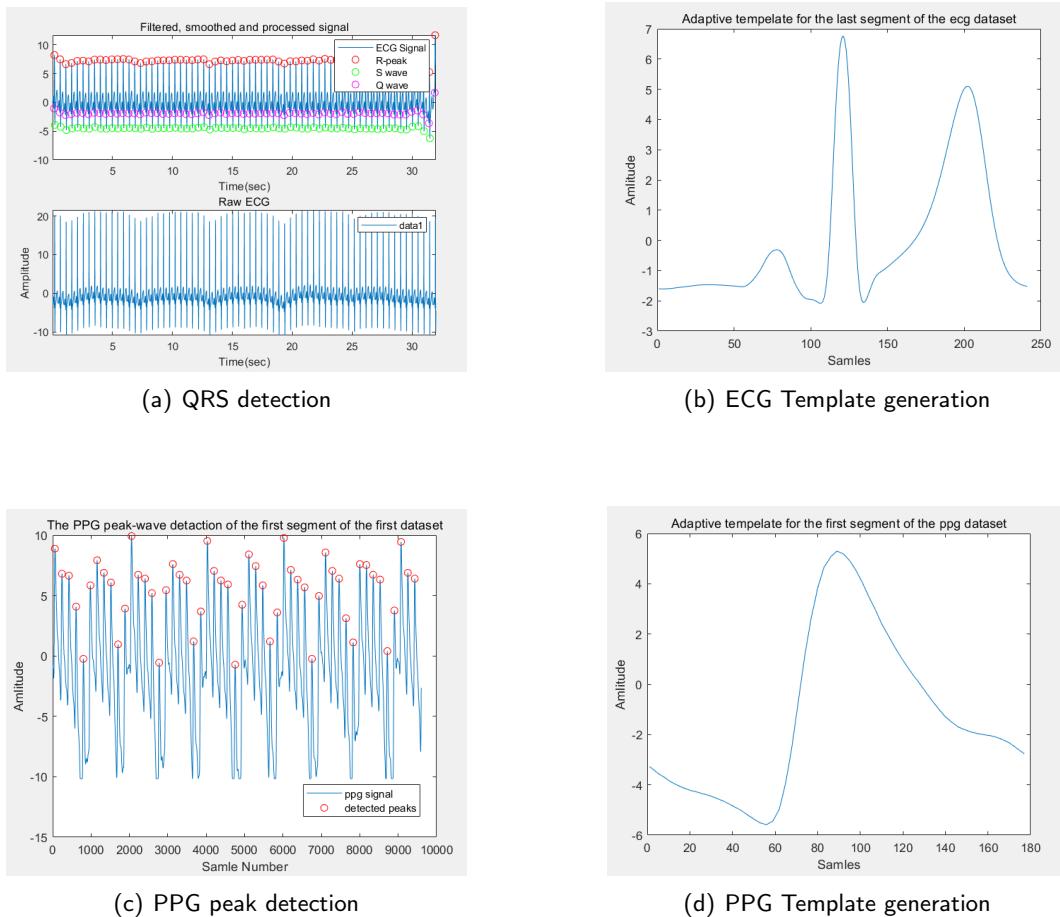


Figure 6.2: Process for template generation for one segment

This has the advantage of allowing the detected peaks to be in the middle of the range when the template is created, which allows the quality of the signal to be judged effectively by the capture and evaluation of ECG and PPG features. As there are many detectable features on a segment, the use of averaging in the calculation of the window size, the template and the calculation of the correlation

coefficient representing the segment results in a more rigorous and accurate calculation process.

I also encountered some problems and challenges. For example, the PPG peak detection often detects features that are not what we need, and this requires adjusting the minimum detection distance and setting thresholds to improve the accuracy of the detection.

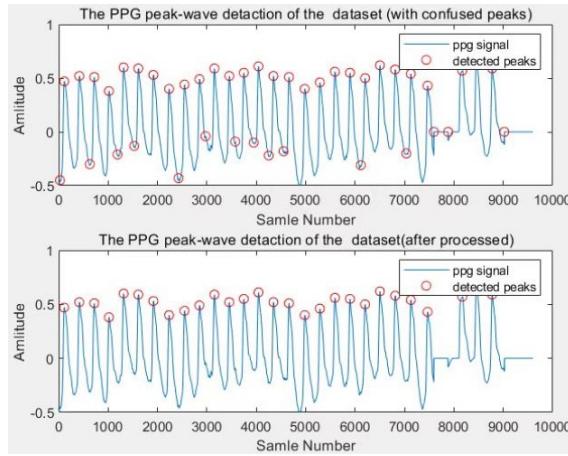


Figure 6.3: Optimization process of detection

After the calculation is completed, I want to check the validity of this method. According to the logic of this algorithm, the signal that is less affected by noise, i.e. the better quality of the signal will have a higher correlation coefficient, thus increasing its weight in the fusion. For this, I pick out different segments of the ECG and PPG signals with different correlation coefficients and verify them separately.

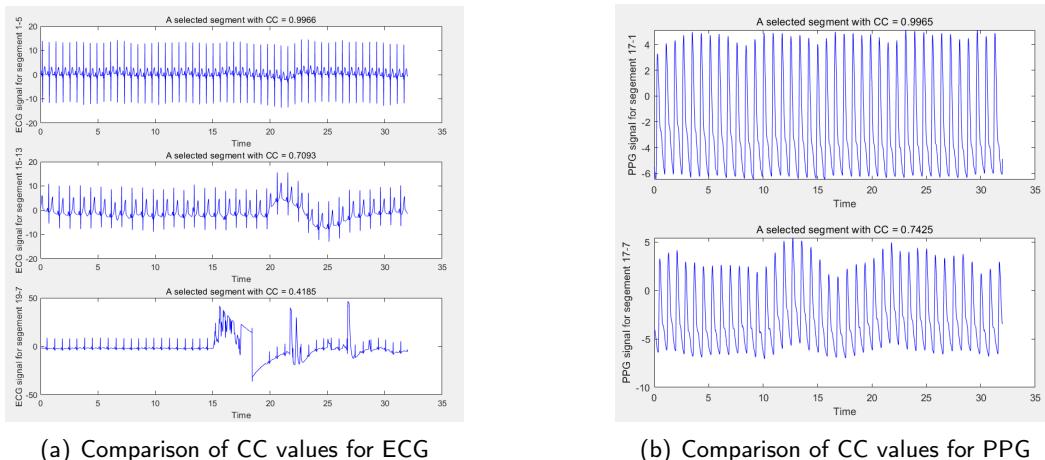


Figure 6.4: Process for template generation for one segment

It can be seen that with the same calculation method, the correlation coefficient I calculated does reflect the quality of the ECG/PPG signal on the fragment, with a small correlation coefficient indicating that the detected signal is very irregular and unstable on the fragment. Overall, the signal quality of the PPG signal is better than that of the ECG signal.

I next tested the performance of the calculation when different levels of noise were added. I added different levels of Gaussian white noise to the same set of data and could see that the correlation coefficient tended to increase as the signal-to-noise ratio increased and was essentially at the same level.

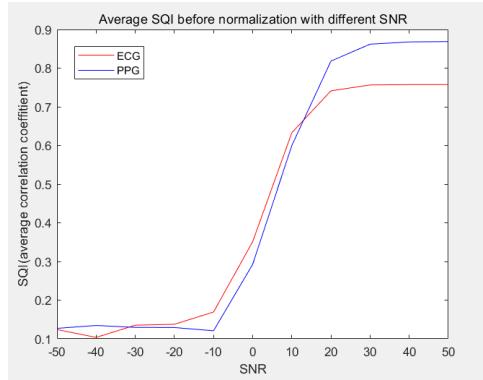


Figure 6.5: Average SQI before normalization with different SNR

### 6.2.2 Method 2:A calculation method based on extracted signals

The method derives the SQI by calculating the coefficient of variation of the extracted signal. the smaller the value of the coefficient of variation (after extraction) for a fragmented ECG or PPG signal, the larger the SQI. the CV is the ratio of the standard deviation to the mean value and is expressed by the formula:

$$CV = \sigma/\mu \quad (6.1)$$

The standard deviation is used to reflect the extent to which individual data values deviate from the mean of the data, i.e. whether the data are concentrated or not. The larger the value of the standard deviation, the greater the deviation of the individual data from the mean, and then the less representative the mean is of all the data. The coefficient of variation is also known as the standard deviation coefficient. A small standard deviation means that the data are less discrete, which means that the signal is of better quality.

