Signal Analysis and Classification of Photoplethysmography (PPG) Waveforms for Predicting Clinical Outcomes

Student: Jingkai Wang
CID: 01910969
Supervisor: Dr. Bernard Hernandez Perez
Co-Supervisor: Dr. Pantelis Georgiou

A Thesis submitted in fulfilment of requirements for the degree of
Master of Science
Communications and Signal Processing
of Imperial College London

Department of Electrical and Electronic Engineering
Imperial College London

Abstract

The photoplethysmogram (PPG) is a signal that measures pulsatile volume of blood in tissue and is collected by portable, non-invasive pulse oximeters. PPG signals can be used to predict the severity of dengue patients during dengue epidemics. However, raw PPG waveforms are susceptible to artifacts and noise. The classification of PPG signal quality is therefore crucial. In this work, PPG waveforms are classified into three categories according to their quality, and seven signal quality indices (SQIs) are applied to extract the features of each segment of PPG. At the same time, the spectrogram of each PPG sample is calculated as another feature. With the help of k-means clustering on the SQI features, each sample is labelled with a signal quality as ground truth. Apply an autoencoder to reduce the dimension of SQI features to observe the data samples in 2-D, and compare the PCA method. Using the above PPG samples and labels as dataset, apply decision tree, random forest, and AdaBoost classifiers on raw PPG data or SQI features. Build a convolutional neural networks (CNN) model for 2-D spectrograms to learn the training set data, and predict the test set data. Experimental results show that: SQIs are effective PPG features that facilitate quality labelling and dimensionality reduction. The spectrogram is also an effective PPG feature, and the CNN has good stability, good flexibility, high accuracy and fast speed to be an ideal PPG signal quality classification method.

Acknowledgment

I would like to extend my sincere gratitude to my supervisor Dr. Bernard Hernandez Perez for his support and encouragement throughout this project. He often cared about my progress, discussed the problems I encountered, and gave me practical suggestions in a timely manner. His careful and patient guidance was always unreserved, and he enlightened me on how to learn new skills and how to solve problems. I admire his professional skills in signal processing and machine learning. His meticulous attitude towards science has deeply influenced me.

I was also fortunate to be helped by Stefan Karolcik, who have provided valuable resources, pointed out the areas in my work that needed improvement and shared valuable project management strategies.

I would also like to thank Dr. Imad Jaimoukha and Philippos Arkis Hadjimarkou for their kind support.

Finally, I would like to express my heartfelt thanks to my beloved parents for their selfless support, encouragement, and love across the ocean during the pandemic.

Contents

List of Figures

List of Tables

Abbreviations

PPG: Photoplethysmogram

ECG: Electrocardiogram

HR: Heart Rate

SQI: Signal Quality Index

SNR: Signal-to-Noise Ratio

ZCR: Zero-Crossing Rate

MCR: Mean-Crossing Rate

MSQ: Mean Signal Quality

PCA: Principal Component Analysis

SVD: Singular Value Decomposition

ANN: Artificial Neural Networks

CNN: Convolutional Neural Networks

MSE: Mean Squared Error

LR: Learning Rate

Chapter 1

Introduction

1.1 Overview

Photoplethysmogram (PPG) is a non-invasive circulatory signal related to the pulsatile volume of blood in tissue and is typically collected by pulse oximeters [3]. Each year, an estimated 390 million dengue infections occur around the world, of which 96 million manifest clinically (with any severity of disease) [13] [14]. Effective identification of severe cases is crucial. Pulse oximeters are low-cost wearable devices that would monitor patients during epidemics to support medical triage and management in severe dengue.

Our clinical partners, Oxford University Clinical Research Unit (OUCRU) in Vietnam has been collecting prospective clinical data from patients with dengue [15]. These data include 6,000+ hours of continuous raw PPG and ECG waveforms collected during hospitalisation. However, these raw data collected via pulse oximeters can be subject to artefact and noise, and is very susceptible to patient movement; therefore, cannot be used directly to provide effective medical advice.

It is extremely critical to distinguish the quality of the PPG signal and extract the effective signal of high quality. We can build a model to segment the PPG signal and classify each segment. For the raw data without quality labels, how to automatically label it is the problem of our research.

1.2 Aim

We aim to analyse the raw PPG waveforms and develop signal quality indices (SQIs) to label the segmented data and visualise the signal quality clusters in 2-D space.

The ultimate goal is to achieve automatic labelling of PPG quality via classification methods, when a dataset with both the input features (SQIs, raw data, spectrograms) and quality outcomes (classes) are ready.

1.3 Report Outline

- Chapter 2 will cover biomedical signal processing, acquisition of PPG signals, the characteristics and medical applications, and quality of PPG signals.
- Chapter 3 will review related work in other papers, including PPG signal quality analysis and classification methods.
- Chapter 4 will show our work, including the design steps of PPG signal analysis: data preprocessing, feature extraction, semi-automatic labelling, visualisation, and automatic labelling.
- Chapter 5 will present the experimental design, implementation, results, and discussion.
- Chapter 6 will draw conclusions and talk about future work.

Chapter 2

Background

This chapter describes the application, acquisition, characteristics, issues, and quality of PPG signals.

2.1 PPG Signal

Pulse oximeter is a commonly used sensor in medicine and has the advantages of being cheap, non-invasive, portable and small. PPG signals are common biomedical signals collected from pulse oximeters to estimate the skin blood flow using infrared light [1]. PPG is widely used to measure the heart and respiratory rates [16], oxygen saturation [17] [18], blood pressure [19], cardiac output [20], and for assessing autonomic functions [1].

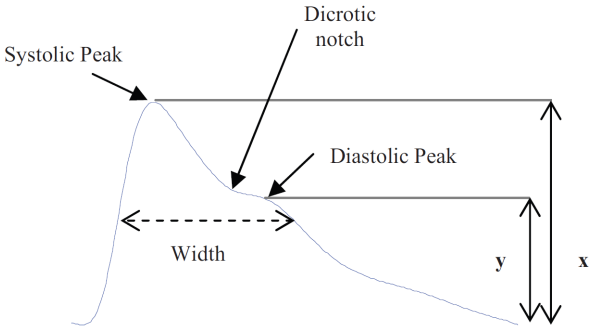


Figure 2.1: PPG pulsatile phases [1]

A typical PPG pulse as in Fig. 2.1 consists of two phases: (1) the rising edge of the

pulse called the anacrotic phase, and (2) the falling edge of the pulse called the catacrotic phase [1]. The first phase is about systole, and the second phase is about diastole and wave reflections from the periphery. A dicrotic notch is usually seen in the catacrotic phase of subjects with healthy compliant arteries [1].

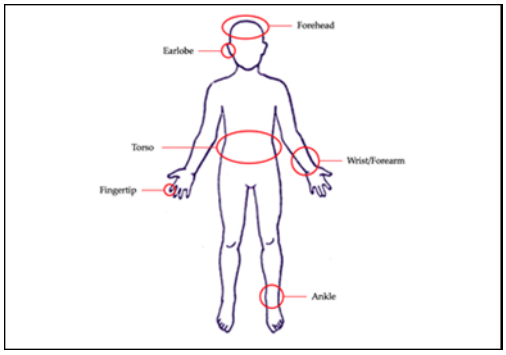


Figure 2.2: Common measurement sites for PPG [2]

Wearable PPG sensors can only be placed in specific body locations, as shown in Fig. 2.2. However, different measurement locations have different accuracies. While the use of specific body locations such as fingers, earlobes and forehead is most common, researchers are considering using other body locations for more convenient alternatives [2].

2.2 Signal Quality

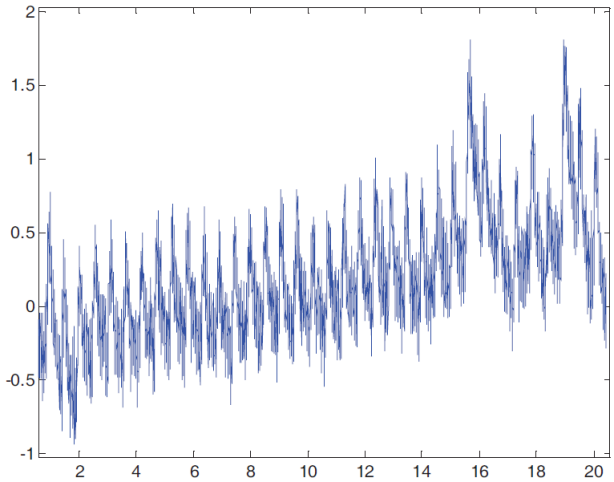
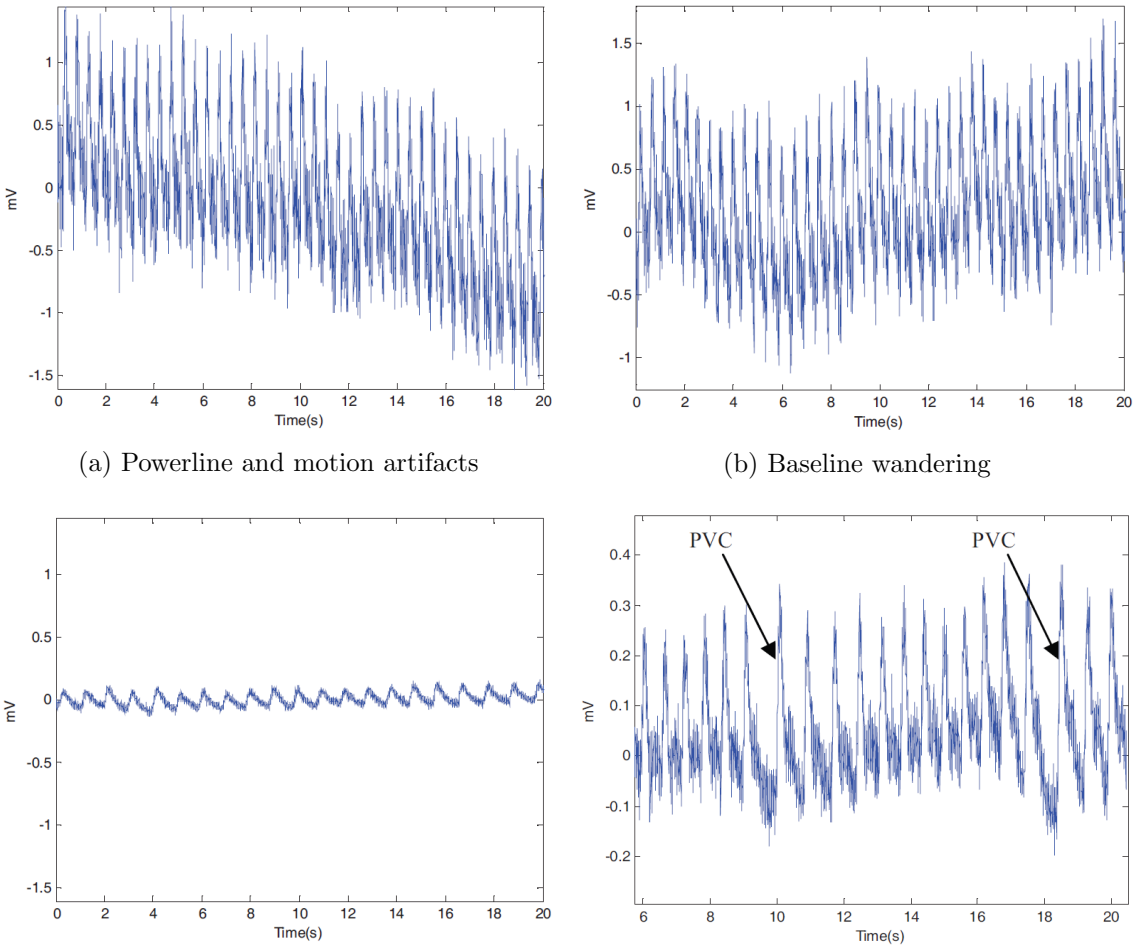


