Imperial College London

Department of Electrical and Electronic Engineering

Final Year Project - Interim Report

Final Year Electrical and Electronic Engineering Project

ELEC97032 - 2021-2022

Author Name: Manginas Vasileios
Author Email: vm3218@ic.ac.uk
Author CID: 01542774

Contents

Chapter 1

Introduction

1.1 Motivation

Dengue is a a viral disease found primarily in countries with tropical and subtropical climates, and is transmitted through the bites of infected mosquitoes [1]. Although dengue outbreaks occur mostly in Latin American and Asian countries, around 4 billion people, more than half of the world's population, are at risk of viral infection, with an estimated number of 390 million cases annually [2]. While the majority of cases exhibit mild or even no symptoms, leading to an overall mortality rate of less than 1%, dengue patients occasionally progress to a more serious stage of the disease, severe dengue. One potentially lethal manifestation of severe dengue is Dengue Shock Syndrome (DSS). This dangerous complication, which will be referred to as "shock" in the remainder of this report, is associated with fatality rates ranging from $1 - 10\%$, and approaches 30% when untreated [3]. While there is no direct treatment for severe dengue or DSS, detecting deterioration early on lowers the fatality rate dramatically to 1% [1]. Despite this, in the areas affected by the disease, dengue epidemics can place extensive pressure on the healthcare system, especially during rainy seasons when transmission rates increase. This makes early detection of the disease much more challenging. Coupled with the fact that many of these areas include primarily lower-income countries, it is obvious that assistance in the form of patient management and prevention is both decisive and urgent.

Photoplethysmography (PPG) is an optical technique used to detect volumetric changes in blood in peripheral circulation, achieved by measuring and recording changes in the absorption of light through the tissue on measuring sites [4]. This process results in the PPG signal, a signal rich in information. This allows for estimation of vital sign parameters, such as a patient's heart rate, and is therefore suitable as a monitoring device. PPG sensors utilise simpler hardware than other alternatives, are non-invasive, cheap to manufacture and purchase, and thus promptly available [5]. Given the above, PPG sensors present themselves as an ideal candidate for patient monitoring in the case of massive-scale diseases, such as dengue, especially in lower-income countries.

In this light, it is undeniable that utilising PPG data with the aim of aiding the management of dengue outbreaks is a goal truly worth pursuing. It is this large potential for positive impact that serves as the main motivational force behind the entirety of this project.

1.2 Project Objectives

In broad terms, this project aims to explore clinical questions for dengue patients using PPG data, signal processing, and machine learning techniques. In most cases, these questions will be equivalent to investigating whether we can establish a relationship between a clinical event, such as shock or fluid administration, and a patient's PPG. If this process is successful, we would then be able to predict that event given a PPG signal. For example, if we were able to discover a pattern between a shock event and a patient's PPG, we would then, given a patient who is just admitted, be able to predict through their PPG whether they are, or will soon be, in shock. Ultimately, if this could be used to assist in early detection of disease progression, as conceptually easy as this process is, it would truly be able to save lives.

For some clinical questions preliminary work has already been done [6]. Thus, one of the main goals is to validate as well as to further explore these early results. Primarily, this can be achieved due to the existence of a new dataset. In Section 1.2.1, we present a brief overview of the new dataset and its contents in order to provide further context for the material we have to work with. After this, we will examine the technical objectives of this project in more detail by introducing the implementation pipeline in Section 1.2.2.

1.2.1 New Dataset

The new dataset was acquired in HTD (Hospital for Tropic Diseases) in Ho Chi Minh city, Vietnam in partnership with OUCRU (Oxford University Clinical Research Unit). Collected using a commercial wearable PPG device from SmartCare [7] featuring a finger probe, this dataset is larger and, in a sense, more complete than its predecessors, which were used in the aforementioned preliminary work. A simplified representation of the dataset structure is shown in Figure 1.1 in the form of a directory tree [1]. We see two folders within the patient folders in the raw dataset: PPG, and Monitor. The PPG folder refers to the commercial wearable PPG device whereas the Monitor folder refers to a GE-brand patient monitor with synchronized ECG and PPG acquisition. Finally, a patient can have more than one file in the case where there is more than one period of PPG recording. The clinical spreadsheet, on the other hand, includes a multitude of data for each patient, such as personal information (their age, sex, etc.), vital signs parameters (heart rate, oxygen saturation, etc.), shock and reshock date and time, fluid administration, and many others.

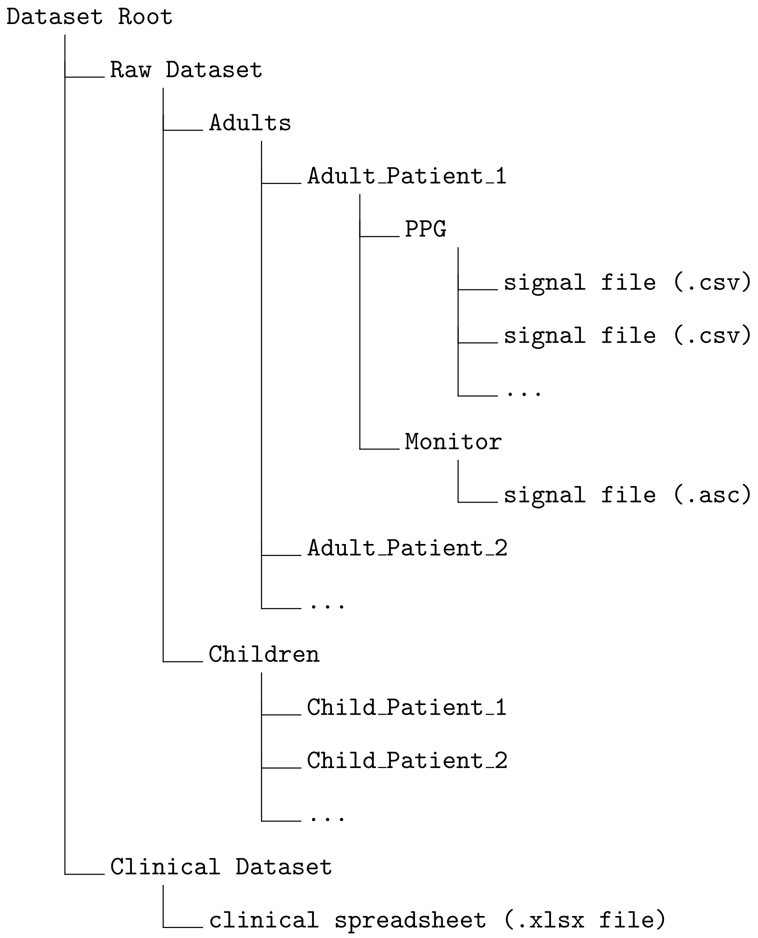
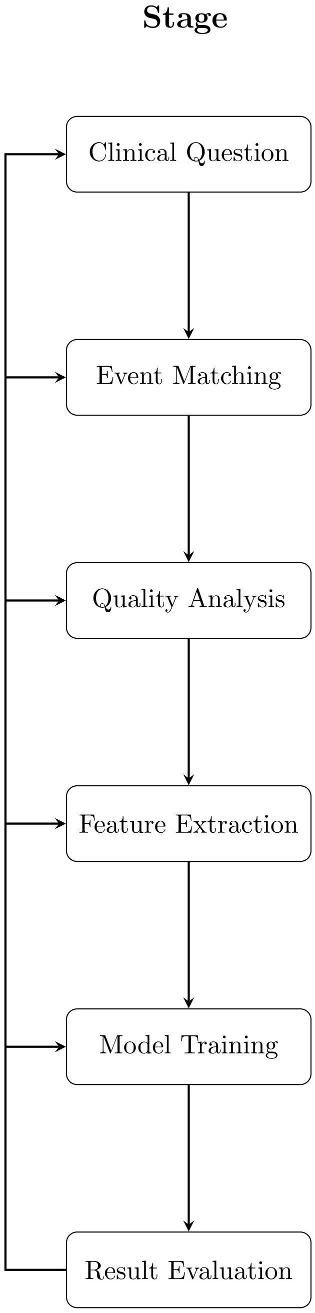


Figure 1.1: Simplified representation of the dataset structure in the form of a directory tree. The raw dataset contains the patients' PPG signals, whereas the clinical dataset includes vital parameters, as well as other clinical events, such as fluid administration and shock/reshock.

[1] We refer to the representation as simplified since the full depth (*i.e.,* from the dataset root to the furthest file) is only shown for one patient, as well as because we only show folders and files that are relevant to this project, and neglect the rest.

1.2.2 High-Level Pipeline

In this report, as well as throughout the project in general, we think about tasks and data flow in terms of an implementation pipeline. For this section, we present a simplified, high-level version of the pipeline, including only the core blocks of the project. Figure 1.2 depicts these main blocks, as well as a brief overview of each block. We will provide further details on all parts in Section 3.



Overview

The clinical question is the input to the entire pipeline. Obtaining it is a decision informed by both technical and clinical teams, since it is vital that it is both feasible and clinically significant. Through it we aim to investigate whether we can discover a relationship between a particular clinical event and a patient's PPG signal.

Given a clinical event, as specified by the clinical question, we iterate through all patients and retain the ones that have experienced this event, and whose PPG was being recorded at the time of occurrence. These two conditions ensure that we have sufficient data to investigate this clinical question for that particular patient.

If a patient is matched with a clinical event, meaning that they have sufficient data to support it, we perform an additional check to ensure that the signal segment of interest is of satisfactory quality, since PPG signal quality easily degrades due to noise. This is achieved through signal preprocessing and the use of Signal Quality Indices (SQIs).