After the first part of the extraction, I had two sets of data:

- The first set was ECG-FM and PPG-BW, which were verified by my group member to be the best performing of the six signals he processed, as discussed in Part 1-1.
- The second set of signals contained two more groups.
  - One is the ECG and PPG extracted by their respective AM and HRV and then weighted by a certain ratio.
  - The other group is the unprocessed AM and HRV signals. I need to calculate the SQI\_AM and SQI\_HRV initially to complete the first stage of fusion for ECG and PPG, and then find the SQIs of the resulting signals for each segment to support the final stage of fusion.

In this way we made various bold attempts to produce different results for comparison and analysis of the extracted signals, providing ideas for subsequent optimisation.

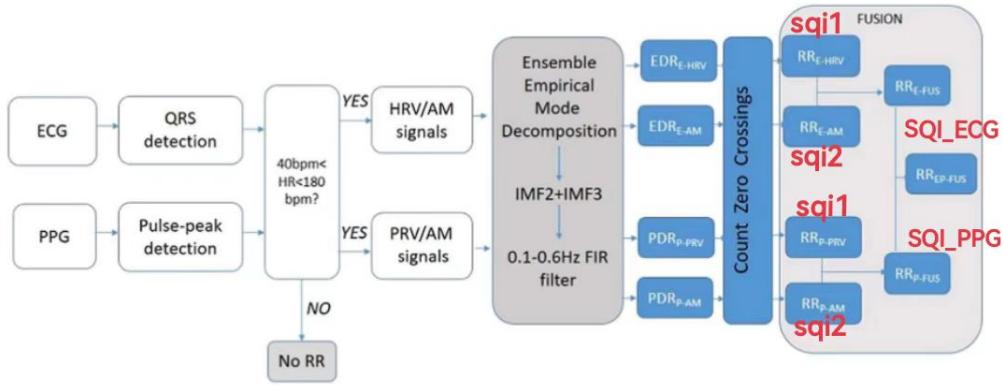


Figure 6.6: The flowchart of one proposed algorithm for data fusion[5]

This method is based on the improved fusion algorithm proposed by Christina Orphanidou in 2017[5]. The exact flow of this method is shown in the figure above. In this algorithm, the last signals used for fusion are the one with the smaller coefficient of variation of the two extractions of the ECG and the one with the smaller coefficient of variation of the two extractions of the PPG. Therefore, only two of the four extractions are used for fusion in each segment. For this method, I also performed SQL calculations and substituted them into the fusion stage, but the results were not good, so we do not discuss them here and finally decided to involve all four extractions.

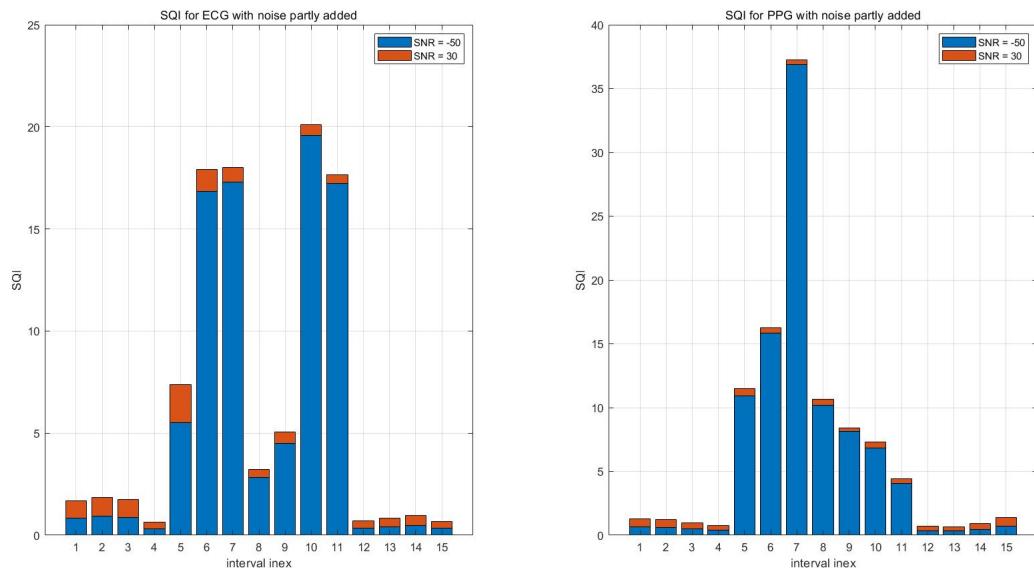


Figure 6.7: SQL for ECG and PPG signals with noise partly added

After calculating the SQL, I tested its validity by adding noise. First, I added varying degrees of noise to the 5th - 11th of the 15 parts of a data set, and we can clearly see from the graph that at a signal-to-noise ratio of 30, the value of the SQL tends to zero, close to what it would be in the absence of noise interference, while at a signal-to-noise ratio of -50, the SQL value approaches 60 at its maximum, implying a great degree of data dispersion.

This calculation of the SQL gives a direct and effective indication of the quality of the extracted signal. In the fusion phase we want a smoother and more stable and focused signal, and for the calculation of CV, it gives us exactly the reference for signal reliability.

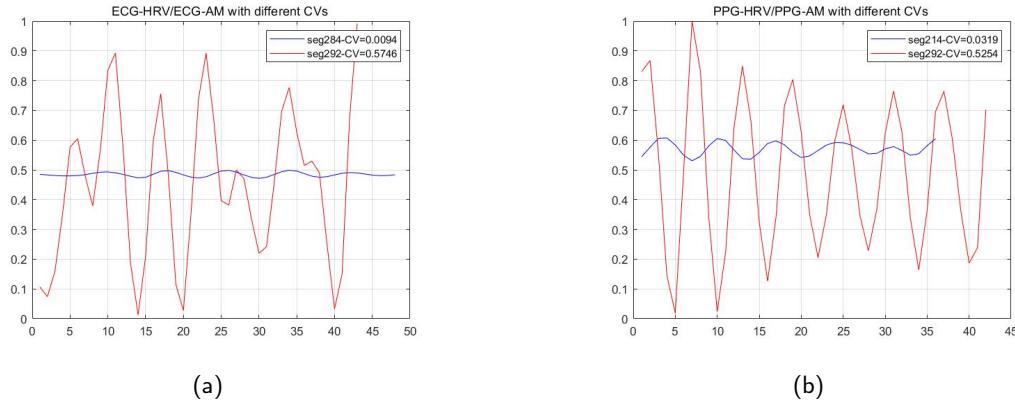


Figure 6.8: Comparisons of the quality of extracted signal labelled with different SQI

Next, I investigate the linear relationship between SQI and SNR. For the 630-segmented signal (15\*42), the mean values were calculated after adding noise at each of the 11 different SNRs to obtain more accurate data. It can be seen that for both ECG and PPG signals, the SQI values show an overall decreasing trend as the SNR increases, and that the SQI values for ECG and PPG are essentially at the same level with the same noise effect.

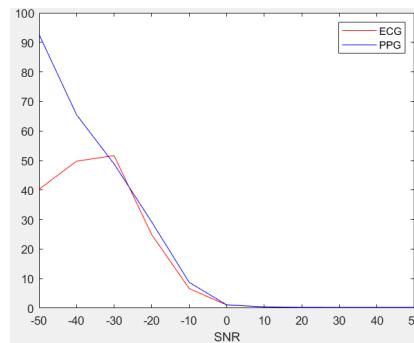


Figure 6.9: SQI with different SNR

Note: The SQI here refers to the coefficient of variation obtained after calculation and should be finally normalised. The smaller the CV, the better the quality of the signal and the larger the final SQI value for fusion.

### 6.2.3 Method 3: Calculation method using both the original signal and the extracted signal

In the calculations of the two methods described above, I sought to analyse each set of extracted or unprocessed ECG/PPG signals and use the SQIs as my analysis to provide a reference for the fusion stage. But I am also curious about if it is feasible if we start with the results and find a reference that is representative of the signal's weight of the fusion.

Our AE is the error in the respiration rate of each segment and the formula for the source of this error is:

$$\frac{\text{Number of detected peaks}}{32} \times 60\text{-reference value} \quad (6.2)$$

Our goal is to reduce this AE, so I suddenly had the idea that before fusion, we could detect and

analyse the error of the respiration rate of the different signals (ECG/PPG) on each segment, and derive SQIs from these values of error, the smaller error would be assigned a larger SQL. And I finally took the average of ECG\_SQL and PPG\_SQL of 630 fragments respectively, as the SQL of ECG and PPG during the whole fusion process.