Figure 2.3: Artifacts in one PPG signal [1]

The quality of the PPG signal depends on the location and the properties of the subject's skin at measurement, including the individual skin structure, blood oxygen saturation, blood flow rate, skin temperatures and the measuring environment [1]. These factors produce several types of additional artifacts that may be included in the PPG signal. This may affect the extraction of features and thus the overall diagnosis [1]. Fig. 2.3 shows a PPG sample with different artifacts: motion artifacts, muscle artifact, arrhythmia, high frequency artifact, and low amplitude [1]. Fig. 2.4 shows common artifacts as main challenges in processing the PPG signals: powerline interference, motion artifact, baseline wandering, low amplitude, and premature ventricular contraction [1].



(a) Powerline and motion artifacts

(b) Baseline wandering

(c) Low Amplitude

(d) Premature Ventricular Contraction

Figure 2.4: Different artifacts in PPG waveforms [1]

PPG samples can be manually annotated into three groups in a research [3]:

1. Excellent for diagnosis: PPG signals with salient systolic/diastolic waves and dicrotic notches.

2. Acceptable for diagnosis: PPG signals where the systolic/diastolic waves and dicrotic notches are not obvious but heart rate can be determined.

3. Unfit for diagnosis: PPG signals where the heart rate cannot be determined and neither the systolic/diastolic waves nor the dicrotic notches can be distinguished.

The examples of above categories are shown in Fig. 2.5. The classes of PPG quality in our work refers to this annotation, but it is slightly different from this one.

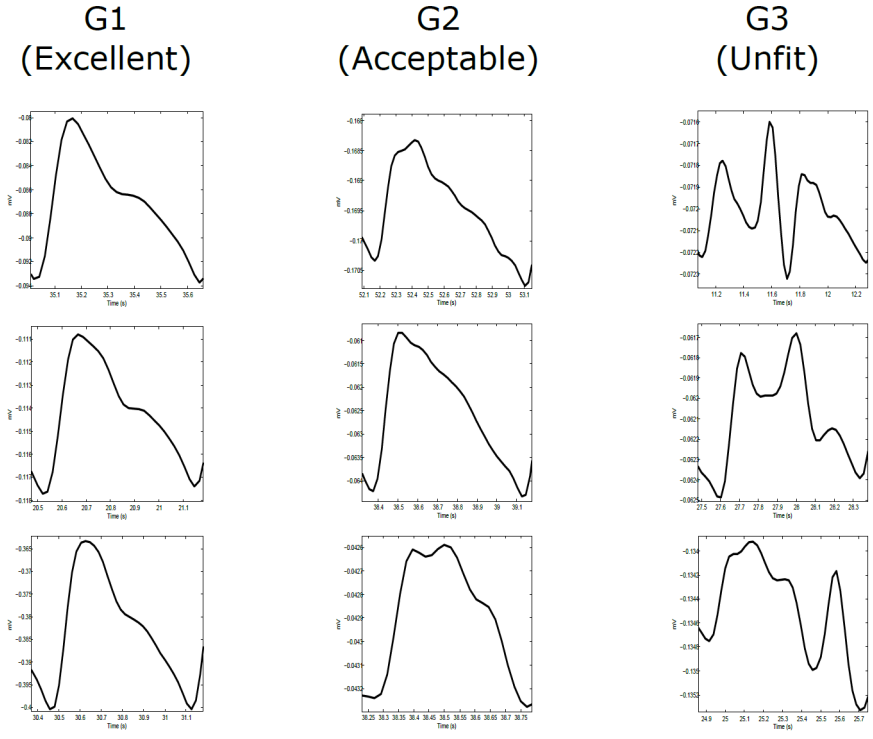


Figure 2.5: Annotated PPG qualities [3]

This chapter provides a preliminary background to the properties and characteristics of PPG signals. The next chapter will cite more papers to introduce different PPG signal analysis methods.

Chapter 3

Literature Review

This chapter will explore the literature related to this project. These literatures either describe the properties of PPG waveforms, introduce the strategies that can be referenced for signal feature extraction, or recommend methods that work well for classifying signal quality. Our method is more or less inspired by these literatures.

As shown in Fig. 3.1 in [1], commonly used structure for PPG diagnostic system consists of three stages: (1) preprocessing stage to emphasise the desired waves; (2) feature extraction stage to detect the desired waves; (3) classification and diagnosis stage to find an index or a measure using the extracted features. Our process design basically refers to these steps.

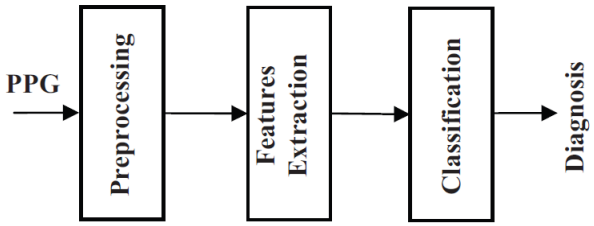


Figure 3.1: Structure of PPG diagnostic system [1]

Since the rapidly increased applications of portable devices to collect PPG signals, developing optimal signal quality indices (SQIs) is crucial to classify the signal quality from these devices [3]. Eight SQIs were calculated and evaluated: perfusion, kurtosis, skewness, relative power, non-stationarity, zero crossing, entropy, and the matching of

systolic wave detectors in Elgendi's work [3]. Here are a few SQIs that we have chosen. He also annotated the PPG signals as excellent, acceptable, and unfit.

Orphanidou et al. presented a signal quality index (SQI) algorithm which is intended to evaluate whether reliable heart rates (HRs) can be obtained from electrocardiogram (ECG) and photoplethysmogram (PPG) signals collected using wearable sensors [4]. The flowchart of the algorithm is shown in Fig. 3.2. The output of the algorithm is delivered in a binary format, "good" (i.e., a reliable HR can be derived) and "bad" (i.e., a reliable HR cannot be derived) to simplify interpretation and facilitate applicability [4].

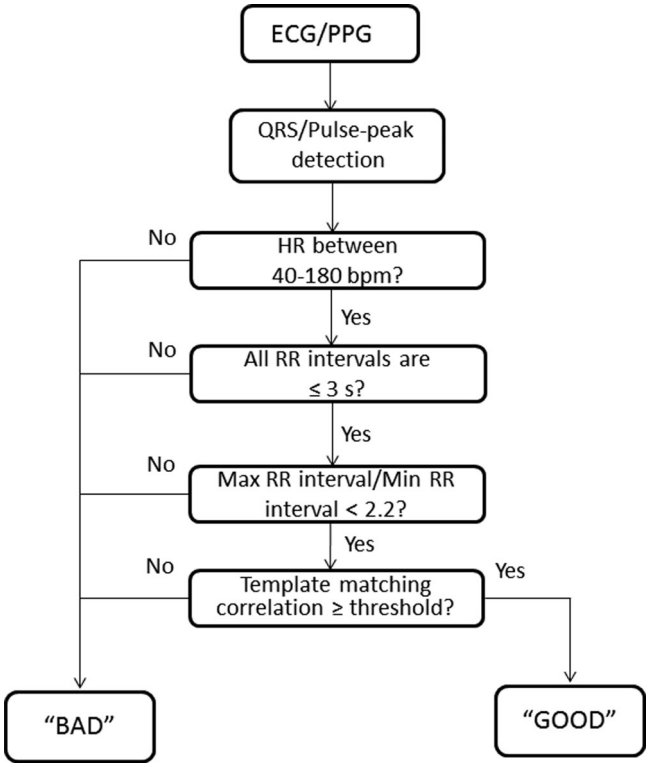


Figure 3.2: Flowchart of SQI algorithm [4]

Li et al. defined four individual SQIs: direct matching SQI, linear re-sampling SQI, dynamic time warping SQI, and clipping detection SQI [21]. They used dynamic time warping (DTW) to stretch each beat to match a running template, and combine it with several other features related to signal quality, including correlation and the percentage of clipped beat. The features were then passed to a multi-layer perceptron neural network to learn the relationships between the parameters in the presence of good- and bad-quality

pulses [21].

Pradhan et al. selected SQIs as follows: number of peaks by Billauer's algorithm, number of zero-crossings, accelerometer features, correlogram features, median noise ratio per pulse, median relative power per pulse, and standard deviation of Shannon energy per pulse [22]. There are several SQIs that we have chosen from here. Five classifiers were evaluated using the annotated dataset. The result shows that Random forest performs the best, with accuracy of 74.5%. The Decision tree has accuracy of 66.9%, while the other three classifiers are lower: Naive Bayes 63.6%, Multi-class SVM 43.5%, and k-nearest neighbour 42.9 % [22]. That is why we choose random forest and decision tree as two of our classifiers.

This chapter reviews some relevant literature that will be helpful in our system design in the next chapter.

Chapter 4

Analysis and Design

In this chapter, we introduce the analysis methods and design ideas to achieve the goals of PPG signal analysis and classification. The following designs are covered in this chapter: preprocessing of raw data, feature extraction, clustering for semi-automatic quality labelling, dimensionality reduction for visualisation and classification for automatic labelling. The flowchart of these designs is shown in Fig. 4.1.