Having obtained a signal segment of respectable quality containing the clinical event, we split the signal into smaller segments, transforming each of them into a feature map. As the choice of extracted features in this stage heavily influences overall performance, we aim to explore multiple techniques and algorithms.

At this point, for each of the patients matched with the clinical question, we have obtained the signal segment containing the clinical event, have ensured satisfactory quality, and have obtained a collection of feature maps. These feature maps, along with their corresponding labels, comprise our training set for the machine learning model.

Through the training and validation of the machine learning model we obtain a collection of performance metrics, as well as the training and validation curves. These provide useful information and can be used as feedback. However, this is not straightforward, since they are affected by all earlier stages, meaning that feedback is not direct for each stage.

Figure 1.2: Implementation pipeline showing the main blocks of the project (left) and brief summary for each of these blocks (right).

Finally, we note that perhaps the most important objective of this project is that the work done here is useful (and usable) to future research of the topic.

Chapter 2

Background

2.1 Dengue Virus

Dengue is a viral infection caused by the Dengue virus (DENV), part of the Flaviviridae family [1]. The transmission of the virus primarily occurs through the bites of infected female mosquitoes of the species *Aedes aegypti* and, less frequently, *Aedes albopictus*. While dengue can be found in many countries around the world, bringing more than half the planet's population at risk, outbreaks are most common in Central and South America, Southeast Asia, the Pacific Islands, and the Caribbean [8]. Figure 2.1 shows a global map where areas are coloured according to the level of risk of a dengue outbreak.

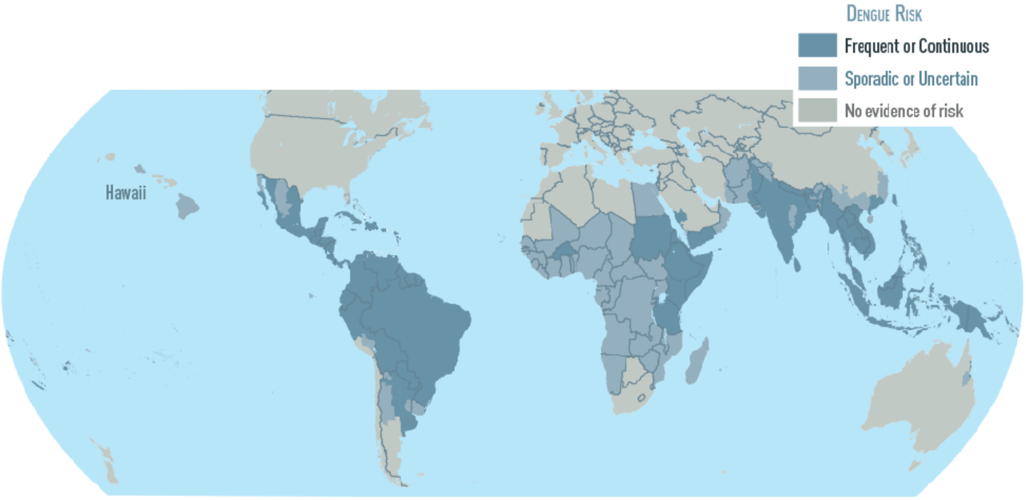


Figure 2.1: Dengue map, areas are coloured according to the risk level of a dengue outbreak (image from [9])

Dengue is often divided into two categories: dengue (with and without warning signs), and severe dengue [1]. In general, most cases exhibit only mild symptoms or are completely asymptomatic. For dengue, these include high fever, as well as possible headaches, joint and muscle pain, rash, vomiting, swollen glands, abdominal pains, and nausea. In the case of severe dengue and the further complication of Dengue Shock Syndrome (DSS), the patients may experience severe bleeding, respiratory distress, organ impairment, as well as circulatory collapse or plasma leakage [1, 10].

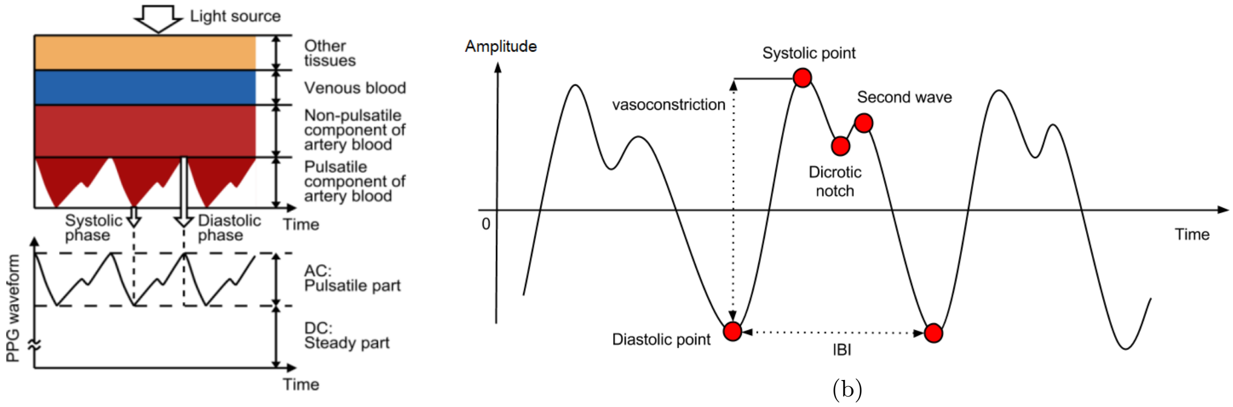
There are currently believed to be four distinct serotypes of the virus, DENV-1, DENV-2, DENV-3, and DENV-4, meaning that the same individual can be infected up to four times. It is worth mentioning, however, that it has been speculated that the differentiation according to serotypes may not paint the entire picture, and that instead, what is more significant are the antigenic differences between individual strains [11]. *Katzelnick et al.* (2015) found differences of similar order between strains of the same serotype and between ones of different serotypes, implying that "an individual infected with one type may not be protected against antigenically different viruses of the same type, and that in some cases the individual may be protected against some antigenically similar strains of a different type" [11].

Further complicating the situation, it has been observed that the second time an individual is infected, they exhibit more severe symptoms than the first time. *Halstead* (2002) proposed the concept of “antibody-dependent enhancement (ADE) of infection”, suggesting that the antibodies developed from the first infection serotype actually exacerbate the spread of the disease the second time [12]. Later research concluded that ADE does in fact occur, but only at a precise range of antibody concentrations. More specifically, only intermediate levels of antibodies aggravated the disease, while high levels offered protection against severe disease and low levels had little effect: “Too much or too little—better than some” [13].

There is no directed treatment specifically against dengue [1]. Recommended practices for symptomatic dengue patients consist of fluid administration for hydration and paracetamol as analgesic and antipyretic [10]. For severe dengue, patients might require blood transfusions in cases of substantial bleeding or urgent administration of intravenous fluids for resuscitation in the case of plasma leakage. Prevention for dengue mainly comes in the form of infection avoidance (suggested practice is use of mosquito repellent). A vaccine also exists, but this has been shown to be unreliable for general administration. Although the vaccine provides protection for individuals who have contracted the virus prior to vaccination, it turns out to be harmful to people who have never been infected in the past. For those, not only does it show increased hospitalisations relative to an unvaccinated control group, but also heightens the risk of severe dengue progression [14, 15].

2.2 Photoplethysmography

Photoplethysmography (PPG) is a measuring technique used to detect changes in blood volume at a particular measuring spot [16]. This is achieved via the use of optical devices, namely a light source and a photodetector. When the light source is directed into skin, part of the light is absorbed by skin tissue, as well as by the blood vessels within that tissue. If the wavelength of the light is within a specific range (more specifically around the boundary between visible and infrared), the amount absorbed by the skin tissue itself is constant and small in magnitude. This is turn means that the remainder of the light, which will be detected by the photodetector, can be used to estimate volumetric changes in blood content. Although the absorption, reflection, and scattering processes of light within tissue and blood vessels are not simple, in general, the larger the amount of blood within the vessels, the larger the light attenuation [4, 5, 16]. This process is depicted in Figure 2.2a along with a sample PPG waveform in Figure 2.2b.



(a)

(b)

Figure 2.2: (a) Principle of operation for a PPG device (image from [17]) (b) A sample PPG waveform (image from [18])

In modern PPG devices, the light source and photodetector can be placed in two configurations: on opposite sides of the measuring site tissue (transmission mode), or on the same side (reflection mode) [19]. Figure 2.3 depicts these two possible modes for PPG devices using finger probes. PPG sensors can also utilise different light sources. The most common choices are infrared light emitting diodes (IR-LEDs), red LEDs, and green LEDs, each having slightly different characteristics [4, 5, 17]. The device responsible for the data acquisition of this project used a red, as well as an infrared LED, resulting in two distinct signals [7]. It is important to note here that LEDs, as well as photodetectors, are very simple components, and it is because of this simplicity in hardware that PPG devices are considerably cheaper than other alternatives within the monitoring device market.