#### A.Reference respiration rate:

In the database we use, the respiration rate reference is stored in structure **reference.rr.co2**, with **x** representing time and **y** corresponding to the instantaneous respiration rate. Therefore, we have two ways of finding the respiration rate reference.

- Let the Reference\_RR of each fragment the closest value to the respiration rate detected at the end of the desired fragment.

| reference.rr.co2 |             |
|------------------|-------------|
| reference.rr.co2 |             |
| 字段               | 值           |
| x                | 1x93 double |
| y                | 1x93 double |

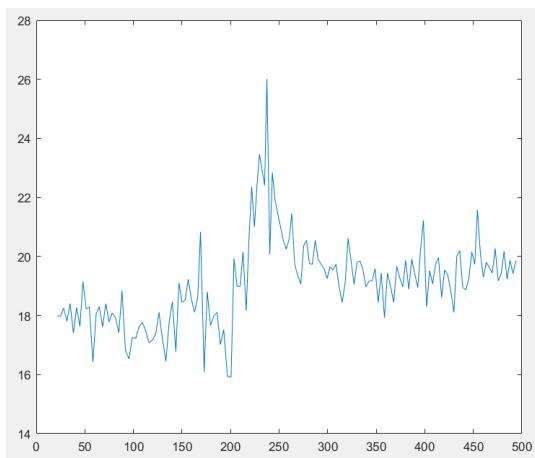
(a) Reference in dataset

|         |         |         |
|---------|---------|---------|
| 2       | 3       | 4       |
| 25.6733 | 30.9767 | 36.6233 |
| X       |         |         |
| 2       | 3       | 4       |
| 12.9403 | 11.3136 | 10.6257 |
| Y       |         |         |

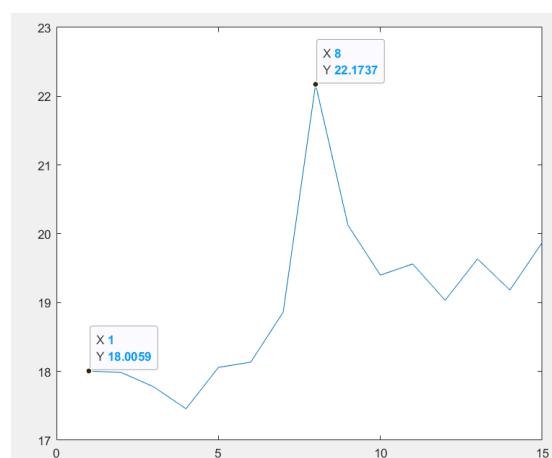
(b) Instantaneous rr value for 32s

Since each dataset is 480s (144,000 samples, 300Hz) and each is 32s (9600 samples) when cut into 15 parts, I have taken a reference value of **x** (time) that is closest to the value of **y** (instantaneous respiration rate) at 32s, 64s ..... 480s.

- Let the Reference\_RR for each fragment the average of all respiration rate values detected for each fragment. Before the calculation, the respiration rate of a segment is a discrete value, after the processing we can get the corresponding respiration rate value for each segment.



(c) Discrete reference values



(d) One value for each segment

The next comparison of the two calculation methods on the final fused AE gives the table 7.2 . It can be seen that Reference\_RR performs better when taking the instantaneous values, so we decided to use the respiration rate calculated by the first method as the reference value.

## B.More Exploration:

After the above calculations, we have obtained a set of SQI values for the ECG and PPG signals used for fusion, next I would like to make some adjustments to make this value more accurate and thus obtain a smaller error in the fusion detection phase.

I took into account that the process of extracting the signals and calculating the respiration rate values earlier was not perfect, so it is possible that the SQI values that were assigned to these 630 segments were either large or smaller. The results might have been better if the original signal had been combined to change the pre-assigned SQI of some of the segments, thus changing the final mean value used for fusion. We can start with one of the original signals, ECG or PPG, and here, I chose the PPG signal. So, I designed an optimization method based on the algorithm proposed by Yu-Chia Yang et al. in 2020[15] on top of my original calculation method.

First, I used a bandpass filter (0.5-15Hz) to remove the baseline of the PPG. Next, the PPG signal of each segment was extracted with the following feature values.

| Feature Type              | Features                                                                 |
|---------------------------|--------------------------------------------------------------------------|
| Statistic Features        | Median<br>Range<br>Standard Deviation<br>Kurtosis<br>Skewness<br>Entropy |
| Frequency Domain Features | PSD in{1Hz,3Hz,5Hz,7Hz,LH,FH }<br>FREQSTD,LF/HF                          |

\* PSD: Power Spectral Density; LF: PSD in 0.01Hz-3Hz; HF: PSD in 1Hz-3Hz

Now we have the 14\*630 ppg signal feature data. The next step is to optimize on the original calculation as shown in the flow chart.

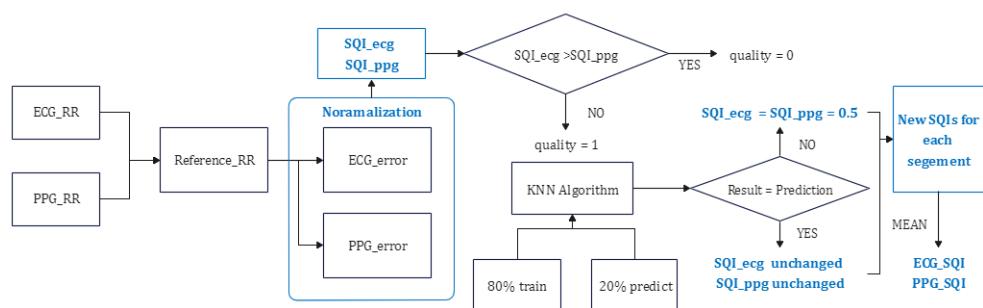


Figure 6.10: The flowchart for optimization calculation

- The quality of the ppg signal is first classified as follows.  
 $SQI_{ecg} > SQI_{ppg}$  quality = 0 (Lower quality)  
 $SQI_{ecg} \leq SQI_{ppg}$  quality = 1 (Higher or Medium quality)
- The principle of the KNN algorithm (K-nearest Neighbour) is that when a new quality value is predicted, the class to which the quality belongs is determined by the class to which it is nearest to the K points. This means that the computer automatically finds the K points from the training set that are most similar to the Features provided by the predicted point and classifies the predicted point according to the category of those points. In my code:

- $K = 5$
- Input Features: The 14 signal features given in the table above
- The situations where prediction errors occur and the subsequent processing are described in this table:

| Conditions  | Prediction Result                     | Explanation                                                                                                                                                                             | New SQI settings                                                                             | Explanation              |
|-------------|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------------------------|
| Condition 1 | quality=0 but being predicted to be 1 | This segment of ppg signal is classified having a lower quality, but the ppg signals (at least 3 over 5) having similar features with it were classified as a higher or medium quality. | SQI_ecg=SQI_ppg=0.5<br>(No effect on segments with the same signal quality at the beginning) | Improving the SQI of PPG |
| Condition 2 | quality=1 but being predicted to be 0 | This segment of ppg signal is classified having a higher or medium quality, but the ppg signals (at least 3 over 5) having similar features with it were classified as a lower quality. |                                                                                              | Improving the SQI of ECG |

- Since the training and prediction sets were 80% and 20% each time, I did this five times each time using group 504 of 630 as the training set (looking for datasets with similar FEATURES) and group 126 as the test set (looking for data with incorrect predictions to change the SQI values).

| Time | Accuracy           |
|------|--------------------|
| 1    | 0.6428571428571429 |
| 2    | 0.5714285714285714 |
| 3    | 0.6111111111111112 |
| 4    | 0.5000000000000000 |
| 5    | 0.5476190476190477 |

After adjusting the new resulting SQI\_ecg and SQI\_ppg for 630 segments, I calculate their mean values separately here to obtain the final SQI values used for fusion. And this is the end of the optimisation algorithm.

| Signal Type | Fianl SQI for fusion (before) | Fianl SQL for fusion (optimized) |
|-------------|-------------------------------|----------------------------------|
| ECG         | 0.478538                      | 0.456224                         |
| PPG         | 0.521462                      | 0.543776                         |

The final two sets of SQIs used for fusion are shown in the table, and the final results of our attempts to fuse these two sets of SQIs is shown in Chapter 7 table 7.3.

## 6.3 Discussion

The three methods I used made full use of each data set, analysing the original signal or the extracted signal for each segment. Of these, methods one and two have been tested for the validity of the SQI values by adding noise, and method three is a new calculation attempted based on existing data, independent of noise.