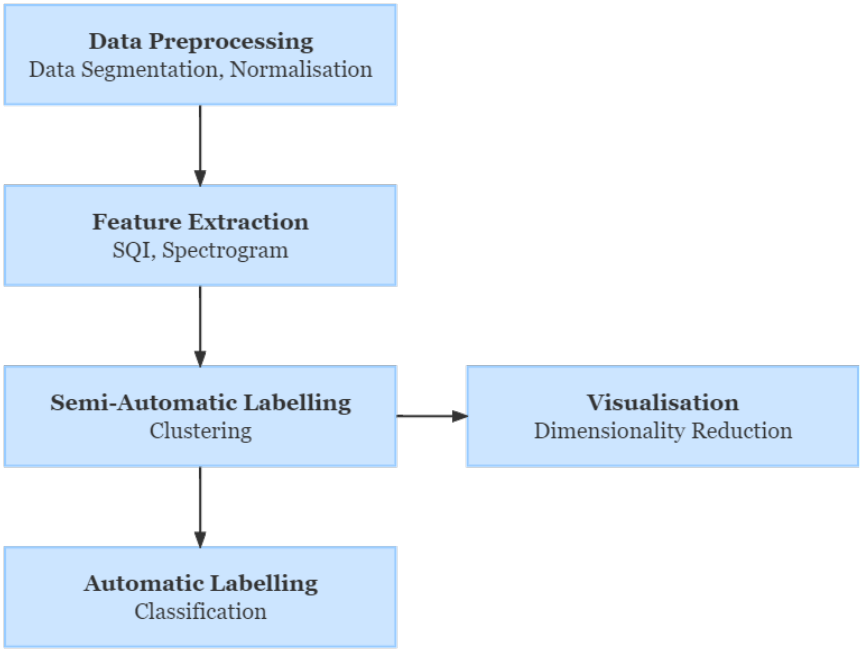


Figure 4.1: Design of PPG signal analysis and classification

4.1 Data Preprocessing

The data preprocessing stage is essential and the first step in PPG signal analysis, whether for clustering or classification. PPG signals are typically collected hundreds or thousands of times per second and thus too long when it is acquired continuously for tens of hours. To facilitate analysis, the raw data is split into segments of equal length, and each segment is treated as a sample. Since the range of data from different patients may vary, normalisation is applied before segmentation.

4.1.1 Data Segmentation

As discussed above, raw PPG signal is long and difficult to handle directly. Also, the beginning and end of the data may be affected by the power on/off of the pulse oximeter hardware or the acquisition process, resulting in distortion and abnormality. To solve these problems, we trim the head and tail of the signal a few minutes each, and split the remaining signal into *windows* of the same length. The trimmed signal is negligible over the entire signal length, while the signal in each window is long enough to show multiple pulses (cycles) and is an indicator of signal quality. In subsequent processing, each window is treated as a sample.

4.1.2 Normalisation

In order to overcome the problem that different patient data may come from different pulse oximeters and the data range may differ, we normalise the signal after trimming the head and tail, and then segment it into windows. Instead of using standardisation that map the original data to zero mean and unit standard deviation, we apply normalisation (min-max scaling) to preserve the shape of the original signal [23]. The calculation is

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

where x is each original data point, $x_{min} = 0$ and $x_{max} = 1$. The normalised data is therefore scaled in the range of $[0, 1]$.

4.2 Feature Extraction

To facilitate PPG quality labelling and classification and also reduce the amount of raw data, we "compress" the original PPG signal to extract useful features and discard redundancy. Efficient ways to represent PPG signal windows are to represent them in the form of Signal Quality Index (SQI) or spectrogram. Both methods are described in detail below.

4.2.1 Signal Quality Index (SQI)

The shape, trend and quality of a PPG signal can be characterised by different indicators. These characterisations of signal quality are called Signal Quality Indices (SQIs). Seven SQIs are applied and calculated [3] [24] [25]. The definition and mathematical representations are listed below:

1. Skewness

Krishnan, Elgendi, et al. found skewness as an effective indicator of corrupted PPG signals [26] [3]. Skewness is a measure of how much the probability distribution of a real-valued random variable deviates from the normal distribution [27]. The skewness of an ideal normal distribution is zero. The probability distribution with its tail on the right side is a positively skewed distribution, and the one with its tail on the left side is a negatively skewed distribution [27].

The skewness of a random variable X is the third standardised moment, i.e.

$$skewness = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{s_x}\right)^3 \tag{4.2}$$

where $\bar{x}$ and s_x are the empirical estimate of mean and standard deviation of data point x respectively, and N is the number of data point in the windowed PPG.

2. Kurtosis

Kurtosis is a useful indicator for PPG signal quality, discovered by Elgendi, Selvaraj, et al [3] [28]. Kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable [29]. Probability distributions with large kurtosis exhibit tail data exceeding the tails of the normal distribution, while the ones with low kurtosis exhibit tails that are less extreme than the tails of the normal distribution [29].

The kurtosis of a random variable X is the fourth standardised moment, defined as

$$kurtosis = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{x}}{s_x}\right)^4 \tag{4.3}$$

where $\bar{x}$ and s_x are the empirical estimate of mean and standard deviation of data point x respectively, and N is the number of data point in the windowed PPG.

3. Signal-to-Noise Ratio (SNR)

Signal-to-Noise Ratio (SNR) is a measure commonly used in communications and signal processing that compares the level of a desired signal to the level of background noise [30]. There are multiple SNR definitions and here we use the ratio of mean to standard deviation of a given PPG measurement [25]:

$$SNR = \begin{cases} 0, & \text{if } \sigma = 0 \\ \frac{\mu}{\sigma}, & \text{otherwise} \end{cases} \tag{4.4}$$

where μ and σ are the mean and standard deviation of PPG signal respectively. In practice, mean and standard deviation are calculated in windowed PPG samples.

4. Zero-Crossing Rate (ZCR)

Zero-Crossing Rate (ZCR) is the rate at which a signal changes from positive to zero to negative or from negative to zero to positive [3]. ZCR is defined as

$$ZCR = \frac{1}{N}\sum_{i=1}^{N}\mathbf{I}\{y(x_i) < 0\} \tag{4.5}$$

where $y(x)$ is the filtered PPG signal x of length N. The $\mathbf{I}\{A\}$ is an indicator function which is equal to 1 if its argument A is true and 0 otherwise.

5. Mean-Crossing Rate (MCR)

The calculation of Mean-Crossing Rate (MCR) is the same as ZCR, while the only difference is the input $y(x)$ to be replaced by unfiltered $(x - \bar{x})$, where $\bar{x}$ is the sample mean of data x [25].

6. Mean Signal Quality (MSQ)

Mean Signal Quality (MSQ) is introduced by Elgendi [3] since different PPG peak-detection algorithms are sensitive to different types of noise [31]. The comparison of how accurate multiple PPG systolic wave detectors isolate each event (such as a beat or noise artifact) provides an estimate of the level of noise in the PPG [3]. Two systolic wave detection algorithms were used since both are quick to implement and measures the PPG signal from different perspectives. One is from the SciPy built-in package, and the other is Billauer's algorithm based on local maxima and minima [24] [22]. The matching of the algorithms is defined as follows:

$$MSQ = \frac{\#(S_{SciPy} \cap S_{Billauer})}{\#(S_{Billauer})} \tag{4.6}$$

where S_{SciPy} and $S_{Billauer}$ represents the lists of systolic waves detected by the SciPy and Billauer's algorithm [25]. $\#(A)$ represents the length of list A, which is the number of elements in list A. MSQ tracks the agreement between two peak detectors to evaluate

quality of the signal.

7. Perfusion

As Elgendi stated, perfusion is a gold standard for assessing PPG signal quality [3] [32] [33] [34]. Perfusion Index (PI) is the ratio of the pulsatile blood flow to the non-pulsatile static blood flow in a patient's peripheral tissue, such as finger tip, toe, or ear lobe [35]. It is an indication of the pulse strength at the sensor site: the PPG is a graphical representation of the perfusion index available in many pulse oximeters [36]. The calculation of perfusion index is

$$PI = \frac{y_{max}(x) - y_{min}(x)}{|\bar{x}|} \tag{4.7}$$

where $\bar{x}$ is the sample mean of the raw PPG signal x, and $y(x)$ is the filtered PPG signal.

4.2.2 Spectrogram

The other useful PPG feature extraction method is spectrogram. Spectrogram is a visual representation of the signal frequency spectrum as it changes with time [37]. A typical spectrogram of a speech signal waveform is shown in Fig. 4.2. Spectrogram is calculated by evaluating the short-term discrete Fourier transform [5]. Red areas in the figure show high intensity.

There are two reasons for choosing spectrogram as a feature representation of PPG signal. Firstly, spectrogram is a representation of how the frequency components within a signal vary with time. Since PPG signals of different qualities exhibit significantly different frequency components at different times, spectrogram is useful for time-frequency analysis. Secondly, spectrogram is a 2-dimensional representation of 1-dimensional PPG waveform. Therefore, representing PPG as spectrogram can facilitate input into convolutional neural network for classification.

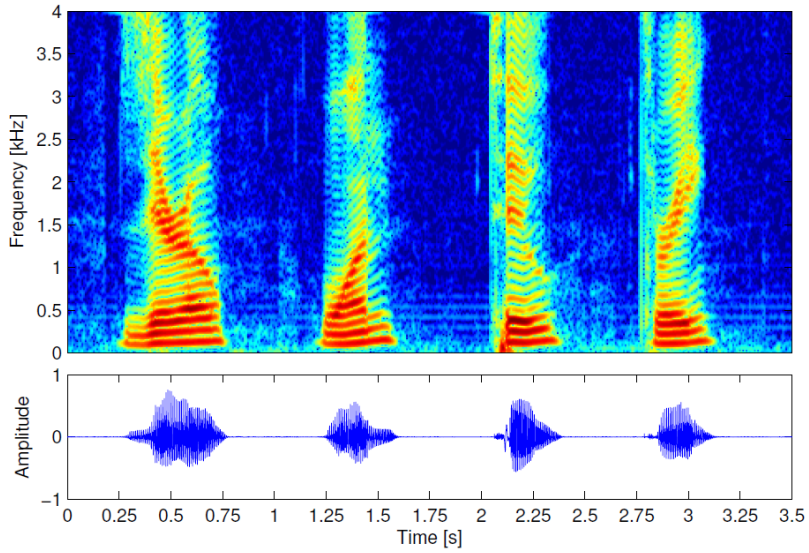


Figure 4.2: Spectrogram of a speech signal waveform [5]

4.3 Semi-Automatic Labelling

The goal of this project is the classification of PPG signal quality, i.e. the automatic labelling of PPG data windows. Semi-automatic labelling is defined by us as somewhere between the manual labelling by the observation and the automatic labelling by the classifiers. Since our goal is to eventually classify signals as a supervised learning task, we need to label the raw data and create a labelled dataset. To label the raw PPG waveforms, we employ the unsupervised clustering method.