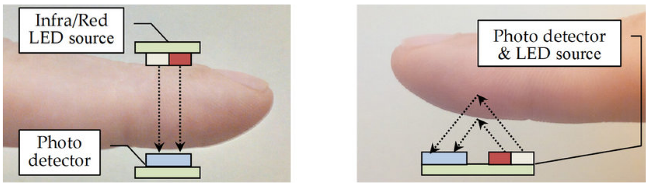


Figure 2.3: Transmission (left) and reflection (right) modes of a finger PPG sensor (image from [19])

Regarding real world usability, the clinical utility of PPG devices has been a topic of research for the larger part of the past century. For a detailed review of the (early) history of PPG research, and since historical analysis is beyond the scope of this report, we refer to two excellent review articles: *Challoner et al.* (1974) [16], and *Allen* (2007) [4]. In recent decades, aided by technological advancements in the areas of optoelectronics, semiconductors, digital signal processing, and clinical instrumentation in general, PPG has gathered more and more interest as a clinical tool [4, 5]. In the past years, this has further been enforced by the large rise in popularity of wearable devices, both for clinical, as well as for health and fitness purposes. In the current era, research has used PPG to measure (with varying degrees of success) several vital signs, such as heart rate, blood pressure, oxygen saturation, respiratory rate, hematocrit and temperature, further increasing its potential as a monitoring device. Details of how these are achieved will be explored in Section 2.3.3. Despite the above, the potential of PPG is still being investigated.

2.3 Processing and Analysis of PPG Signals

The raw dataset as presented in Section 1.2.1 cannot directly be used as the input to the machine learning algorithms of the pipeline. This is because it is relatively unstructured (signals for each patient are of different length, patients can have more than one PPG recording), and may contain parts of unusable quality (patients may not be wearing the sensor correctly or at all, noise may be too significant in some segments). The aim of the steps presented in Sections 2.3.1, 2.3.2, and 2.3.3, is to transform the unstructured and noisy raw dataset to a structured and cleaner representation of our data, which can then be used as input to our machine learning models. This process consists of preprocessing, quality analysis, and feature extraction, each of which will be analysed in detail in the next sections.

2.3.1 PPG Preprocessing

Preprocessing refers to a collection of algorithms the main goal of which is to remove unwanted noise from the PPG signal. In our application, preprocessing consists of filtering, normalising, and smoothing.

2.3.1.1 PPG Filtering

PPG signals are highly susceptible to noise interference. This can be attributed to several factors, such as the surrounding environment, respiration, and motion artefacts [20, 21]. Some of these forms of noise even overlap with the signal in the frequency domain, bringing forth the need for effective filtering to reject as many unwanted signal components as possible. It's important to note that the literature on signal denoising is vast, meaning that there exists a very large number of techniques and algorithms for the task. Examples include machine learning methods, such as autoencoders, digital filters (FIR/IIR), wavelet denoising, dynamic time warping, clustering, and heuristic algorithms [22]. Since advanced techniques for optimal denoising are beyond the scope of this project, we limit our background research and use to conventional filters.

Effective filtering requires sensible choices both for the filtering frequency range, as well as for the filter type. Research exists for both of these aspects. Regarding the former, most studies use a filter with lower cutoff frequency in the range of $0.1 - 1$Hz and an upper cutoff frequency of $10 - 20$Hz, although this isn't always the case. For example, *Moraes et al.* (2018) claim that the PPG pulse range exists within the range of $0.5 - 4$Hz, implying a similar passband for the filter [21]. Another study perhaps worth mentioning is by *Allen et al.* (2004), who investigated the cutoff frequency of the highpass filter, ultimately suggesting 0.15Hz [23]. *Waugh et al.* (2018) also used a passband of $0.15 - 20$Hz in their study [24].

Regarding filter type, *Liang et al.* (2018) performed a thorough investigation on short ($2.1s$) PPG signals, experimenting with 9 different filter types, each with 10 orders. Using the skewness SQI, which will be discussed in detail in Section 2.3.2, as a performance metric, they found that optimal filtering was achieved with a 4^{th} Chebyshev II filter, surpassing the previously considered "gold standard", the Butterworth filter. They also validated that "the lower the order, the better the filter performance in analyzing biomedical signals" [20].

2.3.1.2 PPG Normalising and Smoothing

The amplitude of the AC component of the PPG varies wildly from application to application, as it is dependent on a large number of factors, such as the measuring spot, the light source and photodetector, and the gain used internally by the sensor. We therefore require normalisation in order to standardise the dynamic range of our signal to $[-1, 1]$. Regarding smoothing, again the literature is extensive. We can approach smoothing through different types of averaging, such as moving average or the Savitzky-Golay algorithm, or by using other methods, such as spline smoothing.

2.3.2 Signal Quality Indices (SQIs)

Signal quality indices (SQIs) are mathematical concepts or algorithms used to describe the quality of a signal. In this project, we utilise SQIs for PPG quality analysis, and more specifically to reject PPG signal segments that are of poor quality due to large amounts of noise. These signal segments would not be beneficial to the training of our machine learning models, and could even prove to be harmful since they provide inaccurate representations of PPG signals.

The excellent article by *Elgendi* (2016) investigated eight different SQIs, perfusion, kurtosis, skewness, relative power, non-stationarity, zero crossing, entropy, and the matching of systolic wave detectors, in order to find the optimal SQI for classifying PPG signal quality [25]. He found that the skewness SQI was best for assessing quality, outperforming the previously considered "gold standard", the perfusion index SQI. That said, the study was conducted using only healthy subjects. Furthermore, it was reported that the performance of the skewness SQI deteriorated as the window size, on which it was calculated, increased. We should keep this in mind, since our work will more than likely be using segments longer than $3 - 5s$. Finally, investigating the use of a combination of SQIs is also mentioned as future work [25].

Several different techniques and algorithms for PPG quality checking have also been found, although most of these lie beyond the scope of this project. One example is by *Orphanidou* (2018), who proposed an SQI which labelled a signal segment as "good" or "bad" based on whether a reliable heart rate could be derived [26]. Other more advanced techniques that we discovered in literature include dynamic time warping, fuzzy neural networks, deep convolutional neural networks, and random forest classifiers [27, 28, 29, 30].

2.3.3 Feature Extraction

Feature extraction refers to the process of creating a representation of a given signal by means of a particular method or algorithm. The aim of obtaining this representation rather than using the original signal is to reduce the amount of data to be processed, while simultaneously retaining the useful information from the signal, or even extracting information that wasn't already there. In the past, feature extraction for PPG signals has followed both blind and insightful approaches, as defined by *Elgendi et al.* (2018) [31]. An insightful approach refers to a system which extracts or localises features from the PPG signal or parts of it, whereas a blind approach rather uses the entire raw signal. We investigate both of these in the next sections.

2.3.3.1 Raw Signal, Derivatives, and Localised Features

The most straightforward choice for feature selection is to use the the entire raw signal after preprocessing. This is the easiest approach, but also the most "blind" one. An extension of this would be to use localised

features from the raw signal, such as the amplitude of specific points, or the distance between two of them. *Elgendi* (2012) carried out extensive analysis on these localised features, along with the corresponding clinical phenomena and causes linked to each feature [32].
Besides the raw signal itself, we can look at the first and second derivatives of the PPG signal, as these have been shown to hold important information [32]. *Takazawa et al.* (1998) introduced the first and second derivatives of the PPG signal [33]. *Elgendi* (2012) provided a thorough review as well as further analysis on the first and second derivative in [32], and also attempted to standardise terminology on these features in [31]. As in the case of the raw signal, we can extract specific local features from the derivative signals. Again, *Elgendi* (2012) extensively presented this in [32].

2.3.3.2 Vital Signs and Clinical Data

Using vital signs and clinical data as features is probably the most intuitive choice, at least from the clinical staff's perspective, since it is these clinical features that they themselves use for a diagnosis. We have found several examples where these vital signs were directly used as features in the ML algorithms [34, 35, 36]. We will explore the actual performance of these ML models in Section 2.4.2. We see two ways to obtain such features. The first way is to directly access them from the clinical dataset. The largest issue with this is that vital signs are not taken regularly in a standardised manner across all patients, and so it would be very hard to create a well-structured dataset of clinical features solely from the clinical spreadsheets. The second way, which is more involved but also more promising, is to estimate these vital signs from the PPG signal and its derivatives. This has the advantage that we can estimate these at any points where we have a PPG signal.

It is important to note that we are only interested in vital signs that have been shown to have some kind of correlation with dengue, as this would imply that there may be some benefit in using these vital signs as features. Below we have categorised some of the common clinical features that are related to dengue and have been investigated through PPG signals. We note that we do not present this as a thorough literature review of all the research on vital sign feature extraction from PPG, but rather as a collection of vital signs that we have found which have been shown to be correlated to dengue and which can be estimated through the PPG signal.

1. **Cardiovascular:** Dengue has been observed to have an effect on the cardiovascular system, although "the pathophysiology of cardiac disease in dengue infection is unclear" [37]. Despite bradycardia being the most common cardiovascular abnormality, others have also been observed [37, 38], meaning related clinical features might contain useful information.
   (a) **Heart rate:** Heart rate is the most basic and easiest vital sign to estimate since the PPG period is closely linked to a single heart beat. *Reiss et al.* (2019) provide a review of the classical methods to heart rate estimation, as well as their novel deep learning approach in [39].
   (b) **Blood pressure:** Cuff-less methods to estimate blood pressure (BP) using only the PPG signal have been proposed in recent years. One example is by *Addison* (2016), who introduced slope transit time (STT), a method which unlike its predecessor, pulse transit time (PTT), which required both ECG and PPG measurements, needs only the PPG signal [40] [41]. Motivation for BP estimation through PPG was further enforced by *Martínez et al.* (2018), who concluded that "PPG holds most informative features that exist in [arterial blood pressure] ABP" [42].
   (c) **Aging index (AGI) and arterial stiffness:** First investigated by *Takazawa et al.* (1998) through the amplitudes at several points of the second derivative [33], and was reviewed by *Elgendi* (2012) in [32]. An algorithm for this process was proposed by *Pilt et al.* (2013) in [43].