### A.Problems of the implementation

- **Method 1:**

Due to the irregularity of the signal, both the QRS detection of ECG and the pulse peak detection of PPG will have inaccuracies even if different thresholds have been set for certain data sets to make the detection more accurate. For example, when setting the threshold for PPG peak detection to discard some incorrectly detected points will also discard some other correct points.

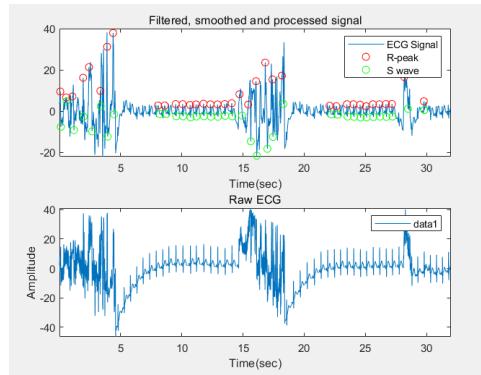


Figure 6.11: An example for the problem in R-peak detection

- **Method 2:**

The addition of SNR is not regular enough. I wanted a more stable, linear, smooth curve to make the results more accurate but tried curve fitting and did not find the right formula for the fit.

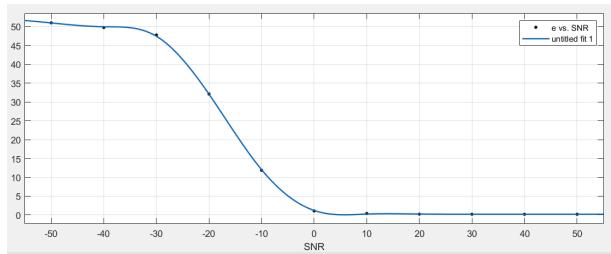


Figure 6.12: Expected curve for SNR and SQI(method 2)

## B.A new method

There is another computational method combining template correspondence and Toeplitz correlation matrix proposed by Arlene John et al[12]. The advantage of this method are:

- Gets one sqi per sample and then normalizes. We need 32s per fragment, i.e. 9600 sets of SQIs for the mean. Increasing the number of samples involved in the computation may lead to more accurate results.
- This method used curve fitting to ensure that the sqi values and the noise are linearly increasing and validate the results for Guassian Noise, Power line interference, Baseline Wander, Music Artifacts, Electrode Motion, which are the five types of noise that can easily affect signal quality in signal collection on mobile devices. When generating the template, we align the fudical points of the first 10 cycles of the signal.

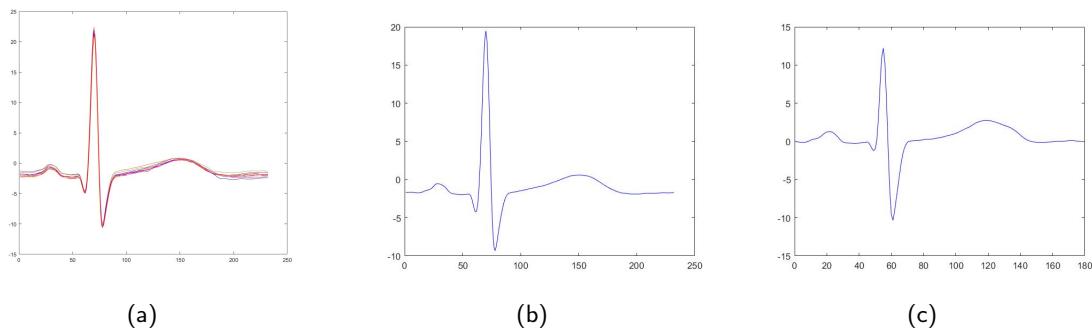


Figure 6.13: Choosing ECG template

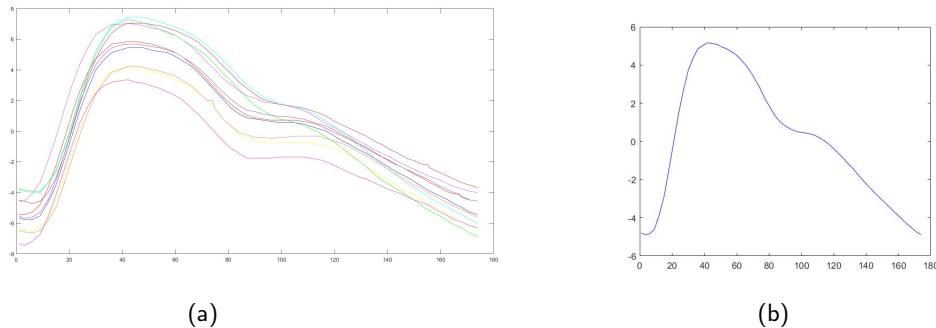


Figure 6.14: Choosing PPG template

This method requires that the cleaner signals are first selected to make templates. I have looked at all 42 groups and found two cleaner ecg groups and one cleaner ppg group and made templates. Two ecg (dataset 1 and dataset 39) signals I selected having reference rr at 18 and 16 respectively. Since the normal respiratory rate per minute for males is 16-20 and for females is 18-22, these two groups templates can both be used with ppg template and substituted into the final calculation.

## C. Analysis

The five sets of sqi used for fusion correspond to the results in 7.3, with method 3 optimised to give the best results. In the meantime, I have drawn the scatter plot 6.15 for a discussion of the sqi itself. The orange dots represent the number of cases where the ECG signal is of higher quality, while the blue dots represent the number of cases where the PPG signal is of higher quality or both have the same signal quality.



Figure 6.15: Scatter plot for SQI analysis

When I reduced the number of ecg signal better segments to below 193 and increased the number of ppg segments better or the same quality signal segments to above 437, there was a reduction in the final AE. So I presume that our overall fusion is better when the blue and orange counterparts are within the interval shown above. This may give me some reference for the future process of calculating SQIs and examining theis effects.

---

# Chapter 7: Data Fusion Using Machine Learning

---

## 7.1 Introduction

This part mainly focuses on the data fusion part using the data from previous parts including the features from ECG and PPG signal and the weight coefficients SQL. Data fusion is a process of combining many different kinds of information which contain the same messages together to get a single, consistent and clean representation[26]. To be specific, the work includes data fusion using methods based on machine learning. At first, the idea is to use the neural network like the convolution neural network, but later it is found not available for some reasons which will be explained in detail. The further attempt is about the regression method. The following sections will explain each of the points here in detail. Figure 7.1 shows the basic flow of this part.

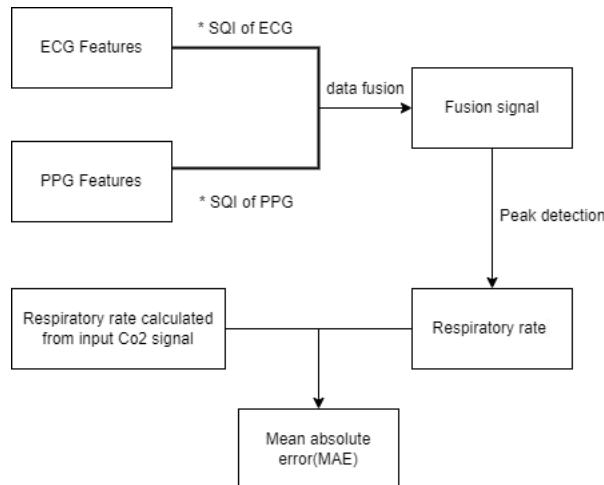


Figure 7.1: The flowchart of fusion

## 7.2 Traditional method of sensor signal fusion

As A. John[13] mentioned in his article, the fusion is implemented based on the ECG, PPG wavelet coefficients and cSQLs, which is a kind of weighted additive fusion of the sequence.

$$F[n] = \sum_{i=1}^m W_i[n] S_i[n] \quad (7.1)$$

, where  $F[n]$  stands for the new sequence of signals,  $W_i[n]$  stands for the cSQLs, and  $S_i[n]$  stands for the d2 and d3 wavelets of ECG and PPG signal.

Here in our project, the ECG and PPG wavelet coefficients are corresponding to the features extracted

---

from ECG and PPG signal, the cSQLs are corresponding to the SQL coefficients. Therefore, a new representation can be get using our data and A.John's idea.

$$F[n] = \sum_{i=1}^m (W_i[n]S_i^{ecg}[n] + (1 - W_i[n])S_i^{ppg}[n]) \quad (7.2)$$

,where both SQL coefficients are normalized, so the sum of SQL coefficients in the same group is 1.

However, this weighted additive fusion is not so accurate since the SQL is a signal quality index, which only tells the fact that the signal can be trusted to what extent. The weighted additive fusion is rather rough. It is impossible to extract the trend and outline of the input signal exactly since it contains some noise.