4.3.1 Clustering

Clustering is the task of automatically finding similarities between data points and forming them into groups called clusters [38]. Here we cluster the windows of PPG based on similarity, and this similarity is measured using 7 SQIs computed in a 7-dimensional space. The number of clusters is set to 3, representing 3 qualities as discussed in Chapter 2. According to the possible distribution of the 3 clusters, we choose the k-means clustering algorithm.

4.3.2 K-Means

The k-means algorithm clusters data by dividing the samples into k groups of equal variance and minimising a criterion called inertia or within-cluster sum-of-squares [39]. The algorithm requires specifying the number of clusters. It scales well to large number of samples and has been used in a wide range of applications in many different fields [40].

The steps of the k-means algorithm are as follows [40]:

1. Randomly initialise k centroids (the centre of a cluster) in the workspace
2. Assign each data point to the nearest centroid to create k clusters of data points
3. Calculate the means of the data points in each cluster
4. Update the centroids to be the means calculated in step 3
5. Repeat from step 2 to 4 until values converge

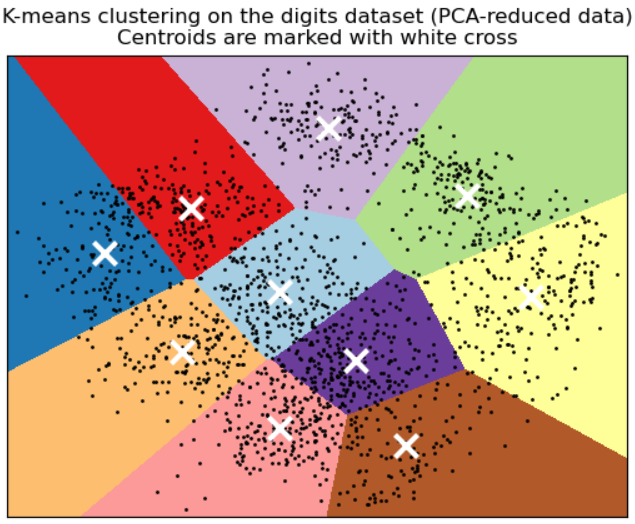


Figure 4.3: K-Means clustering on the handwritten digits data [6]

K-means is an iterative algorithm: on each pass it gets closer to the solution and stops when the solution converges or we ask it to stop. It tries to keep cluster means as

far apart as possible while keep variation within clusters as small as possible. A typical result of k-means clustering on the handwritten digits dataset is shown in Fig. 4.3 [6].

After applying k-means to the SQIs of each PPG window, we obtain the class of each sample. In practice, we use Mini Batch K-Means as a variant of the k-means algorithm which uses mini-batches to reduce the computation time, while still trying to optimise the same objective function [41]. A mini-batch is a subset of the input data which is randomly sampled in each training iteration. These mini-batches greatly reduce the amount of computation required to converge to a local solution [41].

4.4 Visualisation

Now we have built a labelled dataset through k-means clustering, we would like to see if the labels are meaningful and what similarities are there between the PPG samples. The windowed PPG is represented by 7 SQIs, which means that each window is mapped in a 7-dimensional space, and humans cannot perceive a space higher than 3 dimensions. Therefore, we use dimensionality reduction techniques to project the data point of each window in a 2-dimensional space.

4.4.1 Dimensionality Reduction

In this task, each PPG sample is represented by 7 SQIs, i.e. 7 features. It is unlikely that each feature is equally important for discrimination or classification. Classification accuracy is usually not proportional to dimension. So we can get similar (or even the same) classification results with fewer dimensions. Lower dimensionality also reduces computational demands and enables data visualisation. To visualise the data (and improve interpretability), we need to reduce data dimensions and preserve data structure [7].

In this step, both linear and nonlinear dimensionality reduction techniques: principal component analysis (PCA) and autoencoder are introduced.

4.4.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear dimensionality reduction algorithm using Singular Value Decomposition (SVD) of the data to project it to a lower dimensional linear space [42]. The input data is centered but not scaled for each feature before applying the SVD. PCA is defined as an orthogonal projection of the data onto a lower dimensional space such that the variance of the projected data is maximised [7]. The dimensions are determined by current data, making PCA adaptive to current conditions. Some high-dimensional data projected to 2-dimensional space is illustrated in Fig. 4.4.

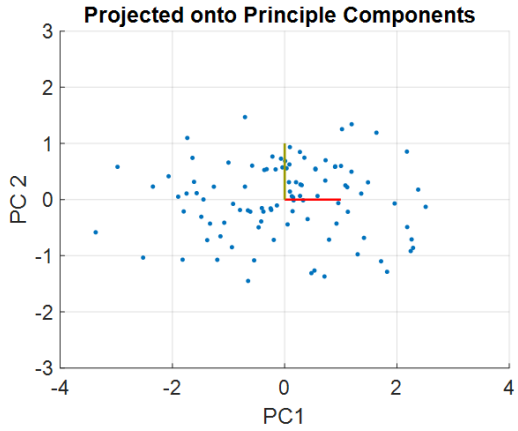


Figure 4.4: Projection onto 2-D by PCA [7]

The PCA steps include standardise the data, compute the covariance matrix, calculate eigenvectors and eigenvalues, select a feature vector, and project data onto principle components [7]. In very high-dimensional spaces, Euclidean distances tend to become inflated (this is an example of the so-called "curse of dimensionality"). "Running a dimensionality reduction algorithm such as principal component analysis (PCA) before k-means clustering can alleviate this problem and speed up the computations." [39].

4.4.3 Autoencoder

Autoencoder is an unsupervised learning technique and a special type of artificial neural network (ANN) used to learn efficient coding of unlabelled data [8]. The encoding is validated and refined by trying to regenerate the input from the encoding. The autoencoder

learns a latent space representation (encoding) for the input data by training the network to ignore insignificant data such as noise [8]. Therefore, autoencoder is ideal for dimensionality reduction. An autoencoder consists of three parts: encoder (compression), latent space (coding), and decoder (reconstruction), refer to Fig. 4.5.

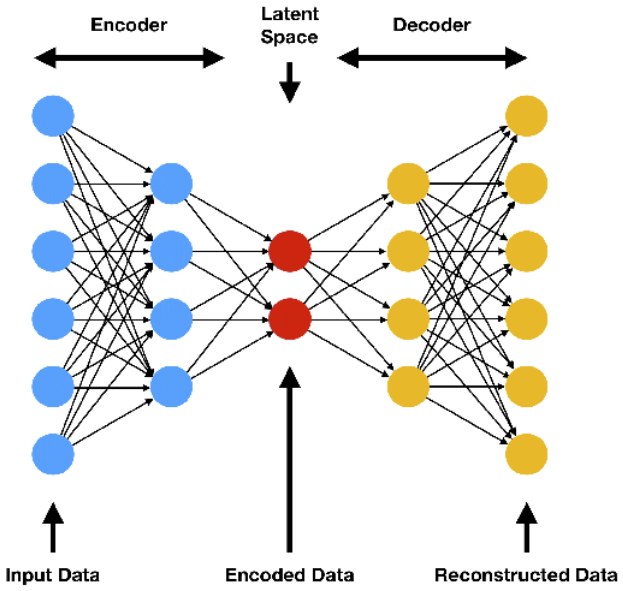


Figure 4.5: The structure of autoencoder [8]

Given a dataset of PPG features like SQIs, the autoencoder first encodes a 7-dimensional sample into a 2-dimensional latent representation, then decodes the latent representation back to the 7-dimensional sample. The loss function is mean squared error (MSE) between the input data and output reconstruction. The autoencoder learns to compress the data while minimising the reconstruction error.

4.5 Automatic Labelling

The ultimate goal of this project is to achieve automatic labelling of PPG quality, when a dataset with both the input features (SQIs, raw data, spectrograms) and quality outcomes (classes) are ready. Since the dataset has been labelled, automatic labelling is a kind of supervised learning aims to determine a function that approximates the given relationships between the PPG data and quality.

The classification algorithms we use for comparison are: decision tree, random forest, AdaBoost and convolutional neural networks (CNN). The input sample of the first three algorithms is the SQI and the raw signal, and the input sample of CNN is 2-dimensional spectrogram images. The algorithms are briefly introduced as follows.

4.5.1 Decision Tree

Decision tree is a non-parametric supervised learning method for classification and regression. The goal of a decision tree is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features [11]. A tree can be viewed as a piecewise constant approximation [9].

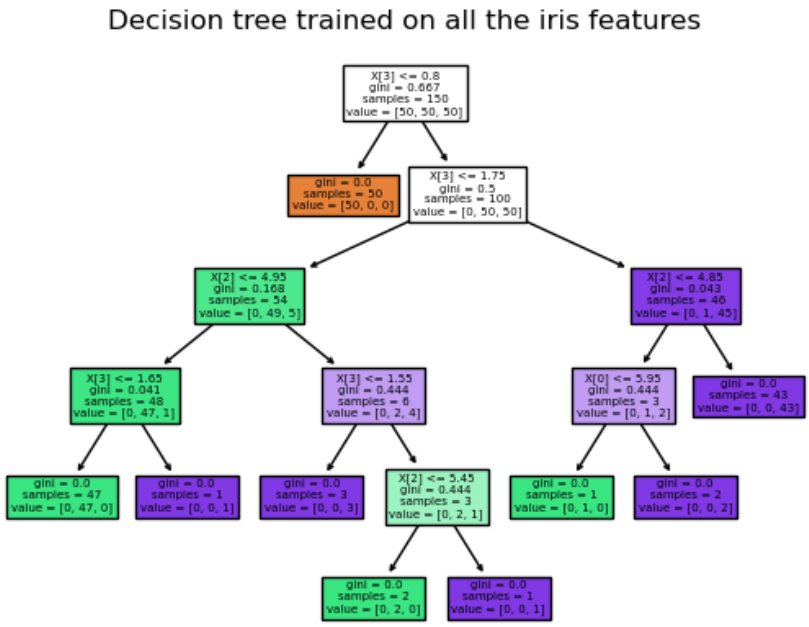


Figure 4.6: A decision tree trained on the iris dataset [9]

A typical decision tree plot is illustrated in Fig. 4.6. In the figure, each node is a test on the data, each branch is the outcome of the test, each leaf node is a final decision [11]. The root node is a decision that has highest influence in splitting the data. The variables and outcomes can be repeated.