2. **Respiratory:** Dengue can also have respiratory and pulmonary manifestations [44, 45]. According to the WHO, one of the potential symptoms of severe dengue is respiratory distress, implying that respiration rate and oxygen saturation (SpO2) may offer useful information [1].
   (a) **Respiratory rate:** The PPG waveform is heavily influenced by respiration rate and so its detection has been established and is relatively straightforward [46, 47].
   (b) **Oxygen saturation (SpO2):** The most common usage of PPG signals is for pulse oximetry, the measurement of oxygen saturation. We refer to the excellent progress article by *Tamura* (2019) for further information on various aspects of pulse oximeters [48].

3. **Blood components and features:** *Chaloemwong et al.* (2018) investigated the differences in complete blood count (CBC) between patients with acute febrile illness as a result of dengue infection, and as a result of other causes, and found that "the dengue group [compared to the control group] had higher haemoglobin levels and a higher hematocrit as a result of the plasma leakage" [49]. *Ralapanawa et al.* (2018) also found a significant difference in haemoglobin value between dengue patients who progressed to severe dengue and patients who did not. These results imply that extraction of haemoglobin and hematocrit value might be of interest.
   (a) **Haemoglobin value:** *Kavsaoğlu et al.* (2015) investigated this by using several PPG features, feature selection algorithms, and machine learning algorithms [51]. In fact, they extracted 40 time-domain features, 21 of which were from the PPG signal, 8 from the first derivative, 7 from the second derivative, and 4 from the combination of the two derivatives.
   (b) **Hematocrit:** Hematocrit estimation through PPG is a challenging task. Although we found studies exploring this, the literature was visibly much more limited [52, 53].

2.3.3.3 Time-Frequency Representations

Time-frequency analysis is a feature extraction approach which is very commonly used for PPG data, as well as for biomedical signals in general. Time-frequency methods are used to create a representation which displays the signal in the time and frequency domain simultaneously. The motivation behind this lies in the fact that these two domains are closely linked for any function, suggesting that having both together might result in a powerful representation.

Traditionally, the algorithms used are the Short Time Fourier Transform (STFT) and the Continuous Wavelet Transform (CWT). For the STFT, the signal is split into windows, and the Fourier transform is applied on each one individually. This leads to a uniform time-frequency spectrum, the spectrogram [54]. The CWT does not explicitly window the function, but rather produces a time-frequency representation by convolving the signal with a new function, the wavelet. The wavelet function is scaled (stretched or compressed along the time axis) and then shifted (translated along the time axis). As the wavelet is scaled by different factors, changing its frequency range, and shifted along the signal, this creates a time-frequency spectrum, the scalogram. Due to this scaling and shifting approach, the scalogram offers varying time-frequency resolution, which is often seen as an advantage over the STFT's fixed time-frequency resolution [54]. This distinction is displayed in Figure 2.4.

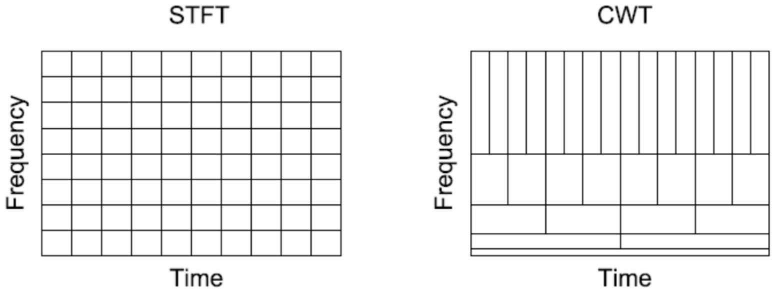


Figure 2.4: STFT spectrogram (fixed time-frequency resolution) and CWT scalogram (varying time-frequency resolution) (image from [55])

That said, this technical difference has not prevented studies from utilising the STFT. For example, *Tjahjadi et al.* (2020) used the STFT for feature extraction for PPG signals, while *Allen et al.* (2021) used the CWT [56, 57]. These are only two of the numerous examples using time-frequency analysis as a feature extractor for PPG data. It is worth noting that a plethora of more advanced time-frequency analysis algorithms have been developed and studied in order to improve upon the time-frequency spectrum resolution of STFT and CWT. *Wang et al.* (2006) compares various of these algorithms, such as continuous wavelet transform (CWT), fixed-frequency complex demodulation (FFCDM), and variable-frequency complex demodulation (VFCDM), with regards to resolution [58]. Additionally, some of these have been compared with regards to PPG data [59]. Finally, we note that an improved time-frequency resolution does not necessarily imply better feature extraction, since other factors such as sparsity and dimensionality of feature space are also very important.

2.4 Machine Learning

Through its immense rise in popularity in the past decades, machine learning (ML) has established itself as an incredibly powerful tool, delivering high performance in tasks such as classification, regression, and pattern recognition across a vast range of applications. Despite being heavily affected by two of ML's most pressing issues, its interpretability and explainability, the biomedical and clinical sectors have been no exception, and have seen a plethora of ML applications.

In Section 2.4.1 we provide a brief overview of some of the ML algorithms that are relevant to this project. By relevant we mean that we believe they are likely to be considered and utilised in the "Model Training" block of the high-level pipeline, as presented in Section 1.2.2. In order to further support our decisions, for each of the ML models presented in Section 2.4.1 we also note studies that have used these types of models in an application for dengue or PPG (Section 2.4.2).

2.4.1 Relevant ML Algorithms

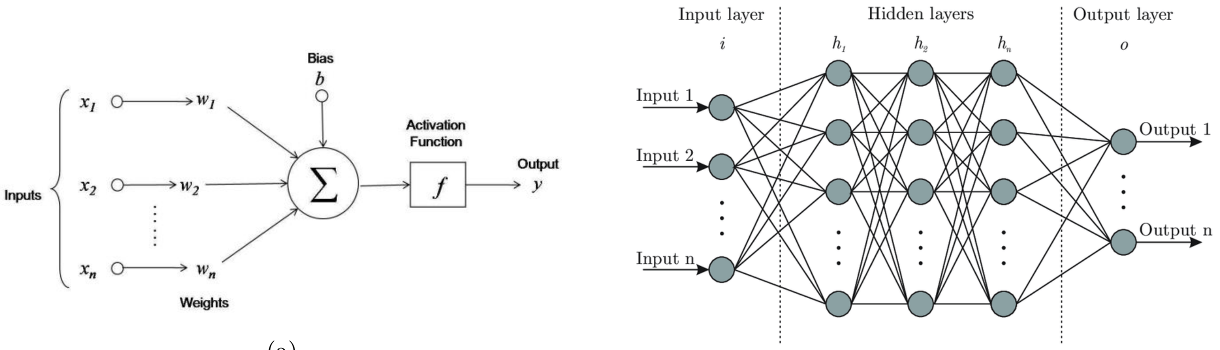
Given that our aim is to train using labelled segments of the PPG signals, we only consider supervised ML algorithms in the next sections. Supervised machine learning refers to the subset of ML algorithms which work and achieve learning through labelled data. Naturally, the labels in this project are dependent on the clinical question. For example, if we were investigating whether a patient is in shock, each segment would have to have a binary label indicating the shock status (1 - shock, 0 - no shock). Note that labels do not have to be binary.

2.4.1.1 Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are a class of supervised machine learning algorithms loosely inspired after the operation of the biological brain. They are formed through the interconnection of a collection of individual units named artificial neurons. The structure of an artificial neuron and of an example artificial neural network are shown in Figures 2.5a and 2.5b respectively. The artificial neuron operates by multiplying each input with a weight, summing these products along with a bias, and finally passing that sum through an activation function. The output y is given by the equation below, where the second expression is derived by defining $x_0 = 1, w_0 = b$.

$$y = f\left(b + \sum_{j=1}^{n} w_j x_j\right) = f\left(\sum_{j=0}^{n} w_j x_j\right) \tag{2.1}$$

The ANN is constructed by placing consecutive columns of artificial neurons called layers. The first layer is called the input layer, followed by a number of "hidden" layers and finally an output layer. Every neuron of layer j is connected to all neurons of layers $j-1$ and $j+1$, meaning that its inputs are the weighted outputs of the previous layer's neurons, and its output after being weighted is one of the inputs of the next layer's neurons. This is the reason we refer to the layers of ANNs as "fully connected" or "dense".



(a)

(b)

Figure 2.5: (a) Structure of an artificial neuron unit (image from [60]) (b) Example of the architecture of an artificial neural network (ANN) (image from [61])

Learning in the ANN is achieved by changing the weights across the network in order to minimise a loss function. When an input is fed through the network, a process called a feedforward pass, the loss function is computed at the output layer. This refers to the error between the network's prediction for a particular input and the correct output for that input, the label. Based on this error, or more precisely the average of the error for several training examples, all the network parameters, consisting of the weights and biases, are adapted according to the optimiser and the backpropagation algorithm. These two algorithms combined are able to determine how much each of the network parameters should change in order to minimise the loss function. The process of feedforward, backpropagation, and weight update is repeated until the optimiser deems the current value of the loss function to be at or near a global, or most likely local, minimum.

There is a large number of parameters that can be changed in an ANN model, such as network architecture (number of layers, neurons per layer, activation function), optimiser and optimiser learning rate, batch size, and regularisation strategies, just to name a few. Thus, finetuning is a task that must be approached in a structured way. We will provide further information on how this is attempted in our project in Section 3.5. Finally, it is worth noting that for this section, as well as for the ones that follow, we will not analyse in depth all the aspects of these complex systems, and rather aim for conciseness and clarity.