## 7.3 Proposal solutions

In order to make the weight coefficients to be more reliable, we combine the traditional method with machine learning. First, the idea is to use the convolution neural network to do the weighting computing, but some issues are found during our work which will be illustrated in the following subsections. So we step back to find a substitution using regression which is also based on machine learning. We think that combining all the input data together to find the output of the weighted sum will make the results better.

## 7.4 Convolution neural network(CNN)

CNN is our first idea to be used in data fusion. The CNN is widely used in image processing, for its three advantages: 1) will not lose spatial information, 2) too many parameters make it hard to train the parameter, and 3) too many parameters also lead to the overfitting quickly. I think that since this method is available for 2-dimensional data, so it must work in the 1-dimensional data. Then we begin to learn the basic information about CNN. We learned that there are several differences compared with the fully connected neural network.

### 7.4.1 The convolution layer

The first is the convolution layer, in the 2-D aspect, there is a convolution kernel matrix to multiple with the image itself then the kernel matrix will keep moving until all the image data is calculated and saved in a new matrix. The process is visible shown in Figure 7.2. The convolution process is to minimize the number of points meanwhile keep all the key information. In this step, there is also a useful but not necessary process called padding. Without the padding operation, the edge information will be detected far less than the information in the centre. The padding operation is shown in Figure 7.3. In our 1-D project, we tend to regard the ECG and PPG signals as the images in 2-D examples and the convolution kernel is not a matrix but its corresponding weighted parameter. And the padding in 1-D is just adding one zero at both begin and end of the signal.

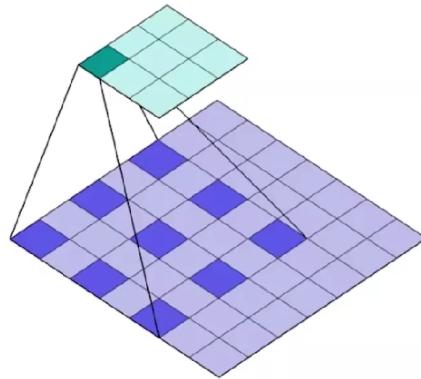


Figure 7.2: The principle of convolution layer in 2-D model

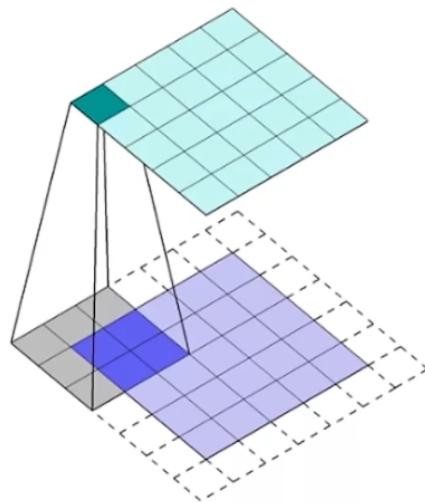


Figure 7.3: Padding operation

#### 7.4.2 The pooling layer

Then the second is the pooling layer, this step can be regarded as a process to extract feature extraction from the output of the convolution layer without much loss. There are two main pooling methods, max pooling and average pooling. It is easy to understand, the first is to get the max value in an area as the representative of this area, and the second is to calculate the average of the area to represent this area. The default area size is a  $2 \times 2$  or  $3 \times 3$  matrix. Figure 7.4 will show an intuitive example. (This is only an example, the input data should be normalized and their range should between 0 and 1 or -1 to 1)

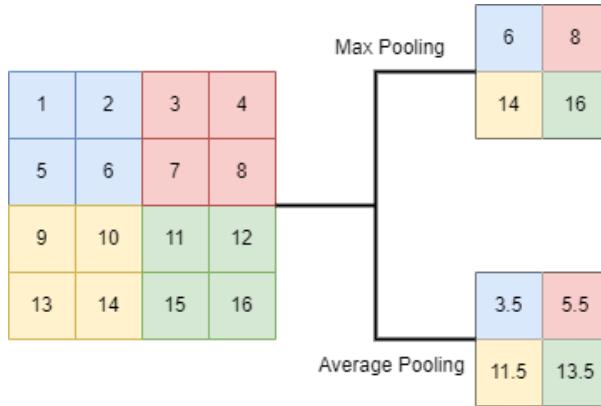


Figure 7.4: The principle of two pooling layer

### 7.4.3 The activation function and fully connected layer

Finally, the work is to choose the suitable activation function to strengthen the features, the usual chooses are sigmoid function, tanh function and Relu function.

$$f(x)_{sigmoid} = \frac{1}{1 + e^{-x}} \quad (7.3)$$

$$f(x)_{tanh} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (7.4)$$

$$f(x)_{Relu} = \begin{cases} 1 & x \leq 0 \\ x & x > 0 \end{cases} \quad (7.5)$$

Then the output should be flattened as the input of the fully connected layer. The later work is the same as a usual neural network to train the parameters using forward propagation and backward propagation.

We have found some existing networks that satisfy the requirements on Kaggle, but we abandon using CNN at last. I will explain it in the following part.

### 7.4.4 Why CNN not used

As we do the feature extraction from the raw data, we found that CNN cannot be used for several reasons. 1) the raw data is reshaped based on the time interval, every 32 seconds is regarded as a piece of input, but we found the data was not the fixed-length after extraction. Since some patients may suffer from a disease which will increase the number of peaks of either ECG or PPG signal. This leads to a fact that the feature points in this time interval may be much more than the other segments. The variable number of points in the samples makes that the input to the CNN is not fixed in length. 2) The size of the dataset is not big enough for a neural network, when we divide the data into 32 seconds segments, there are 630 groups in all. However, the datasets are far larger for most of the neural network model.

## 7.5 Fusion method used in the project – SVR

The fusion method we use in this project is the Support Vector Regression(SVR), which contains an existing method called LinearSVR in Python. We are very confused about the difference between linear regression and non-linear one, it takes some time to figure it out. The following part will tell why we use LinearSVR and why not use non-linear regression.

### 7.5.1 The principle of SVR

In the normal regression, once the point is not on the regression curve then the L2 loss should be calculated. However, in SVR, a tolerant area is introduced. For example, there are some spots in the 2-D plane, and the overall trend of these spots is a linear function, represented by  $f(x) = w^T x + b$ . Then there are two other linear functions  $f(x) + \epsilon$  and  $f(x) - \epsilon$ , the area between these new lines is regarded as the tolerant area. The spots within the area can be regarded as lying on the function  $f(x)$ . It is shown in Figure 7.5, the red solid line represents the regression curve and the two imaginary lines limit an area where once the points lie in, their L2 loss will be zero. The other points not in the tolerant area will calculate the L2 loss and sum these losses up and try to minimize the sum to find the best solution to the regression.

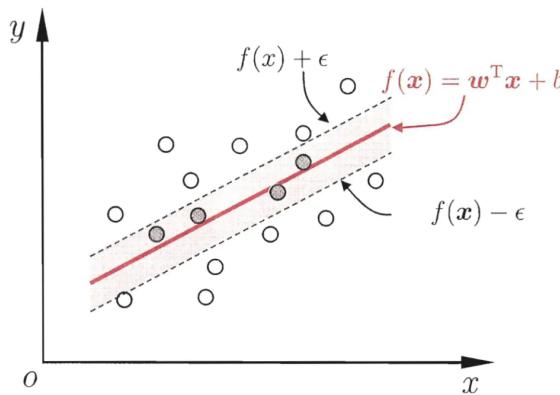


Figure 7.5: SVR principle

### 7.5.2 Why SVR and why linear

The reason why we choose SVR is that the final purpose of our project is to calculate the number of peaks per 32 seconds to get the respiratory rate and then find the difference between the calculated respiratory rate and the real respiratory. So what we need is the exact trend of the signal instead of the exact location of each point at every time slot. The suitable size of tolerate area can make the program converge quickly without affecting the final outputs. Then the reason why we use the linear regression is that we find that the linear regression is not only used for the output that is a straight line in 2-D or a plane in 3-D. For example, the scatter diagram in Figure 7.6 is obvious a quadratic function, whose expression is as below:

$$y = w_1 * x_1^2 + w_2 * x_1 + c \quad (7.6)$$

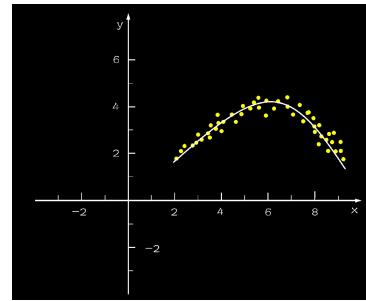


Figure 7.6: An example quadratic function

The  $x_1^2$  term can be replaced by another variable  $x_2$ . In this way, a curve in 2-D space can be discussed in a 3-D schematic diagram as shown in Figure 7.7. Besides, we can add more high-power variable to fit the real trend, according to the Taylor Series, a function can be expressed as we adding endless high-power variables. We use the LinearSVR can add some high-power terms to make the L2 loss as small as possible meanwhile avoid the overfitting.