4.5.2 Random Forest

Random forest is an *ensemble learning* method for classification that operates by building multiple decision trees at training time [10]. Ensemble learning combines the predictions of several base estimators built with a given learning algorithm in order to improve generalisability/robustness over a single weak estimator [43]. Ensemble learning consists of two main methods: averaging methods (e.g. random forest) and boosting methods (e.g. AdaBoost) [43].

Each decision tree in a random forest is built based on different parts of the training data. The decision trees together vote for an outcome [11]. Training of each node is performed with only some of the random selected variables, and is called Feature Bagging or Random Subspace Projection [11]. This increases variation in the Trees, reduces overfitting and increases accuracy. See Fig. 4.7 for a diagram of random forest.

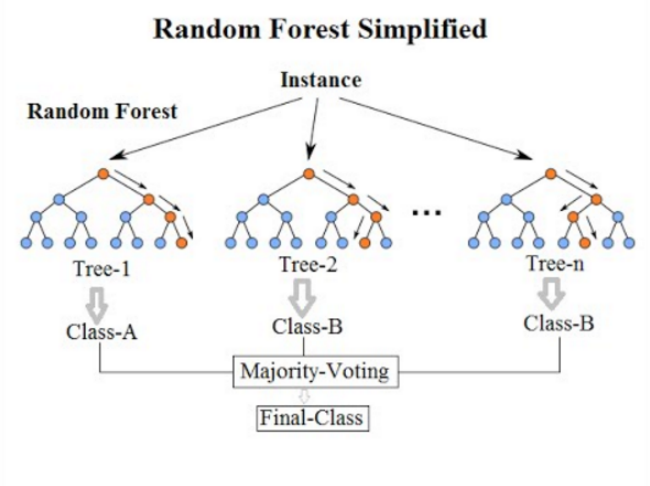


Figure 4.7: A diagram of random forest [10]

4.5.3 AdaBoost

AdaBoost, short for Adaptive Boosting, is a boosting method of ensemble learning. It generates decision trees (weak learners) sequentially and each new weak learner attempts to improve on previous classification result, see Fig. 4.8.

The decision trees used in AdaBoost are stumps generated sequentially rather than all

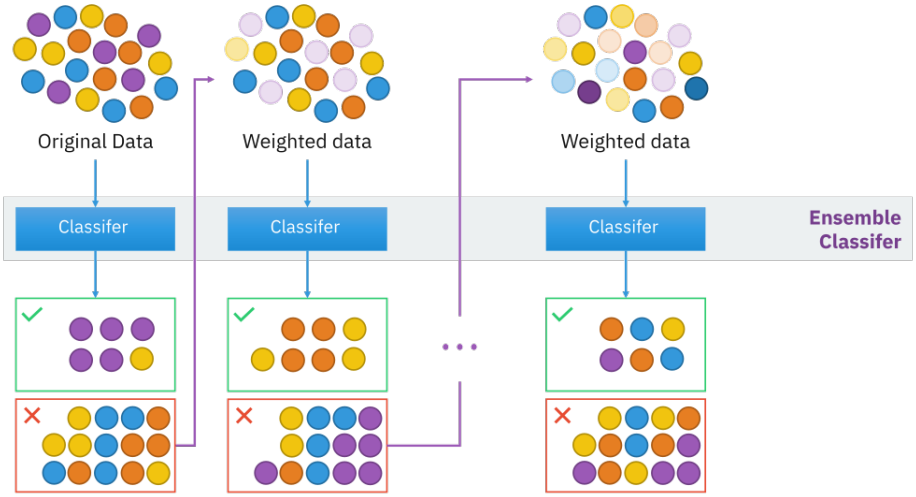


Figure 4.8: A diagram of boosting [11]

at once. A stump is a decision tree with just a root node and two leaves. Successive stumps are influenced by previous stumps. The different stumps are generated by weighting the training data to focus on prior errors. The weights are updated on each training iteration. Additionally, the output of each decision stump is weighted. So, some decision stumps have more influence over the output than others [11].

4.5.4 Convolutional Neural Networks (CNN)

The convolutional neural network (CNN) is a neural network in which at least one layer is a convolutional layer [44]. The neural network is a model that, taking inspiration from the brain, is composed of layers (at least one of which is hidden) consisting of simple connected units or neurons (nodes) followed by non-linearities [44]. The non-linearity comes from the activation function (ReLU, sigmoid, etc.) that takes in the weighted sum of all of the inputs from the previous layer and then generates and passes an output value (typically nonlinear) to the next layer [44]. Convolutional neural networks have had great success in many applications, such as image recognition, image classification, and image segmentation. A typical CNN architecture is shown in Fig. 4.9.

Convolutional neural networks usually consist of some combination of the following layers:

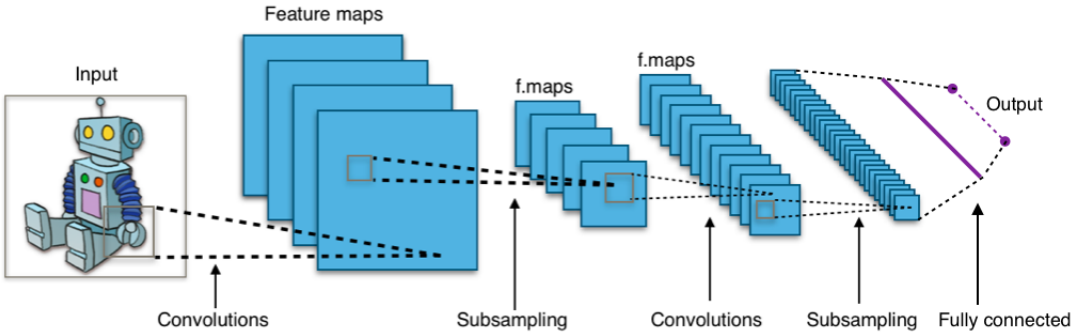


Figure 4.9: Typical Convolutional Neural Networks [12]

1. Convolutional layers. The convolutional layer is a layer in which a convolutional filter passes along an input matrix. A convolutional filter is a matrix having the same rank as the input matrix, but a smaller shape. A convolutional layer consists of a series of convolutional operations, each acting on a different slice of the input matrix. The convolutional operation has two steps: (1) element-wise multiplication of the convolutional filter and a slice of an input matrix; (2) summation of all the values in the resulting product matrix [44].

2. Pooling layers. The pooling layer reduces a matrix (or matrices) created by an earlier convolutional layer to a smaller matrix. Pooling usually involves taking either the maximum or average value across the pooled area and helps enforce translational invariance in the input matrix. Pooling for vision applications is known more formally as spatial pooling. Less formally, pooling is often called sub-sampling or down-sampling [44].

3. Fully connected layers. The fully connected layer is a hidden layer in which each node is connected to *every* node in the subsequent hidden layer. A fully connected layer is also known as a dense layer [44].

4.6 Summary

Various designs, algorithms, methods, and principles for PPG signal preparation, analysis, visualisation, and classification have been described above. The next chapter will introduce

the specific implementation, experiments, results and analysis based on this chapter.

Chapter 5

Experiment and Result

In this chapter, we present the implementation and experiments of the PPG signal analysis and classification, and the results are summarised and discussed. Specifically, this chapter is organised in the order of procedures as follows: data preprocessing, feature extraction, semi-automatic labelling, visualisation, and automatic labelling. Please refer to the flowchart in Fig. 4.1 for the design.

All data and code for this project are run on a computer with an AMD R7-4800U 8-core 1.8GHz (max 4.2GHz) CPU and a 16GB DDR4-3200MHz memory.

5.1 Data Preprocessing

5.1.1 Dataset

All the data used in this project is from the **01NVa** dataset collected from various prospective clinical studies which have been conducted by the Oxford University Clinical Research Unit (OUCRU) [15]. These studies have been conducted in healthcare facilities within the Hospital for Tropical Diseases (HTD) in Ho Chi Minh City, Vietnam. For both the children and adult data from 01NVa, we use the adult data and select the PLETH column of each patient's waveform sequence over time. PLETH is short for Photoplethysmogram (PPG).

The raw PPG data and statistics of different adult patients are shown in the Table. 5.1

Patient number	PPG length	Time (h)	Mean	Std	Min	Max
01NVa-003-2001	5626377	15.628	29698.443	17799.433	0	65535
01NVa-003-2012	5176634	14.379	17115.179	20088.398	0	65535
01NVa-003-2023	1645899	4.571	13441.664	18889.555	0	65535
01NVa-003-2023	2122127	5.894	25176.773	19579.210	0	65535
01NVa-003-2028	878261	2.439	32693.617	15787.726	0	65535
01NVa-003-2103	5297036	14.713	11461.761	17913.632	0	65535
01NVa-003-2104	5027951	13.966	28195.070	17829.383	0	65535
01NVa-003-2109	3896567	10.823	31364.225	16738.005	0	65535
01NVa-003-2110	4404269	12.234	26306.603	18766.272	0	65535
01NVa-003-2110	189207	0.525	20162.520	19859.245	0	65535
01NVa-003-2162	5464398	15.178	28381.130	18558.764	0	65535

Table 5.1: 01NVa PPG dataset and statistics of some adult patients

(some patients have multiple PPG recordings). The raw PPG waveform is sampled 100 times per second (sampling rate $f_s = 100$), the minimum value is 0, and the maximum value is 65535. The reason for the apparent difference in means and standard deviations is that the length of the zero signal varies in different patient data.