2.4.1.2 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are usually constructed as a two-stage system. The second stage of this pipeline is an ANN, that is, a series of fully connected layers leading to one or more output neurons. The number of output neurons as well as the activation function, just like in the pure ANN case, depend on the task the network is being trained for, such as binary classification, multi-class classification, or regression. The first stage of the system, which precedes the fully connected layers, is essentially a feature extractor, the main highlight of which is that it is designed to operate with 2-dimensional inputs. The core of this feature extraction stage is convolutional and pooling layers [62]. Figure 2.6 depicts a common and influential CNN architecture named VGG-16 [63]. We can see that the input image is passed through several convolutional and pooling layers before reaching the fully connected layers and finally the output.

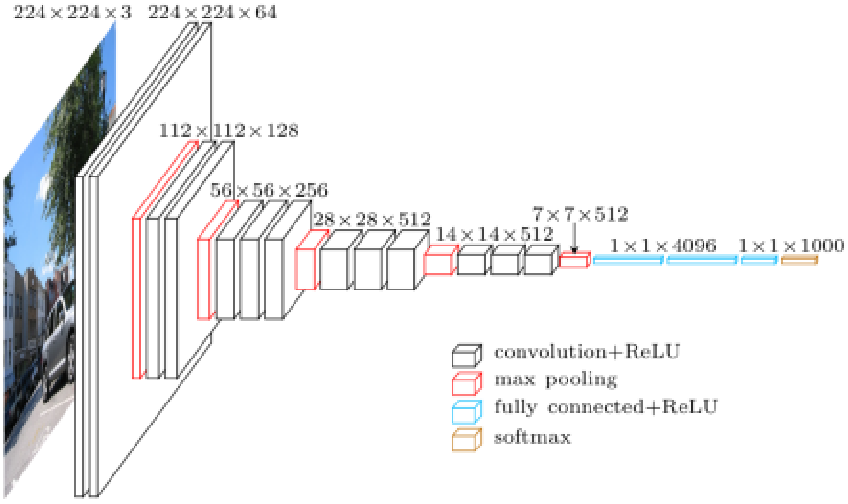
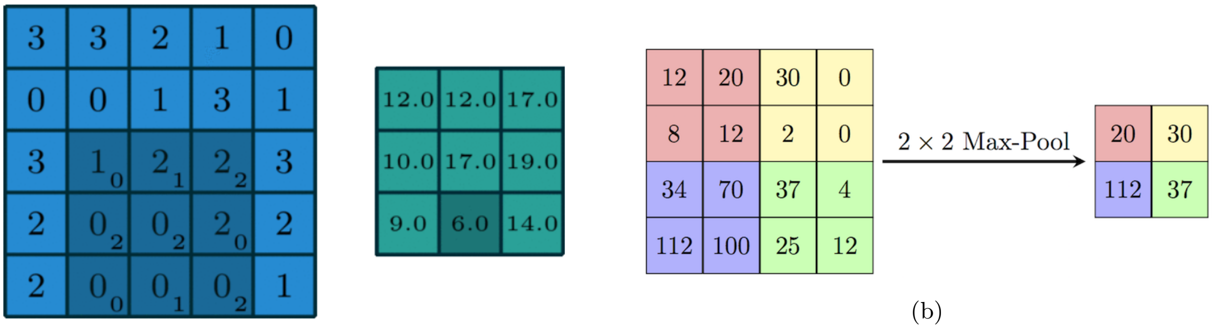


Figure 2.6: Convolutional and pooling layers followed by fully connected layers in the VGG-16 CNN (image from [64])

Convolutional layers operate by sliding a filter over the input feature vector. In CNNs both the filter and the feature vector are 2D images, mathematically denoted by matrices. The sliding process refers to performing an element-wise multiplication between the filter and a part of the feature vector and subsequently moving the same filter across the image and repeating. This process can be seen in Figure 2.7a. Note that the filter is (3x3) and its values are the smaller numbers in the bottom right corner of the darker cells. It is this localisation of features provided by the filter within the input that has allowed CNNs to achieve groundbreaking performance in image-related tasks. The result of this process is another feature map, the matrix on the right. This will then be used as the input image to the next stage.

Pooling layers are used with the aim of downsampling the input representation, reducing its dimensionality and thus the network's computational cost. The most common types of pooling work by picking the maximum, average, or minimum element of a part of the image, leading to max, average, and min pooling. This process is shown for max pooling in Figure 2.7b, where the pooling size is (2x2), meaning that the algorithm picks the biggest number from each (2x2) block, reducing the number of features in the result.



(a)

(b)

Figure 2.7: (a) Principle of operation of convolutional layer, filter is slided across the input image and element-wise multiplication between the filter and the input image occurs at each point (image from [62]) (b) Principle of operation of pooling layer, max pooling with pooling size (2x2) is shown (image from [65])

Learning in CNNs occurs with the same exact way as in ANNs, through backpropagation and an optimiser. The weight that are trainable in this case are not numbers which multiply the outputs of neurons, but rather they are the parameters of the filters, since it is through these filter weights and the elementwise multiplication that the features are localised and extracted. Note that pooling layers, contrary to convolutional layers, do not have trainable parameters. Regarding tuning, CNNs contain all changeable parameters of ANNs along with many more, and so are even harder to tune. As deep learning and computer vision have now been the topic of research for long enough, successful CNN architectures, such as VGG-16 as shown earlier, and good practices have become known, aiding us in the exploration of CNN models.

2.4.2 ML for Dengue and PPG

Machine learning algorithms and techniques have already been applied extensively for both dengue and for PPG signals. Regarding PPG, ML models have been developed both to detect or predict a variety of diseases, as well as to estimate vital signs from the PPG signal. For dengue, ML models have been created in order to explore several aspects of the disease, such as management, prediction, and prevention. Very seldom, however, have ML models that combine PPG and dengue been explored. For us, this is both a positive and a negative fact, since it entails that our work is innovative and to a degree unexplored, but also that there doesn't exist a body of literature which can be used as reference or to provide confidence.

Table 2.1 shows a summary of some of the machine learning-related works that we found. As we previously mentioned the literature on both PPG and dengue separately is extensive, and so we only show some of the studies that piqued our curiosity. In the table we include whether the application is PPG, dengue, or both, the type of ML model used, the inputs, the broad aim of the study, and a reference. For conciseness within the table we use abbreviations for the different ML models. These can be found in Appendix A.

We also comment further on one study in particular [66]. This study refers to an FDA-approved medical device which uses feature extraction and ML algorithms with the aim of detecting haemorrhage. Unfortunately, however, the system, which is named Compensatory Reserve Index (CRI), is owned by Flashback Technologies and thus, even though their study appears to provide meaningful results, no information on internal algorithms is given. Finally, its proprietary nature makes it expensive, meaning that mass population application is unlikely.

Topic	**ML model**	**Inputs**	**Aim**	**Reference**
Both	ANN, CNN, LSTM	Time-Frequency	Investigate the relationship between a PPG signal and dengue severity	(2021) [6]
Both	-	-	Use CRI to detect shock and track the progress made through fluid resuscitation	(2016) [66]
PPG	CNN	Time-Frequency	Outperform classical HR estimation methods through a novel deep learning approach	(2019) [39]
PPG	CNN	Time-Frequency	Detect peripheral arterial disease by finetuning a pretrained AlexNet CNN	(2021) [57]
PPG	CNN, LSTM, variations	Time-Frequency	Detect and delineate PPG beats under noise-corrupted signals	(2021) [67]
Dengue	ANN, SVM, RF, XGBoost	Clinical and personal data	Predict whether a dengue patient would go into shock during their hospitalisation period	(2022) [68]
Dengue	DT, RF	Clinical and personal data	Predict dengue fever	(2020) [34]
Dengue	MLR, MnLR	Clinical data	Predict dengue case fatality	(2021) [35] [36]

Table 2.1: Different types of ML models along with their inputs and aims applied in the topics of PPG, dengue, or both

Chapter 3

Implementation Plan

The implementation plan for this project is based on the high-level pipeline presented in Section 1.2.2. In Sections 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 we aim to further break down and investigate each of the core blocks of the pipeline, presenting their current state and the progress made so far, and subsequently focusing on future steps. Then, in Section 3.7 we present a project schedule by means of a timing diagram.

It is important to note before delving into technical details that one of the main aims of the project is to design and implement the entire system to be modular. More specifically, we pursue modularity in being able to ask different clinical questions, as well as in the ability to easily experiment with different preprocessing, quality checking, feature extraction, and machine learning methods. For these stages we base our design on the literature presented in Chapter 2. That said, we do not consider implementing "as much as possible" from what is presented in Chapter 2 a project objective. In other words, the aim of modularity is not to increase the quantity of implemented material, but rather to enable effective experimentation and exploration of the problem.

3.1 Clinical Question

As explained through the timing diagram of Section 3.7, deciding or settling on a clinical question is not a priority. In fact, it is considered insignificant for the completion of the MVP. This is for several reasons, which we list below:

1. Proposing and deciding on clinical questions is dependent on other parties besides the technical team at Imperial. More specifically, there is input from the clinical team in the hospital in Vietnam, making coordination more challenging.
2. More importantly, identifying what questions can be asked involves analysing the dataset, as it requires knowledge of the clinical events that are encapsulated within the PPG recordings of the patients. We investigate the relevant problem as well as the solution in detail in Section 3.2.
3. Being undecided on the clinical question does not hinder the development or implementation of any part of the remaining system. Thus, we are able to begin implementing the underlying framework without knowledge of the exact clinical question we will be exploring.