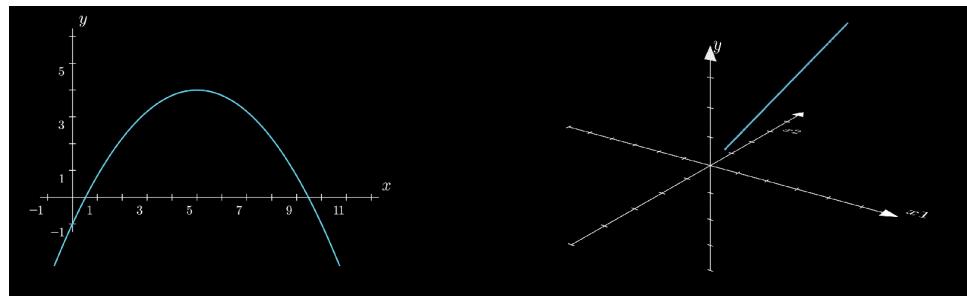


Figure 7.7: SVR principle – from 2-D to 3-D

## 7.6 Data pre-processing

Since the same row in each file has the different length, for example, every line of ECG file, PPG file and Co2 file are different, then padding every row to the same length using the average value of that row shown as Figure 7.8. Then since the maximum column each row is different, and the blank space will be recognized as 'NAN' automatically, so I fill all the blank with '-1' shown as Figure 7.9. When we need to access the data, we will keep reading the number until the '-1' is read.

```
m1=0
sum1=0
for j in range(0, len(data1.iloc[i, :])):
    if data1.iloc[i,j]==-1:
        sum1+=data1.iloc[i,j]
        m1=m1+1
mean1=sum1/m1
for j in range(0, len(data1.iloc[i, :])):
    if data1.iloc[i,j]==-1:
        data1.iloc[i,j]=mean1

m2=0
sum2=0
for j in range(0, len(data2.iloc[i, :])):
    if data2.iloc[i,j]==-1:
        sum2+=data2.iloc[i,j]
        m2=m2+1
mean2=sum2/m2
for j in range(0, len(data2.iloc[i, :])):
    if data2.iloc[i,j]==-1:
        data2.iloc[i,j]=mean2

# # fill the blank block to -1
data3 = pd.read_csv(path3, encoding="GB2312", header=None)
data3.fillna(-1, inplace = True)

data4 = pd.read_csv(path4, encoding="GB2312", header=None)
data4.fillna(-1, inplace = True)

relat100 = pd.read_csv(path5, encoding="GB2312", header=None).iloc[1,:]
relat100.fillna(-1, inplace = True)

data5 = pd.read_csv(path6, encoding="GB2312", header=None)
data5.fillna(-1, inplace = True)

data6 = pd.read_csv(path7, encoding="GB2312", header=None)
data6.fillna(-1, inplace = True)
```

Figure 7.9: Code to fill -1

Figure 7.8: Code to fill mean value

---

## 7.7 The first fusion output

At first, I use the data provided by Qiwei Ke, who mainly uses AM, BW and FM to extract features. I tried to do the fusion using all kinds of combinations, and the output is shown in the chart below. I did not use SQI here since I think SQI weighted coefficients can only affect the output a little but will not influence the overall size. The output in Table 7.1 using the co2 reference value that is directly extracted from the database.

| Signal/AE     | 25%-point | 50%-point | 75%point | mean value |
|---------------|-----------|-----------|----------|------------|
| ECG-AM+PPG-AM | 5.00      | 9.89      | 15.25    | 11.12      |
| ECG-AM+PPG-BW | 5.00      | 9.63      | 14.34    | 10.82      |
| ECG-AM+PPG-FM | 5.62      | 10.03     | 15.24    | 11.29      |
| ECG-BW+PPG-AM | 5.00      | 9.63      | 14.38    | 10.70      |
| ECG-BW+PPG-BW | 4.60      | 8.75      | 13.66    | 10.15      |
| ECG-BW+PPG-FM | 4.99      | 9.60      | 15.19    | 11.08      |
| ECG-FM+PPG-AM | 4.25      | 8.88      | 16.04    | 10.93      |
| ECG-FM+PPG-BW | 4.26      | 7.82      | 13.66    | 9.99       |
| ECG-FM+PPG-FM | 4.83      | 9.38      | 17.12    | 11.57      |

Table 7.1: The fusion output of AM, FM and BW without SQI using approximate reference value

The output in Table 7.2 using the reference value that is calculated as the average value at each time interval is the self-fusion of all the features. I do this work since I think the output is not ideal and I suspect whether the reference value we use is not reasonable.

| Signal/AE     | 25%-point | 50%-point | 75%point | mean value |
|---------------|-----------|-----------|----------|------------|
| ECG-AM+ECG-AM | 5.9       | 10.6      | 14.9     | 11.2       |
| ECG-BW+ECG-BW | 5.0       | 9.5       | 14.2     | 10.4       |
| ECG-FM+ECG-FM | 3.1       | 6.9       | 12.5     | 8.8        |
| PPG-AM+PPG-AM | 4.7       | 9.4       | 15.3     | 10.8       |
| PPG-BW+PPG-BW | 3.9       | 7.2       | 12.5     | 9.2        |
| PPG-FM+PPG-FM | 4.3       | 8.6       | 16.1     | 10.9       |

Table 7.2: The self-fusion output using average reference value

## 7.8 First attempt to improve – Wavelet threshold denoising

The outputs above are unacceptable, so I begin to guess whether the feature data contains the noise that influences the regression, so I implement a denoising function based on wavelet transform which is called wavelet threshold denoising. The signal can be decomposed as Figure 7.10 shown, the low-frequency part can keep decomposing to get the next layer wavelet coefficients. Since the noise is much smaller than the signal itself, which means that the wavelet coefficients are also very small, so we can set a suitable threshold where the value smaller than threshold is abandoned. Then reconstruct the signal to get a denoising signal. The essence of this operation is to delete the useless part of the signal and strengthen the meaningful part. In this way, we can remove most of the noise. It is a pity that the outputs are worse than the outputs before so I did not use the wavelet threshold denoising anymore in the next work.

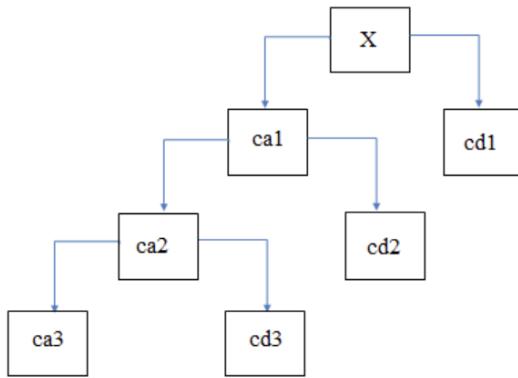


Figure 7.10: Wavelet decomposing, where X is input, ca is low frequency information, cd is high frequency information

## 7.9 Second attempt to improve – Using the traditional method

After the denoising, I wonder whether the problem is the regression code I have written, so I decide to use the traditional method to verify my idea.

The first method is to multiple the whole ECG and PPG signals with the SQIs to get two curves that have the same trend as the original signals but different amplitudes. Then add the corresponding value of the new curves together to get a final curve. Finally, calculate the number of peaks in the final curve.