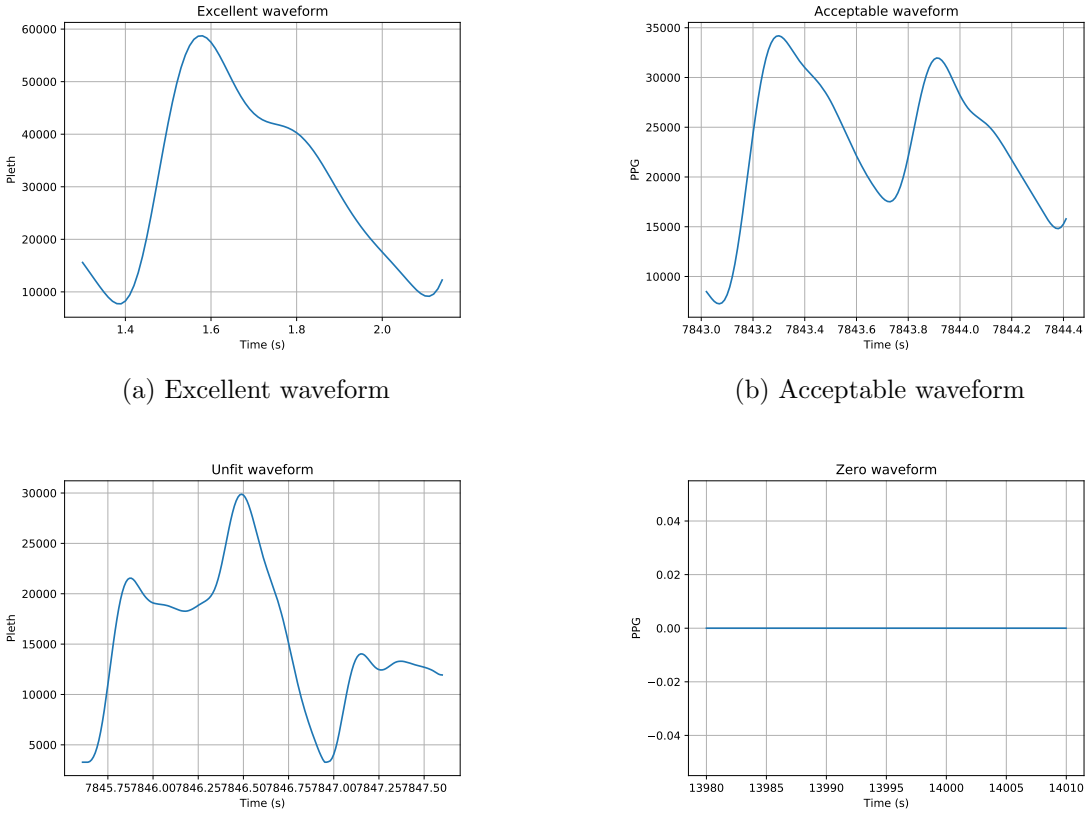
5.1.2 Data Preparation

The quality of the raw PPG signal can be roughly divided into four categories, as displayed in Fig. 5.1: excellent (Fig. 5.1a), acceptable (Fig. 5.1b), unfit (Fig. 5.1c), and zero (Fig. 5.1d).

To simplify the classification, excellent and acceptable waveforms are collectively referred to as *good* waveforms, since we can detect the patient's pulse within these two. Then the three waveforms are categorised as: **good**, **unfit**, and **zero**.

In order to improve the generalisation of the model while avoiding the increase of computational time due to the large amount of data, we use the data of 4 patients (01NVa-003-2001, 2012, 2103, and 2104) and concatenate them together after preprocessing.

The PPG data for each patient are trimmed 5 minutes from head and tail, and the remaining data are min-max normalised to $[0, 1]$ (see Chapter 4.1). In order to make the signal quality of each window as uniform as possible, the normalised data are truncated into 10-second windows. Each window has 1000 magnitude numbers.



(a) Excellent waveform

(b) Acceptable waveform

(c) Unfit waveform

(d) Zero waveform

Figure 5.1: Classes of PPG waveform

After calculation, the number of data samples (windows) of the 4 patients are 5567, 5117, 5238, and 4968. The total number of samples is therefore 5567+5117+5238+4968 = 20890 windows.

5.2 Feature Extraction

5.2.1 Signal Quality Index (SQI)

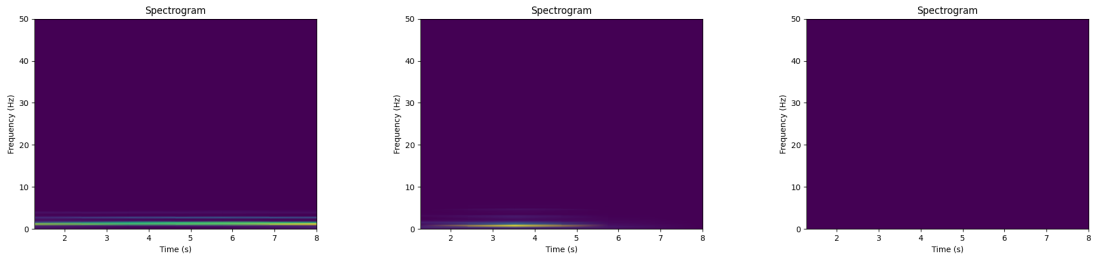
The calculation methods of SQIs are shown in Chapter 4.2.1. Note that we need to replace the *inf* entries in perfusion index with a specified number 100, since perfusion of zero data is infinity. We choose 100 because this is a value that exceeds the normal perfusion range, and will not affect the accuracy of subsequent clustering if it is too large.

The resulting SQI feature is a 20890×7 matrix.

5.2.2 Spectrogram

In this section of spectrogram generation, we use a Tukey window with 1/8 of a window's length overlap at each end, and the size of an output spectrogram is 129×4.

Three spectrogram examples of good, unfit and zero waveforms are shown in Fig. 5.2. As can be seen from the figures, the high frequency components are all 0 and dark. Since most of the frequency entries of the spectrograms are 0 (very sparse) from 21 to 129, the spectrograms are all cropped to 20×4 size.



(a) Good spectrogram (b) Unfit spectrogram (c) Zero spectrogram

Figure 5.2: Classes of PPG spectrogram

The resulting spectrogram feature is a $20890 \times 20 \times 4$ n-dimensional array.

5.3 Semi-Automatic Labelling

The aim of semi-automatic labelling is to generate quality labels $(0, 1, 2)$ for the windowed PPG dataset. We cluster the windows based on similarity measured using 7 SQIs. We employ the Mini-Batch K-Means clustering algorithm which is much faster than vanilla k-means [41] [45]. The settings of maximum number of iterations over the complete dataset is 400, the batch size is 2048, the number of random initialisation is 3, and the random state is 42.

The number of each quality label after applying the mini-batch k-means clustering is listed in Table. 5.2. The clustering results are stable and unchanged in multiple trials.

These labels will serve as the ground truth of windowed PPG dataset for subsequent visualisation and classification.

Label	2	1	0
Quality	Good	Zero	Unfit
Number	9906	7076	3908

Table 5.2: Result of semi-automatic labelling

5.4 Visualisation

The 7-dimensional SQI features of each PPG sample are projected to 2 dimensions, implemented by PCA and autoencoders. By using clustering in the 2-dimensional space, we can intuitively see how the PPG samples are distributed.

5.4.1 Principal Component Analysis (PCA)

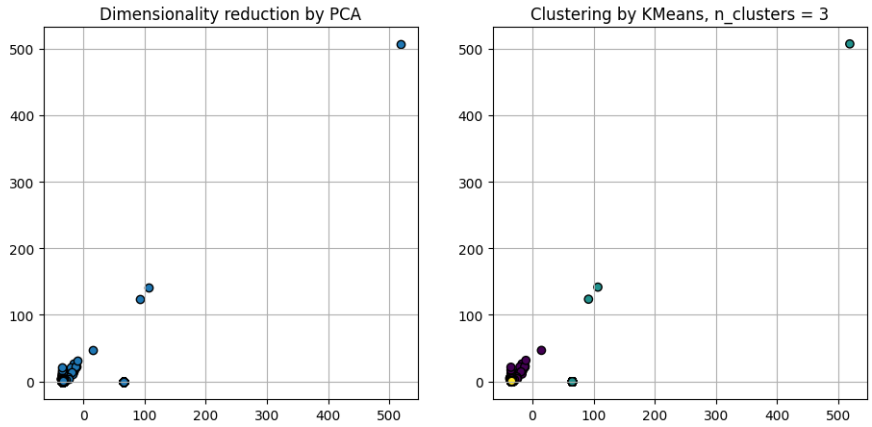


Figure 5.3: SQI features dimensionality reduced by PCA and clustered by mini-batch k-means

The 7-dimensional SQI features are first projected to 2-dimensional space through PCA, and then clustered into 3 clusters by mini-batch k-means. See the visualisation in 2-D in Fig. 5.3. The good PPG samples are the yellow points at about (-33, 0). Zero samples are the blue points at (65.327, -0.461). Unfit samples are the purple points near

the yellow points. The labelling result is: (Good: 10936, Zero: 7076, Unfit: 2878). We can see the clustering of zero samples is accurate, while good and unfit are somewhat different from the ground truth. The results of PCA dimensionality reduction method are stable and remain unchanged in different trials.

5.4.2 Autoencoder

The model and hyperparameter settings of the autoencoder is shown in Table. 5.3. The first row of the table is abbreviated: output shape of encoder dense layers (Encoder), activation function (Activation), learning rate (LR), epoch (Epoch), batch size (Batch), optimiser (Optimiser), loss function (Loss). The encoder and decoder are symmetric in layer outputs to generate latent space representations in the middle. The validation split is 10% of the total dataset.

Encoder	Activation	LR	Epoch	Batch	Optimiser	Loss
[20, 14, 8, 5, 2]	ReLU	1e-3	20	128	Adam	MSE

Table 5.3: Autoencoder settings

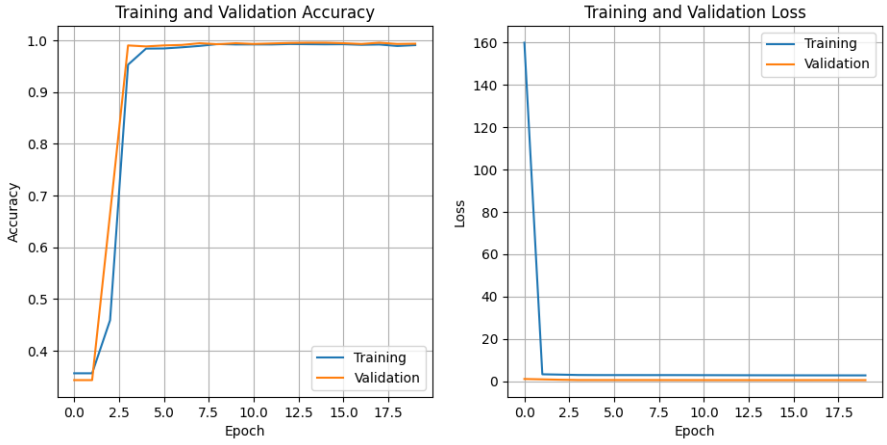


Figure 5.4: Training and validation accuracy and loss curves of the autoencoder

The architecture of the autoencoder model is shown in Fig. A.1 in Appendix A. The 7-dimensional SQI features are first projected to 2-dimensional space by the autoencoder, and then clustered into 3 clusters by mini-batch k-means. The training and validation

accuracy and loss curves of the autoencoder is in Fig. 5.4. It can be seen from the figure that the training results converge quickly, but in most trials, the autoencoder does not converge well. Autoencoder is very hard to tune, train, and converge.