That said, the clinical team has recently suggested a collection of clinical questions that could be considered. Thus, prior to evaluating these suggestions, it is important to establish quality criteria for the clinical questions. These were decided to be clinical significance, feasibility, and the size of the dataset with which they can be investigated. As a first step for this stage we consider only the first two criteria. These, unfortunately however, are to a degree contradictory, presenting us with a tradeoff. More specifically, the clinical team's suggestions offer excellent clinical significance but perhaps limited technical feasibility, while the technical team presents questions which are closer to reality in terms of feasibility but with potentially less significance for a clinical setting. It is expected that we will find a middle ground between the two. However, as aforementioned, this is not a priority at the moment.

3.2 Clinical Event Matching

3.2.1 The Mismatch Problem

Given the contents of the clinical dataset as mentioned in Section 1.2.1, we should, theoretically, be able to ask clinical questions related to any subset of this dataset. For example, given that the clinical dataset includes details on fluid administration, such as which patient was administered fluids, the type and amount of fluid, and the date and time of administration, we should be able to investigate the effect of fluid administration on a patient's PPG. This, unfortunately, is not the case. For us to be able to ask a particular clinical question for a certain patient we require that the events related to that question occur within the period of this patient's recorded PPG. This is best illustrated with an example from one of the patients, as depicted in Figure 3.1.

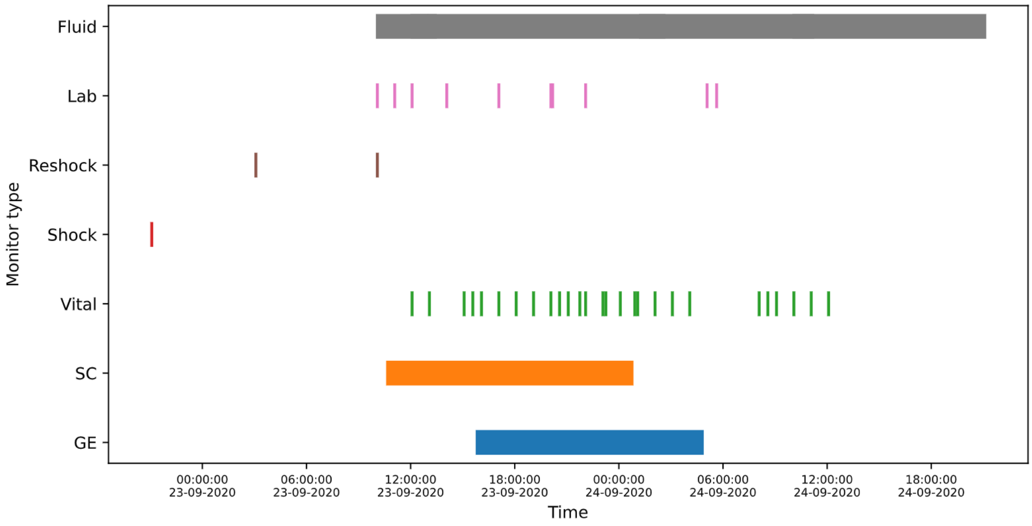


Figure 3.1: Summary of data for one patient. This includes the time intervals where their PPG signal was extracted (SC and GE graphs), as well as any clinical events.

SC (orange) and GE (blue) refer to the two monitoring devices that extract a PPG signal from the patient. We can see that the period during which fluids were being administered (grey) overlaps with the period of PPG recording. This would therefore allow us to investigate the effect of fluid administration further. However, looking at shock and reshock events, these do not occur within the period of recording, and therefore we would not be able to use this patient if we wanted to investigate the effect of shock or reshock on the PPG.

3.2.2 Matching Algorithm Development

In light of the mismatch problem presented in Section 3.2.1 we develop an algorithm which is able to match patients based on a particular clinical question. We can express the functional requirements as follows. Given a clinical question we should be able to iterate through all patients and determine the subset of them who are able to support this question with sufficient data from their PPG recordings. Furthermore, we should note the relevant time interval for each of those patients. By relevant time interval we mean that if a patient's PPG recording overlaps with the clinical event under consideration, we should extract a time interval of interest, for example from some time before the event until some time after the event. Naturally, this is dependent on the clinical event itself. If, for example, we are looking at shock we might focus on three distinct intervals: pre-shock, during-shock, and post-shock. If, however, we were investigating the effect of fluid administration we might only consider time intervals for pre-administration and post-administration. The lengths of these time intervals are also dependent on the clinical question.

Part of the solution had already been developed prior to the beginning of this project. The algorithm is currently able to iterate through the entire dataset and output information for each patient. This is in the form of a Python dictionary in which the key is the patient ID and the value is a collection of information for that patient.

The dictionary for one patient is shown in the code listing of Figure 3.2. We note the following:

- The files from the GE and SC monitors are represented in a list because a patient can have more than one PPG recording time period.
- "Dataframe" refers to the main data structure of the Python data analysis library *pandas*. The fields listed underneath are columns of that dataframe.
- For the shock and reshock dataframes the *Time* field refers to the time of occurrence, while for vital signs and lab tests this refers to the time of measurement.

```
{patientID: # Different files from GE monitor
            [(start_datetime, end_datetime, duration), ...]

            # Different files from SC monitor
            [(start_datetime, end_datetime, duration), ...]

            # Vital signs dataframe
            Time, Temperature, Pulse, LB1, SystolicBP, DiastolicBP, RespiratoryRate, SpO2

            # Shock dataframe
            Time, Pulse, SystolicBP, DiastolicBP, RespiratoryRate, Hematocrit, Temperature

            # Reshock dataframe
            Time, Pulse, LB1, SystolicBP, DiastolicBP, RespiratoryRate, Hematocrit, Temperature

            # Lab tests dataframe
            Time, WBC, HCT, PLAT, UREA, CREAT, AST, ALT

            # Fluid administration dataframe
            StartDatetime, EndDatetime, FluidVolume
}
```

Figure 3.2: Matching dictionary for one patient. The key is the patient ID and the value is a collection of information about that patient.

It should be mentioned that it is straightforward to change the algorithm producing the matching dictionary, meaning that it is easy to remove fields that we do not need, as well as to add fields that we need and which are not already there. Besides this, the remainder of the algorithm only has to iterate through all the patients, that is, all the keys of the dictionary, and verify whether the patient can be matched with the question. There are several reasons why a patient cannot be matched. These are listed below:

- The dataframe corresponding to the clinical event under investigation is empty.
- The time of occurrence / time period of the clinical event does not exist within the time span of one of the files of the SC monitor.
- There is insufficient data before and after the time of occurrence / time period.

The algorithm is depicted as a flowchart in Figure 3.3. The algorithm output should be a list of 4-tuples, more specifically (*patientID, start_datetime, end_datetime, label*). We will refer to this list of tuples as the "patient matrix" in the remainder of this project and report. *start_datetime* and *end_datetime* refer to the beginning and end of the time period of interest for the event under consideration for that patient. Since each of the tuples in the list will later in the pipeline represent an input for a supervised ML model, we require a label for this input, and so we include this field in the tuple. For example, if the task was to classify a segment as shock or no shock, then this information would be included in the label. Finally, note that we need to keep in mind that since we are developing ML models we prefer to have a balanced input dataset, meaning that each class, for example the shock and no shock classes, is of approximately the same size.

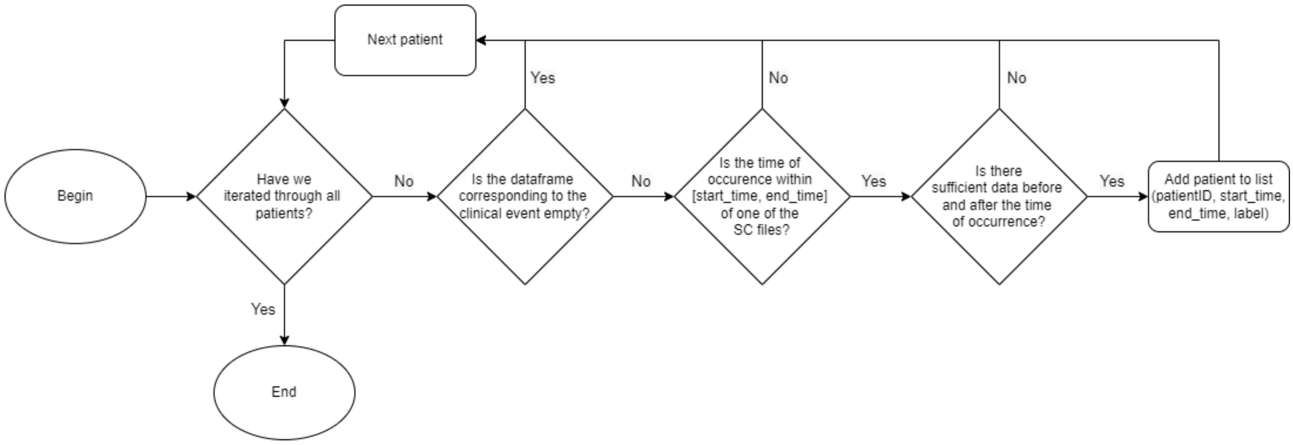

Next patient
Yes
No
No
Begin
Have we iterated through all patients?
No
Is the dataframe corresponding to the clinical event empty?
No
Is the time of occurence within [start_time, end_time] of one of the SC files?
Yes
Is there sufficient data before and after the time of occurrence?
Yes
Add patient to list (patientID, start_time, end_time, label)
Yes
End

Figure 3.3: Matching algorithm flowchart, each patient has to pass several staged checks (shown here as conditionals) in order to be added to the list of matched patients.