The output of the first method is far worse than the output in Table 7.1. And we find out the reason is that due to the characters of the feature extraction, the peaks of ECG and PPG may not match. It exists some time shifts, the peaks may be at the same place as the valleys. This will make the sum at this time slot be zero approximately, this will reduce the number of peaks it should have. This is shown in Figure 7.11

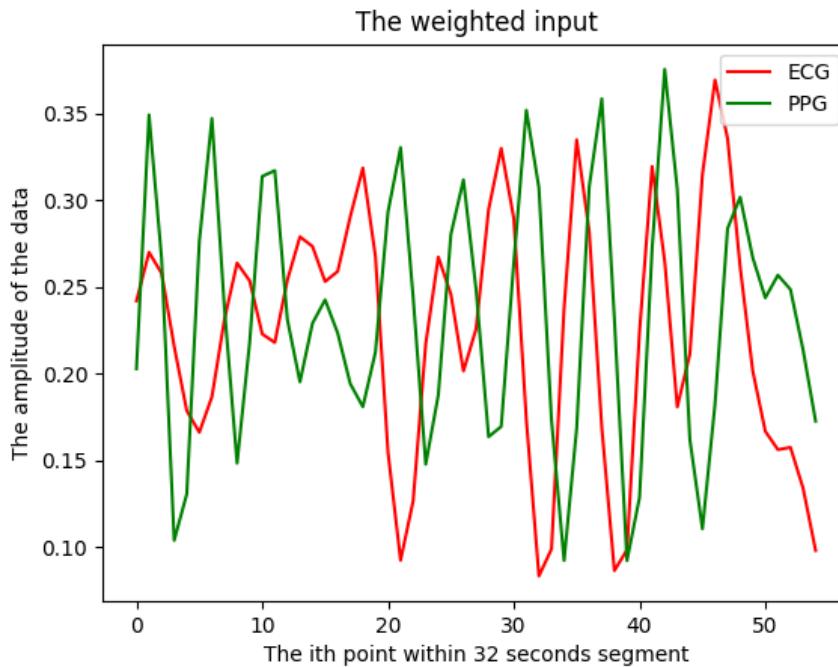


Figure 7.11: Two curves stands for features of ECG and PPG, most of the peaks are staggered

To figure out the problem above, we propose the second method, which is to calculate the corresponding number of peaks in both ECG and PPG signals, then multiple the number of peaks with the corresponding weight coefficients to get a final number of peaks.

The output of the second method has little difference from the output in Table 7.1. So I preliminarily consider that my regression method has few problems.

## 7.10 Reflection on the feature extraction

Since all the bad performances before, we begin to reflect on whether the feature extraction is proper. We use each feature signal itself to fuse with itself and find that the MAE is rather big (Details shown in section 5.3.5). So we use to try to find some new extraction method as section 5.4 mentioned to replace the previous method and hope to find some difference in the final output.

## 7.11 Final attempt

I use Zhipeng Xie's extraction outputs and several groups of SQIs to run the code and find that the AE is much smaller than the previous ones.

| SQIs/AE | 25%-point | 50%-point | 75%point | mean value |
|---------|-----------|-----------|----------|------------|
| SQI-1   | 0.73      | 2.10      | 5.26     | 4.35       |
| RQI-1   | 0.73      | 2.10      | 5.60     | 4.42       |
| RQI-2   | 0.72      | 1.92      | 5.54     | 4.34       |
| SQI-2   | 0.73      | 2.10      | 5.34     | 4.35       |
| SQI-3   | 0.73      | 2.06      | 5.34     | 4.33       |

Table 7.3: The fusion based on Xie's extraction using different weighted coefficients, where SQI-1 is calculated based on the original raw data using the method in section 6.2.1; RQI-1 is calculated based on the extraction features using the method in section 6.2.2; RQI-2 is the updated version of RQI-1; SQI-2 is calculated based on the extraction features using the method in section 6.2.3; SQI-3 is the updated version of SQI-2.

I think the SQI-1 or SQI-3 performed the best. And their corresponding AE scatter diagrams are shown in Figure 7.12 and Figure 7.13.

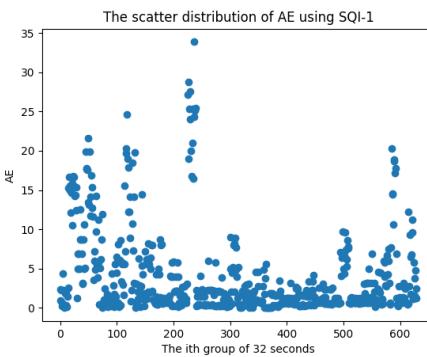


Figure 7.12: The scatter diagram for SQI-1

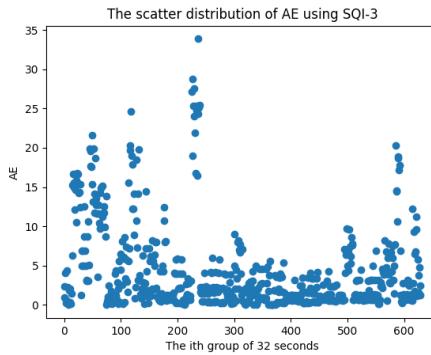


Figure 7.13: The scatter diagram for SQI-3

## 7.12 Unpredictable problems

Firstly, when using the data output from the wavelet threshold denoising, we meet a weird problem. There will have an error when the program is running, and the solution is to do the 'save as' operation to the newly generated data, then the errors will disappear. This problem happened frequently at the work. We guess that is the system file format problem that we cannot fix. Secondly, it is also about the format of the data, since I use Python to implement the regression algorithm, and Python read

---

the .csv file in its own special way. To run the code successfully, we need to check the maximum columns of the files(ECG, PPG and Co2) should be the same. If it is not the same, we should add '-1' in some of them to make the maximum column the same.

## 7.13 Conclusion

In this section, we use the feature extractions and weighted coefficients to do the data fusion based on LinearSVR. We have tried two methods and finally abandon the CNN to choose the LinearSVR. During the code implementation, we meet some data format problems which is due to the characteristics of those files. However, after using different feature extractions and weighted coefficients, we get an acceptable AE.

---

# Chapter 8: Breath Rate Detection And Data Processing

---

## 8.1 Introduction

The main task of this section is to do some auxiliary work in service of the first three sections, which include an algorithm for detecting the number of breaths within a fragment, as well as data format conversion and interpolation tasks.

## 8.2 Breath Number Detection

This part's task is to detect how many breaths happen in a fixed period and then calculate the breath per minute. Here I choose to use the three-point detection, it can effectively avoid double peaks and multiple peaks, resulting in more accurate statistics.[\[5\]](#)

### 8.2.1 Three point detection

Three-point detection means that three points are needed to detect one breath that happened. These three points are peak valley and zero point. Our signals are normalized between 0 and 1 so we can use the cross-point of  $y=0$  and the signal as a zero point. If the signal is not normalized  $y=\text{mean}(\text{signal})$  can be used as a zero axis to find the zero point. We cannot use Matlab's built-in 'fzero' function to find the zero point because our signals are discrete points. My method of judgement is that if the value of the first point is greater than the mean value of the signal and the second point is less than the mean value of the signal, then the centre of the horizontal coordinate of these two points is taken as the zero point and vice versa.

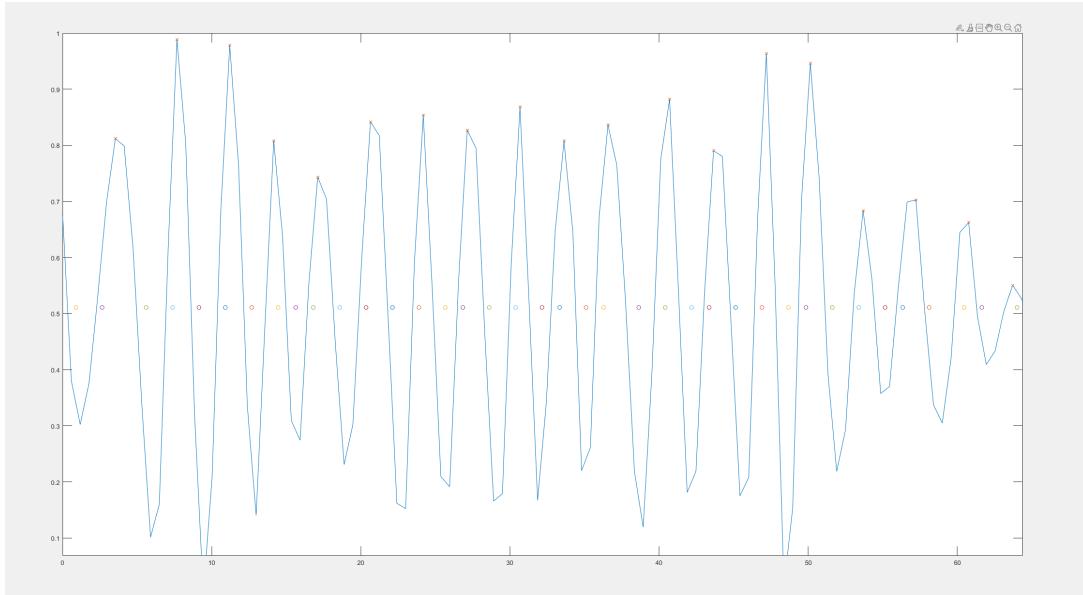


Figure 8.1: An example of three point detect

A breath is judged as follows, if there is a peak or valley between the (i)th zero point and the (i-2)nd zero point we consider it to be a breath.