The 2-D visualisation is in Fig. 5.5. The good PPG samples are the purple dots at about (2, 0). Zero samples are the blue dots at (105.125, 6.918). Unfit samples are the yellow dots next to the purple dots. The labelling result of mini-batch k-means is: (Good: 7132, Zero: 7077, Unfit: 6681). The labelling of zero samples is accurate, while the prediction results of good and unfit deviate more from the ground truth than PCA. The clustering result is always changing and unstable.

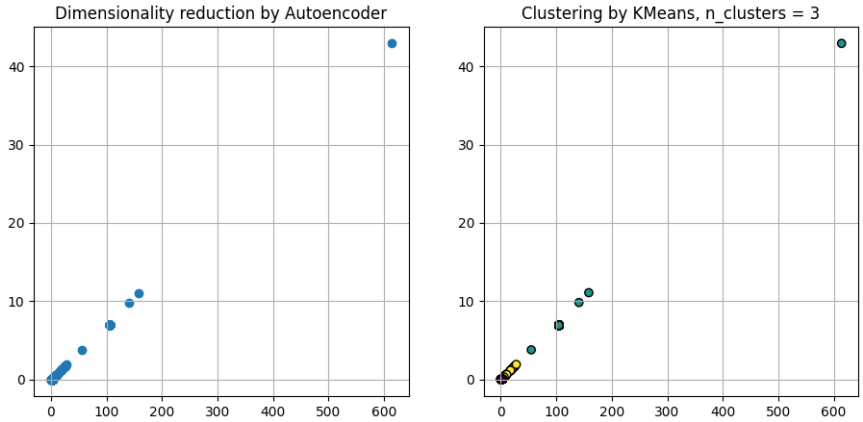


Figure 5.5: SQI features dimensionality reduced by autoencoder and clustered by mini-batch k-means

5.4.3 Discussion

The 2-D visualisations for both PCA and autoencoder are always close to a straight line, which means that one SQI may play a leading role among the seven SQIs in the classification of signal quality.

In the clustering results, the zero signal samples are the most obvious ones: all clustered together. Good samples are also concentrated, but unfit samples are relatively scattered next to the good ones. A few outliers far from the good samples should be clustered into unfit and actually clustered into the zero class. The clustering of zero is the

most accurate, while good and unfit are quite different from the ground truth. The reason is that in the previous step of semi-automatic labelling, outliers were incorrectly labelled as zero samples.

5.5 Automatic Labelling

Automatic labelling is a classification problem of supervised learning, trying to find the correspondence between unknown PPG samples and quality labels. We will experiment with each of the four classifiers: decision tree, random forest, AdaBoost and convolutional neural networks (CNN).

5.5.1 Decision tree, Random forest, and AdaBoost

In order to avoid the particularity of single test results, 10 trials were performed on the SQI and raw PPG data. The decision tree [9], random forest and AdaBoost classification algorithms were tested respectively. For the random forest, the number of trees in the forest is 100, and the bootstrap method is used when building trees [43]. For the AdaBoost, the maximum number of estimators at which boosting is terminated is 50, and the weight applied to each classifier at each boosting iteration is 1.0 [46]. In each trial, 25% of the data were randomly sampled from the dataset as the test set. The average accuracy on 10 trials of the three classifiers on the test set is shown in Table. 5.4.

Input data	Decision tree	Random forest	AdaBoost
Raw PPG signal	0.8422	0.9151	0.7436
SQI	1.0	1.0	1.0

Table 5.4: Accuracy of 3 classifiers on raw PPG signal and SQI

All three classifiers performed perfectly on the SQI data with 100% accuracy, since the SQI itself is the source of the labelled clustering of the windowed PPG data. A decision tree on the SQI data is shown in Fig. 5.6

Random forest is composed of 100 decision trees, so its accuracy rate is higher than decision tree. AdaBoost has the lowest accuracy; therefore, is not suitable for this task.

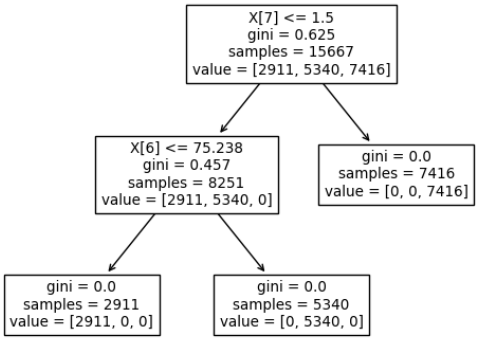


Figure 5.6: Decision tree performed on SQI features

The raw data has not been feature extracted and/or dimensionality reduced, so the amount of data is too large, and the calculation time is very long, more than 15 minutes.

5.5.2 Convolutional Neural Networks (CNN)

The CNN is designed to be a sequential model with 3 convolutional layers, 1 max pooling layer, 1 flatten layer, and 2 dense layers. The 3×3 convolutional filters are applied on input of 20×4 spectrograms. The model architecture is shown in Fig. A.2 in Appendix A. The activation function of all the layers are ReLU, except for the last dense layer to be softmax activated.

The number of epochs is 100, and batch size is 32 or 256 in two trials. Other hyper-parameter settings are in Table. 5.5. The test split is 0.3, and validation split is 0.15. The accuracy and loss for training and validation is shown in Fig. 5.7 for batch size = 32, and shown in Fig. 5.8 for batch size = 256.

Activation	LR	Epoch	Batch	Optimiser	Loss
ReLU & Softmax	1e-3	100	32, 256	Adam	Cross Entropy

Table 5.5: CNN settings

Larger batch size can speed up training: the training time is 2s 6ms/step for 32,

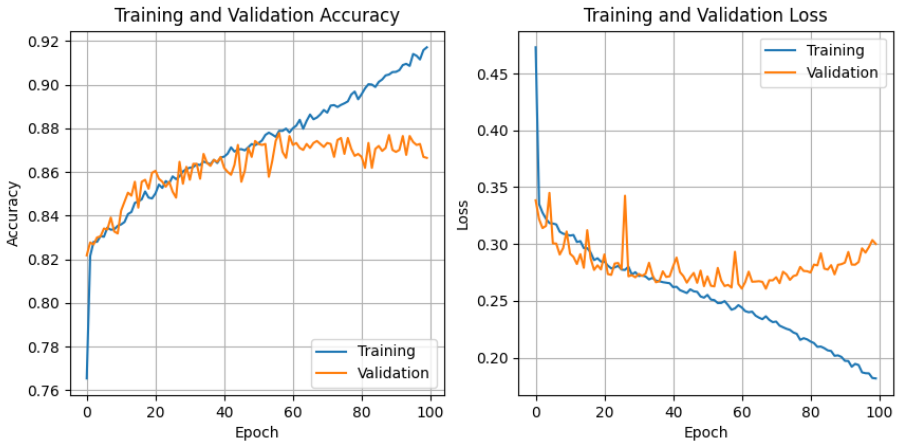


Figure 5.7: CNN training and validation accuracy and loss (batch size = 32)

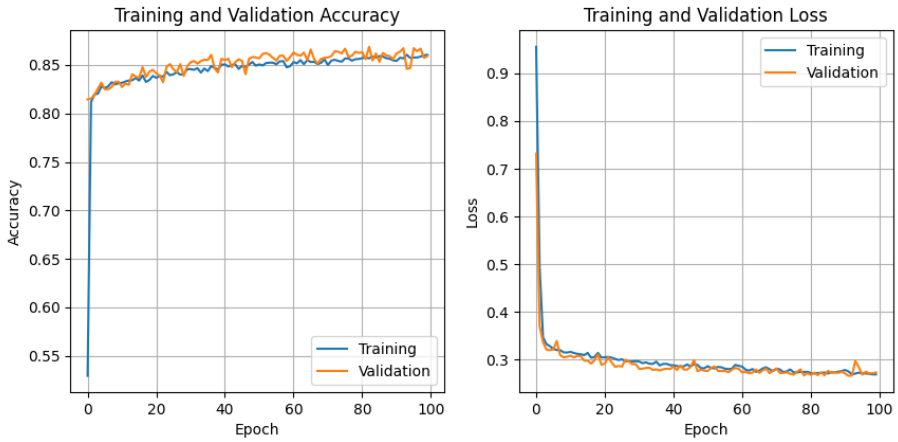


Figure 5.8: CNN training and validation accuracy and loss (batch size = 256)

and 1s 21ms/step for 256. The smaller the batch size, the greater the difference between training and validation accuracy and loss, the greater the fluctuation, and the model tend to overfit the training set.

The test accuracy and loss is listed in Table. 5.6. As the result shows, the smaller the batch size, the higher the accuracy.

5.5.3 Discussion

The accuracy of CNN classifier is slightly higher than that of decision trees and lower than that of random forest. The time taken by CNN to classify spectrograms is much lower

Batch size	Accuracy	Loss
32	0.86708	0.30515
256	0.85352	0.28692

Table 5.6: Test accuracy and loss of CNN on PPG spectrogram

than that of random forest to classify raw data.

Classification using SQIs is 100% accurate in this experiment but the SQIs must be selected and defined properly in advance for classification input. If the SQIs used for semi-automatic labelling is different from the SQIs used for classification, it is likely to reduce the accuracy of automatic labelling. CNN is the recommended classifier because it has high accuracy, does not depend on SQIs, and the computation of the spectrogram is neither difficult nor time-consuming.

5.6 Summary

This chapter contains all experiments, results and discussions. The next chapter will cover conclusions and possible future work.

Chapter 6

Conclusion and Future Work

6.1 Conclusions

1. Carefully chosen SQIs are effective PPG features that facilitate quality labelling and dimensionality reduction.

2. The spectrogram is also an effective PPG feature, and the CNN classifier has good stability, good flexibility, high accuracy and fast speed to be an ideal PPG signal quality classification method.

3. The classification accuracy of random forest is higher than that of decision tree and AdaBoost, but it is slow to calculate on raw data without feature extractions.

4. For raw PPG waveform without labels, it is ideal to label with SQIs. For labelled PPG datasets, computing the spectrogram and using CNN for classification is preferred.

6.2 Future Work

1. Continue the search on finding the most effective SQIs/features for measuring PPG quality. Try to get rid of some SQIs that do not measure quality well.

2. Explore more advanced time-series clustering algorithms.

3. Try to represent the PPG signal quality in continuous scores, and the classification problem becomes a regression problem.