3.3 Quality Analysis

3.3.1 Preprocessing

Preprocessing currently consists of filtering and normalising. For filtering we have created a function *filter_signal*. The input parameters to this function are the signal, a dictionary of filter parameters as shown in the code listing of Figure 3.4, and a Boolean for optionally plotting the filter frequency response. The function creates the filter using these parameters and subsequently filters the signal using cascaded second-order sections. In reality, the function is a wrapper for several functions of the popular Python signal processing library *scipy*, and was written for modularity and ease of use.

```
filter_params = {
    "type" : "cheby",
    "order" : 4,
    "sample_rate_Hz" : 100,
    "low_cutoff_Hz" : 0.15,
    "high_cutoff_Hz" : 20
}
```

Figure 3.4: Filter parameter dictionary

For the filters, as seen in Figure 3.4, we are currently using a 4^{th} order Chebyshev II filter with a $0.15 - 20$Hz passband, as suggested by *Liang et al.* (2018) in [20]. After filtering we normalise the signal to $[-1, 1]$. Smoothing has not yet been researched sufficiently and is thus not implemented.

3.3.2 Quality Analysis

For quality analysis we have adopted a perhaps unexpected approach. Instead of performing quality checking on each individual signal segment as part of the pipeline, we carry out a process on the entire dataset independently and prior to running the pipeline itself. This process includes undertaking quality analysis for all patients throughout the dataset and noting parts that are of poor quality. We chose this approach for the sole reason that quality analysis is dependent only on one factor, the length of the signal segments. Note that we are thinking in terms of signal segments as it is these smaller parts of the signals that we will later be performing feature extraction on. As we expect that most clinical questions will be using the same signal segment length, at least in the beginning of our exploration, if we performed quality checking as part of the pipeline each time, then this would be unnecessary repetition of the same process. Thus, we avoid this by carrying out quality analysis only once outside of the pipeline.

For this process we developed the function *unusable_segment_list*. For a given signal segment length, the output of this process for each patient is a collection of starting times of unusable segments. To achieve this the function executes the following steps for a given patient:

1. Reads the signal from the patient file.
2. Filters the entire signal using the aforementioned *filter_signal* function.[1]
3. Splits the filtered signal into segments. For each segment we call a *segment_is_usable* function which follows the steps below:
   (a) Preprocesses the segment. Since the entire signal is already filtered, currently this only performs normalisation.
   (b) Computes SQIs for the segment.[2] The SQIs to be calculated as well as a range of acceptable values for each SQI are passed through a dictionary where the key is the SQI and the value is the range.
   (c) If any of the calculated SQIs is outside the allowed range then the segment is deemed unusable and the *segment_is_usable* function returns *False*, otherwise returns *True*.
4. If the *segment_is_usable* function returned *False* then the start time of this segment is added to the output list, appropriately named *unusable_segment_start_times*.

We use this within the pipeline in the following way: The matching algorithm explained in Section 3.2.2 has produced the patient matrix, a list of 4-tuples of the form (*patientID, start_datetime, end_datetime, label*). For a particular patient, this is equivalent to providing a labelled time interval of interest. Rather than quality checking the segment defined by that time interval, we only need to ensure that the segment contains sufficiently few unusable parts by looking in the list of unusable segments for that patient. If the segment contains more than would be acceptable, then that patient is removed from the patient matrix. This process is repeated for all patients. After having potentially removed patients who were unfit from a quality perspective, we refer to the result as the reduced patient matrix.

3.4 Feature Extraction

The aim of the feature extraction stage from a data flow point of view is to be able to transform one row from the patient matrix (*i.e.,* one patient) into a (*feature_map, label*) tuple. In essence, this is achieved by applying a FE algorithm on the signal segment defined by the [*start_datetime, end_datetime*] time interval for that patient. The first method which will be implemented to achieve this is using the raw PPG signal itself. In this case the *feature_map* will be an array-like structure holding the signal values. As a first step we will use this uncomplicated approach in order to establish an end-to-end functioning pipeline for the baseline of this project, or MVP. However, as mentioned in Section 2.3.3.1, although using the raw signal is the simplest approach it is also the most uninformative. Thus, we will then focus on the development of other FE strategies, most likely starting from time-frequency representations, such as STFT and CWT, as these methods already exist in popular Python libraries, meaning that integration with the remaining pipeline is expected to be relatively easy. It is important to note that one of the main aims of this stage is modularity, so as to enable easy comparisons between different approaches through their corresponding results.

3.5 Machine Learning Algorithms

For the starting point of our ML model development we look to the closest study that we have found. This is *Hadjimarkou* 's MSc thesis (2021) [6]. He explored several configurations of ANNs, CNNs, and LSTM RNNs, and subsequently presented the highest performing one for each of the investigated experiments. That said, given the fact that we are working with a new dataset as well as potentially different FE methods and thus data representations, we aim to undertake an exploratory approach as well. Nonetheless, we consider his results to be a solid starting point.

[1] We filter the entire signal instead of individual segments because digital filtering naturally introduces large transients. For the filter discussed in Section 3.3.1 these last around 5 seconds during which the signal is essentially rendered unusable. Therefore, rather than filtering each segment individually and "losing" the first 5 seconds of every 10-30 second segment, we filter the entire signal, compromising only the first 5 seconds.

[2] We note that all SQIs that we have implemented (currently only MSQ and zero-crossing rate due to their simplicity) have been copied from the Imperial-developed software library *vital_sqi*.

3.6 Results Interpretation

The result interpretation stage does not contain components that require explicit planning or implementation. The core task of this block is to decide on the evaluating metrics for the ML model performance. Different alternatives for this task are presented in Section 4.2.

3.7 Timing, Completion Estimates, and Fallbacks

The timing diagram for the project schedule is shown in the form of a Gantt chart in Figure 3.5. We included tasks from the beginning until the end of the project, as well as the project deliverables above the timeline.

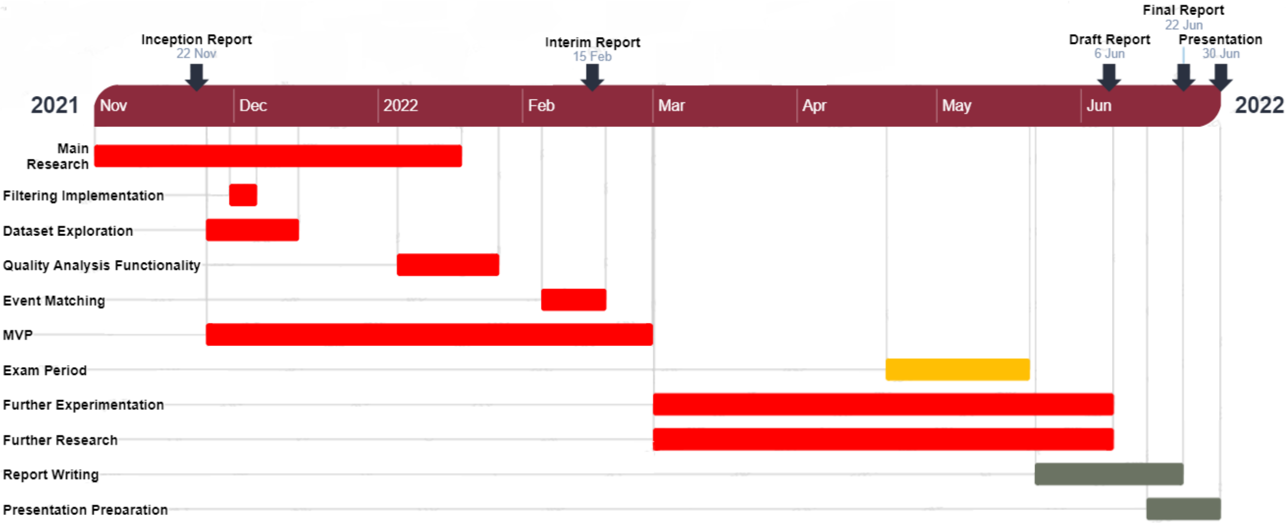


Figure 3.5: Timing diagram for the project schedule, shown in the form of a Gantt chart.

As can be seen in the figure, the two main tasks up until now have been the “Main Research” and the “MVP”. The results of the “Main Research” task have been presented in Chapter 2. We will now provide details regarding the “MVP” task, which refers to the baseline project. The main aim of the baseline project is to establish a functioning pipeline upon which it is easy to build and extend each component. Thus, we aim to adopt the simplest possible approach for all constituent parts, and rather focus on optimising for modularity. Given the above, the baseline consist of the following:

- **Clinical Question:** The choice of clinical question is relatively insignificant for the MVP as we are only concerned with making sure that an arbitrary number of patients is matched with it.
- **Event Matching:** Event matching has to be fully completed for the MVP as it is required for a functional pipeline. For this reason the deadline for it is earlier than the completion of the MVP.
- **Quality Analysis:** While the quality analysis algorithms themselves must be fully functional for the MVP, it is not necessary to have all the SQIs that we wish to explore already implemented. In fact, we will most likely use the SQIs already developed, namely MSQ and zero-crossing rate.
- **Feature Extraction:** As mentioned in Section 3.4, FE does not have to be sophisticated or informative for the MVP, and thus we use the simplest approach, using the raw PPG signal.
- **ML Model:** As in the previous stages, we don’t require an ML model which performs well, and thus we plan to implement a simple ANN or CNN for the MVP with no concern for finetuning.
- **Result Evaluation:** As aforementioned the aim of the MVP is not acquiring results, and so this stage is not truly of interest. That said, aiming for an end-to-end working pipeline, we will again adopt the simplest approach, using accuracy as a metric of the ML model performance.