### 8.2.2 Result Analysis Of Three Point Detection

Overall the results of this detection method are acceptable if the value of the 'zero axis' of the signal can be found accurately. In some signals where the value of the 'zero axis' is defined as the mean value of the signal, there are often cases where both points are above the mean resulting in the zero point not being detected and the breath being missed. Thus the overall performance of this algorithm for the 42 data sets is similar to that of the constrained peak detection method but the three-point detection method is effective in avoiding multiple peak conditions.

## 8.3 Data Format Conversion

Since the signal extraction is done in Matlab but the signal fitting is done in python, we need to convert the data in the mat file into a CSV file to facilitate the data reading in the signal fitting step and fill different signals in the same segment to the same length with itself's average signal value.

---

## Chapter 9: Future Work

---

Firstly, we are only using one database from the web in our project, so many of the problems we encounter are likely to be duplicated or associated with problems that originate in the database itself. We are therefore considering exploring and experimenting with other databases in the future. For the work we have already done, we will focus on the future work of the feature extraction, SQL calculation and data fusion phases.

For signal pre-processing and feature extraction part, although several methods were used to optimise our extracted signals, there also exist some ways to optimise the signal to get a lower AE value. For example, after the peak detection is finished a filter can be added to select the signal whose heart rate is bigger than 40bpm and smaller than 180bpm so we can remove some signals with large noise. On the other hand, an FIR filter can be added after EEMD decomposition to flit the signal and keep the signal frequency between 0.1Hz and 0.6Hz. The signal in this frequency range contains the breathing signal and by filtering it we can remove some of the noise to make the calculation more accurate.

When using the original signal for SQL calculations, due to the irregularity of the signal itself or the strong interference, even after some pre-processing it is not possible to extract all QRS features of ecg and pulse peak features of ppg, which inevitably causes some errors when calculating the fragment correlation coefficients. Optimising the correlation algorithm and adding the detection of other eigenvalues of the ECG signal, such as P-wave and T-wave, are algorithms that can be tried in the future. In method 2, the curve of SQL affected by different degrees of noise is not smooth enough, and it may be better if a suitable function can be found during curve fitting to add coefficients to the original SQL. Meanwhile, the fourth method mentioned in Chapter 6 can also be explored next, if an SQL can be obtained on each sample before averaging the SQL of each segment, it might be more representative for the signal quality of the segment.

After communication with the supervisor, we found that the CNN might be available if we combine many database together and do some zero padding to all the input until they have the same length. However, there are only several signal among 630 groups of data, the zero padding means that many meaningless will in the data. If this is regarded as the input of CNN, then many neural cells may die out soon. It also deserves discussing that many the neural cell die out will have influence on the final performance. However, since CNN is much more reliable than the regression, it also worth having a try. Another point is that there should have a comparison experiment using the signal contains Gaussian noise and the pure signal to figure out the final AE seperately. This will help to determine which group of weight coefficients are better. In this article, we can only analysis according to the AE without adding noise, which will be less convincing.

---

## Chapter 10: Conclusion

---

In conclusion, poor data interpretation caused by noise and motion in IoT wearable sensors can be improved by the solutions proposed in the article, combined with integrated physiological data. Firstly, our project offers two feature extraction methods to find features from raw ECG and PPG signals. While the EEMD method is mainly used due to its good performance. Secondly, we use three SQL calculation methods to obtain the weighting coefficients. Then the machine learning algorithm LinearSVR was used to do the data fusion and the three-point peak detection method was used to find the respiratory frequency, by which the final AE was controlled in the range of 4.33 to 4.42.

---

## References

---

1. Charlton, P. H. *et al.* An assessment of algorithms to estimate respiratory rate from the electrocardiogram and photoplethysmogram. *Physiological measurement* **37**, 610 (2016).
2. Charlton, P. H. *et al.* Extraction of respiratory signals from the electrocardiogram and photoplethysmogram: technical and physiological determinants. *Physiological measurement* **38**, 669 (2017).
3. Pimentel, M. A., Charlton, P. H. & Clifton, D. A. in *Wearable electronics sensors* 241–262 (Springer, 2015).
4. Charlton, P. H. *et al.* Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review. *IEEE reviews in biomedical engineering* **11**, 2–20 (2017).
5. Orphanidou, C. Derivation of respiration rate from ambulatory ECG and PPG using ensemble empirical mode decomposition: Comparison and fusion. *Computers in biology and medicine* **81**, 45–54 (2017).
6. Meredith, D. J. *et al.* Photoplethysmographic derivation of respiratory rate: a review of relevant physiology. *Journal of medical engineering & technology* **36**, 1–7 (2012).
7. Johansson, A. Neural network for photoplethysmographic respiratory rate monitoring. *Medical and Biological Engineering and Computing* **41**, 242–248 (2003).
8. Karlen, W., Raman, S., Ansermino, J. M. & Dumont, G. A. Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE Transactions on Biomedical Engineering* **60**, 1946–1953 (2013).
9. Mirmohamadsadeghi, L. & Vesin, J.-M. Respiratory rate estimation from the ECG using an instantaneous frequency tracking algorithm. *Biomedical Signal Processing and Control* **14**, 66–72 (2014).
10. Wu, Z., Huang, N. E. & Chen, X. The multi-dimensional ensemble empirical mode decomposition method. *Advances in Adaptive Data Analysis* **1**, 339–372 (2009).
11. Madhav, K. V., Ram, M. R., Krishna, E. H., Komalla, N. R. & Reddy, K. A. *Estimation of respiration rate from ECG, BP and PPG signals using empirical mode decomposition* in 2011 IEEE International Instrumentation and Measurement Technology Conference (2011), 1–4.
12. Arlene John<sup>1</sup>, B. C. & John, D. A Generalized Signal Quality Estimation Method for IoT Sensors (2020).
13. John, A., Redmond, S. J., Cardiff, B. & John, D. A Multimodal Data Fusion Technique for Heartbeat Detection in Wearable IoT Sensors. *IEEE Internet of Things Journal* (2021).
14. Christina Orphanidou, T. B., Peter Charlton, D. C. & David Vallance, L. T. Signal-Quality Indices for the Electrocardiogram and Photoplethysmogram: Derivation and Applications to Wireless Monitoring. *Biometrical and health informatics* **19**, 832–848 (2015).
15. Yu-Chia Yang<sup>1</sup>, W.-K. B., Yi-Cheng Lo<sup>1</sup>, A.-Y. ( W. & Lu, S.-J. ECG-aided PPG Signal Quality Assessment (SQA) System for Heart Rate Estimation (2020).
16. Banerjee, T. P. & Das, S. Multi-sensor data fusion using support vector machine for motor fault detection. *Information Sciences* **217**, 96–107 (2012).
17. Begum, S., Barua, S. & Ahmed, M. U. Physiological sensor signals classification for healthcare using sensor data fusion and case-based reasoning. *Sensors* **14**, 11770–11785 (2014).
18. Gu, Q., Jiang, S., Lian, M. & Lu, C. Health and safety situation awareness model and emergency management based on multi-sensor signal fusion. *Ieee Access* **7**, 958–968 (2018).

- 
19. Karlen, W., Turner, M., Cooke, E., Dumont, G. & Ansermino, J. M. *CapnoBase: Signal database and tools to collect, share and annotate respiratory signals* in *2010 Annual Meeting of the Society for Technology in Anesthesia* (2010), 27.
  20. Charlton, P. H., Villarroel, M. & Salguero, F. Waveform analysis to estimate respiratory rate. *Secondary Analysis of Electronic Health Records*, 377–390 (2016).
  21. Pimentel, M. A. F. et al. Toward a Robust Estimation of Respiratory Rate From Pulse Oximeters. *IEEE Transactions on Biomedical Engineering* **64**, 1914–1923 (2017).
  22. Goldberger, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation* **101**, e215–e220 (2000).
  23. Rickards, C. A., Ryan, K. L. & Convertino, V. A. Characterization of common measures of heart period variability in healthy human subjects: implications for patient monitoring. *Journal of clinical monitoring and computing* **24**, 61–70 (2010).
  24. Huang, N. E. et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences* **454**, 903–995 (1998).
  25. Rankawat, S. A. & Dubey, R. Robust heart rate estimation from multimodal physiological signals using beat signal quality index based majority voting fusion method, *Biomed. Signal Process. Control* (2017).
  26. Bleiholder, J. & Naumann, F. Data fusion. *ACM computing surveys (CSUR)* **41**, 1–41 (2009).