4. Real-time PPG quality analysis methods on wearable devices can be studied.

Bibliography

[1] M. Elgendi, "On the analysis of fingertip photoplethysmogram signals," *Current cardiology reviews*, vol. 8, no. 1, pp. 14–25, 2012.

[2] D. Castaneda, A. Esparza, M. Ghamari, C. Soltanpur, and H. Nazeran, "A review on wearable photoplethysmography sensors and their potential future applications in health care," *International journal of biosensors bioelectronics*, vol. 4, no. 4, p. 195, 2018, pmid:30906922.

[3] M. Elgendi, "Optimal signal quality index for photoplethysmogram signals," *Bioengineering*, vol. 3, no. 4, p. 21, 2016.

[4] C. Orphanidou, T. Bonnici, P. Charlton, D. Clifton, D. Vallance, and L. Tarassenko, "Signal-quality indices for the electrocardiogram and photoplethysmogram: Derivation and applications to wireless monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 832–838, 2015, iD: 1.

[5] P. A. Naylor, "Speech processing time-frequency analysis," 2021.

[6] "A demo of k-means clustering on the handwritten digits data." [Online]. Available: https://scikit-learn/stable/auto_examples/cluster/plot_kmeans_digits.html

[7] A. Spiers, "Pattern recognition - dimensionality reduction," 2021.

[8] "Autoencoder," -08-24T03:32:59Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Autoencoder&oldid=1106330415

[9] "sklearn.tree.decisiontreeclassifier." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[10] "Random forest," -08-27T00:53:11Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Random_forest&oldid=1106889998

[11] A. Spiers, "Pattern recognition - classification," 2021.

[12] "Convolutional neural network," -08-30T13:11:06Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1107527575

[13] S. Bhatt, P. W. Gething, O. J. Brady, J. P. Messina, A. W. Farlow, C. L. Moyes, J. M. Drake, J. S. Brownstein, A. G. Hoen, and O. Sankoh, "The global distribution and burden of dengue," *Nature*, vol. 496, no. 7446, pp. 504–507, 2013.

[14] W. H. Organization, "Dengue and severe dengue." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue

[15] "The oucru datasets." [Online]. Available: https://bahp.github.io/vital-oucru-clinical/datasets/overview.html#the-01nva-dataset

[16] K. Nakajima, T. Tamura, and H. Miike, "Monitoring of heart and respiratory rates by photoplethysmography using a digital filtering technique," *Medical engineering physics*, vol. 18, no. 5, pp. 365–372, 1996.

[17] A. B. Hertzmann, "Observations on the finger volume pulse recorded photoelectrically," *Am J Physiol*, vol. 119, pp. 334–335, 1937.

[18] A. D. Foster, C. Neumann, and E. A. Rovenstine, "Peripheral circulation during anesthesia, shock and hemorrhage; the digital plethysmograph as a clinical guide," in *The Journal of the American Society of Anesthesiologists*, vol. 6. The American Society of Anesthesiologists, 1945, pp. 246–257.

[19] D. T. Ubbink, "Toe blood pressure measurements in patients suspected of leg ischaemia: a new laser doppler device compared with photoplethysmography," *European journal of vascular and endovascular surgery*, vol. 27, no. 6, pp. 629–634, 2004.

[20] C. Butter, C. Stellbrink, A. Belalcazar, D. Villalta, M. Schlegl, A. Sinha, F. Cuesta, and C. Reister, "Cardiac resynchronization therapy optimization by finger plethysmography," *Heart Rhythm*, vol. 1, no. 5, pp. 568–575, 2004.

[21] Q. Li and G. D. Clifford, "Dynamic time warping and machine learning for signal quality assessment of pulsatile signals," *Physiological Measurement*, vol. 33, no. 9, p. 1491, 2012.

[22] N. Pradhan, S. Rajan, and A. Adler, "Evaluation of the signal quality of wrist-based photoplethysmography," *Physiological Measurement*, vol. 40, no. 6, p. 065008, 2019.

[23] "sklearn.preprocessing.minmaxscaler." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[24] O. Muttawa, J. R. Manzano, and P. Georgiou, "A real-time independent and inexpensive ppg signal quality classification tool for vital sign monitoring," pp. 1–71, 2020.

[25] "Sqi calculations." [Online]. Available: https://meta00.github.io/vital_sqi/_examples/others/plot_pipeline_02.html

[26] R. Krishnan, B. Natarajan, and S. Warren, "Two-stage approach for detection and reduction of motion artifacts in photoplethysmographic data," *IEEE transactions on biomedical engineering*, vol. 57, no. 8, pp. 1867–1876, 2010.

[27] "Skewness," -07-07T08:20:57Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Skewness&oldid=1096888694

[28] N. Selvaraj, Y. Mendelson, K. H. Shelley, D. G. Silverman, and K. H. Chon, "Statistical approach for the detection of motion/noise artifacts in photoplethysmogram," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 4972–4975.

[29] "Kurtosis," -06-21T21:35:39Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Kurtosis&oldid=1094312470

[30] "Signal-to-noise ratio," -08-17T04:26:57Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Signal-to-noise_ratio&oldid=1104835005

[31] M. Elgendi, "Detection of c, d, and e waves in the acceleration photoplethysmogram," *Computer methods and programs in biomedicine*, vol. 117, no. 2, pp. 125–136, 2014.

[32] M. H. Gehring, H. M. ME, and P. Schmucker, "The effects of motion artifact and low perfusion on the performance of a new generation of pulse oximeters in volunteers undergoing hypoxemia," *Respiratory care*, vol. 47, no. 1, pp. 48–60, 2002.

[33] M. Cannesson, B. Delannoy, A. Morand, P. Rosamel, Y. Attof, O. Bastien, and J.-J. Lehot, "Does the pleth variability index indicate the respiratory-induced variation in the plethysmogram and arterial pressure waveforms?" *Anesthesia Analgesia*, vol. 106, no. 4, pp. 1189–1194, 2008.

[34] C. Shan, "No title," *Motion robust vital signal monitoring*, 2015.

[35] D. A. Colquhoun, K. T. Forkin, M. E. Durieux, and R. H. Thiele, "Ability of the masimo pulse co-oximeter to detect changes in hemoglobin," *Journal of clinical monitoring and computing*, vol. 26, no. 2, pp. 69–73, 2012.

[36] "What is perfusion index (pi)." [Online]. Available: https://www.amperordirect.com/pc/help-pulse-oximeter/z-what-is-pi.html

[37] "Spectrogram," -08-29T21:06:13Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=1107405537

[38] "Cluster analysis," -08-17T04:48:39Z 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=1104837324

[39] "Clustering." [Online]. Available: https://scikit-learn/stable/modules/clustering.html

[40] A. Spiers, "Pattern recognition - clustering," 2021.

[41] "sklearn.cluster.minibatchkmeans." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html

[42] "sklearn.decomposition.pca." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.decomposition.PCA.html

[43] "sklearn.ensemble.randomforestclassifier." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[44] "Machine learning glossary." [Online]. Available: https://developers.google.com/machine-learning/glossary

[45] "sklearn.cluster.kmeans." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html

[46] "sklearn.ensemble.adaboostclassifier." [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

Appendix A

Neural Network Models

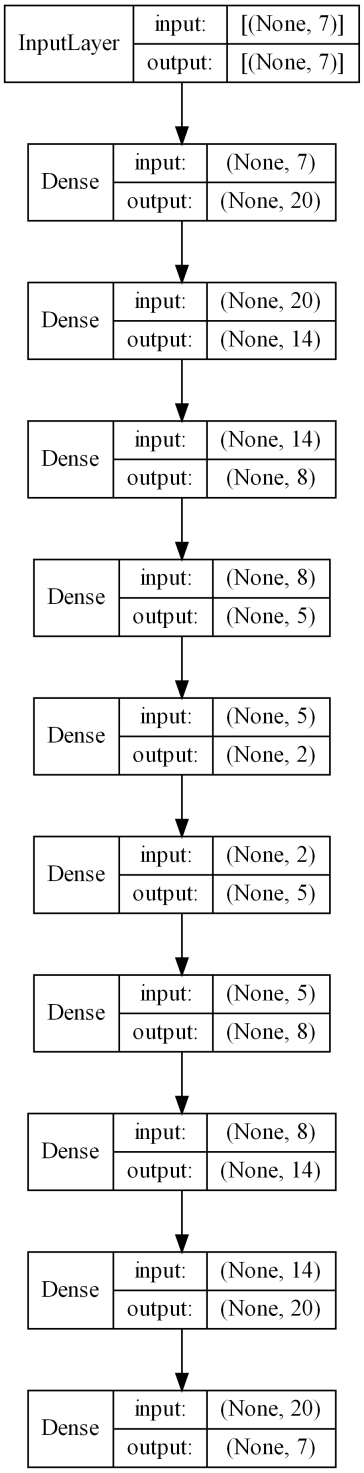


Figure A.1: Autoencoder model

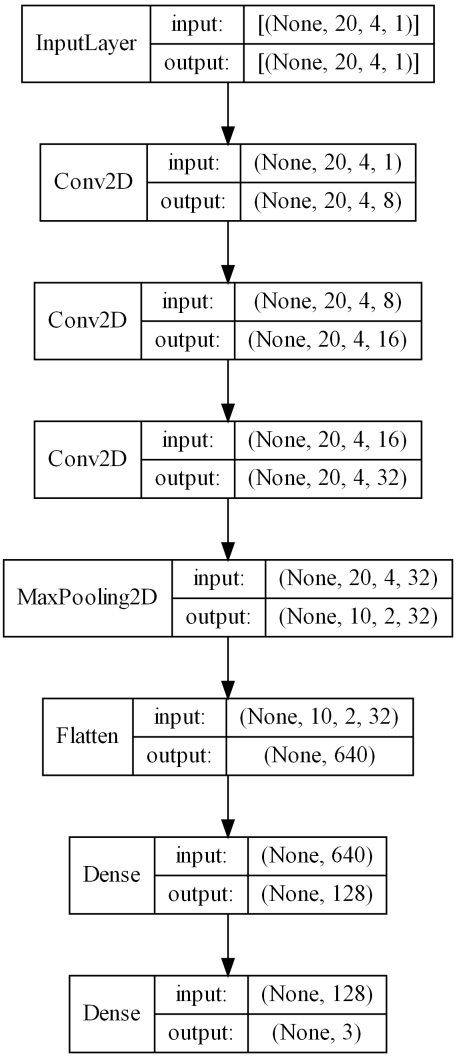


Figure A.2: CNN model