At the moment, filtering and quality analysis functionality have been completed, with event matching being nearly finished as well. Although, as mentioned, these are part of the MVP, we have also explicitly shown them in the Gantt chart, as can be seen in Figure 3.5. As the project is highly exploratory, the majority of the work after the completion of the MVP is concerned with experimentation. Excluding the deliverables of report writing and presentation preparation, the work has been summarised in two tasks: "Further Research" and "Further Experimentation".

This categorisation was done both to avoid unnecessary overcrowding of the chart, and, more importantly, because of the fact that tasks after the completion of the MVP are highly interdependent. By interdependent we mean that, for example, the quality of results acquired from exploring a particular clinical question with a particular configuration of the pipeline will significantly affect the tasks at the time as well as in the future. In general, decisions on the tasks of the current time will be made as we progress through the project. Finally, regarding fallbacks, because of the way we have structured the project tasks, we can always revert to using the MVP and exploring different clinical questions as preliminary results and thus a baseline project.

Chapter 4

Evaluation Plan

4.1 Evaluation Strategies for Pipeline Blocks

For the evaluation of our project again we turn to the high-level implementation pipeline of Section 1.2.2. In Table 4.1 we list the different blocks of the high-level pipeline as well as the evaluation strategy for each block.

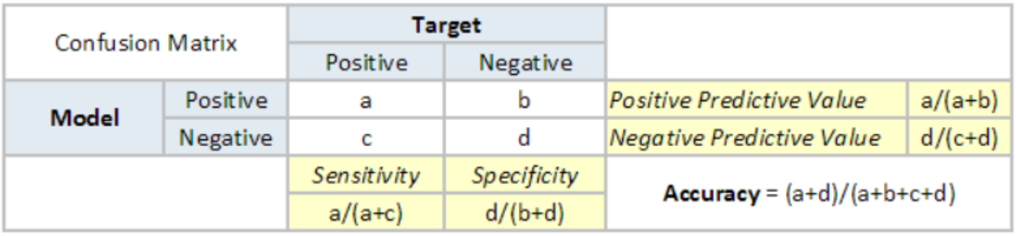
Stage	Evaluation Strategy
Clinical Question	The clinical question can be evaluated based on its clinical significance, feasibility, as well as the number of patients that are matched with it. In reality, the first and the second evaluation criteria are considered when we choose the clinical question and not in retrospect, as explained in Section 3.1. The third evaluation criterion refers to the fact that as the number of patients with sufficient data to support the question increases, so does the training set, which is clearly beneficial for our ML models. This will be determined during the event matching stage.
Event Matching	The event matching stage does not contain any elements that require evaluation.
Quality Analysis	Although it is beyond this project's scope to evaluate the SQIs themselves, we can evaluate the effectiveness of our preprocessing pipeline based on these SQIs. In fact, this is how we can finetune our preprocessing stages, that is, filtering, normalising, and smoothing. For example, if we tune our filters differently and we see that fewer signal segments are rejected by the SQIs, this implies that the new preprocessing pipeline is superior.
Feature Extraction & Model Training	Feature extraction and the ML model class are evaluated together through the model performance. That said, model performance is in fact a product of all previous steps of the pipeline. Thus, we attempt to extract as much information as possible from the performance of the model. Section 4.2 is concerned with the different methods and metrics that we can use in order to evaluate this performance.

Table 4.1: Blocks of the high-level pipeline (left) and evaluation strategy for each (right)

4.2 ML Evaluation Metrics

This section refers to the contents of the "Result Evaluation" block of the high-level pipeline of Section 1.2.2. The choice of metric first and foremost depends on the ML model task, that is, a classification and a regression model would clearly have to use different metrics. Here we focus on the former as our models will most likely only complete classification tasks. Finally, we note that the statistical literature on metrics is extensive and so we will focus on the most commonly used machine learning tools for evaluation.

ML models often use a single number to report model performance: accuracy. However, in a range of applications this does not provide sufficient information for how the model is in fact performing. This is particularly important in medical applications, such as biomedical and clinical scenarios. The concepts of true positive, true negative, false positive, and false negative, as well as others that can be derived from them, are able to provide more information. We introduce these through the confusion matrix as shown in Figure 4.1. After this, Table 4.2 presents the concepts and metrics from the confusion matrix along with a brief description.



Confusion Matrix		**Target**			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Figure 4.1: Confusion matrix along with metrics that can be computed from matrix entries (image from [69])

Concept	**Expression**	**Description**
True Positives (TP)	a	The number of examples that were correctly identified by the model as positive.
False Positives (FP)	b	The number of examples that were incorrectly identified by the model as positive.
True Negatives (TN)	d	The number of examples that were correctly identified by the model as negative.
False Negatives (FN)	c	The number of examples that were incorrectly identified by the model as negative.
Accuracy	$\frac{(a+d)}{(a+b+c+d)}$	The proportion of total predictions that was correctly identified.
Positive Predictive Value (Precision)	$\frac{a}{(a+b)}$	The proportion of actual positive examples from all the examples identified as positive.
Negative Predictive Value	$\frac{d}{(c+d)}$	The proportion of actual positive examples from all the examples identified as positive.
Sensitivity (Recall)	$\frac{a}{(a+c)}$	The proportion of actual positive examples that was identified correctly.
Specificity	$\frac{d}{(b+d)}$	The proportion of actual negative examples that was identified correctly.

Table 4.2: Name (left), mathematical expression (middle), and brief description (right) of the concepts and metrics from the confusion matrix

In the case where we want to maximise precision and recall simultaneously, we introduce the F1 score. The F1 score is defined as the harmonic mean of precision and recall. The choice of harmonic mean over arithmetic mean can be justified as the harmonic mean punishes extreme values. Thus, if a model has either very poor precision or very poor recall the resulting F1 score is low. Further, if we want to assign more importance to recall than to precision we can do so by slightly altering the F1 score equation through the parameter β, resulting in the F-beta score.

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.1}$$

$$F_\beta = \left(1+\beta^2\right) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}} \tag{4.2}$$

The last metric we introduce is the area under the receiver operator characteristic curve (AUC-ROC). The ROC curve is a graphical representation of the performance of a classification model for different values of classification thresholds. The classification thresholds refer to values which we define in order to map the probabilities of logistic classification to a nominal value, that is, one of the classes [70]. The ROC curve depicts performance by plotting sensitivity vs (1-specificity) for different values of the classification threshold. The AUC simply measures the area underneath the ROC curve in the $[0, 1]$ range. Thus, the AUC-ROC metric attempts to provide “an aggregate measure of performance across all possible classification thresholds” [70].

Although we could argue that some metrics have a particular advantage over others, all of the metrics mentioned above have their own drawbacks. In fact, the most common conclusion found in literature regarding the choice of metric for model evaluation is that there is no single metric that is able to encompass the entirety of a model’s performance or which can capture all the desirable properties of a model. For this reason, a collection of several metrics are used to report about a model’s performance [71]. Finally, it is important to note that besides calculating these metrics at the end of training it is very insightful to see how they change as the model trains, as well as the difference in performance between the training and validation sets. These may provide further information, such as whether the model is overfitting.

Chapter 5

Ethics, Legal, and Safety Plan

The majority of ethical, legal, and safety concerns arise from the collection of data from patients. The new dataset was acquired in accordance with the IRB approved clinical study that was conducted at the Hospital for Tropical Diseases (HTD) in Ho Chi Minh City, Vietnam. The study was part of a flagship Wellcome Trust funded project named VITAL. All participants have consented to the study.

Regarding the project itself we argue the following:

- **Ethics:** The only potential ethical concern in this project stems from the use of machine learning models. Models are known to carry over the biases within the input data, such as race, sex, and age, in their outputs. To avoid problems connected to this, we try to consciously remember this fact, as well as never claim that our results are universally applicable.
- **Legal:** All software used, besides the Imperial-developed library *vital-sqi* for which we were given internal access, are free and open source. Additionally, we never share information from the clinical dataset with other parties. Regarding the report, we have accredited parties to the best of our ability.
- **Safety:** This is a 100% software project, and so no safety concerns arise.

Appendix A

Abbreviations

DSS - Dengue Shock Syndrome

PPG - Photoplethysmography
SC - SmartCare monitoring device
GE - General Electric monitoring device

FE - Feature Extraction
STFT - Short Time Fourier Transform
CWT - Continuous Wavelet Transform

ANN - Artificial Neural Network
CNN - Convolutional Neural Network
LSTM - Long Short Term Memory
SVM - Support Vector Machine
RVM - Relevance Vector Machine
RF - Ranfom Forest
DT - Decision Tree
XGBoost - Extreme Gradient Boosting
MLR - Multiple Linear Regression
MnLR - Multinomial Logistic Regression

AUC-ROC - Area Under the Receiver Operator Characteristic Curve

Bibliography

[1] W. H. O. (WHO). *Dengue and Severe Dengue.* URL: `https://www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue`. (last accessed: 28.01.2022).

[2] W. M. Program. *Dengue.* URL: `https://www.worldmosquitoprogram.org/en/learn/mosquito-borne-diseases/dengue`. (last accessed: 28.01.2022).

[3] L. A. Beltz. *Zika and Other Neglected and Emerging Flaviviruses-E-Book: The Continuing Threat to Human Health.* Elsevier Health Sciences, 2021.

[4] J. Allen. "Photoplethysmography and its application in clinical physiological measurement". In: *Physiological measurement* 28.3 (2007), R1.

[5] D. Castaneda *et al.* "A review on wearable photoplethysmography sensors and their potential future applications in health care". In: *International journal of biosensors & bioelectronics* 4.4 (2018), p. 195.

[6] P. A. Hadjimarkou. "Model Development for Understanding the Relationship Between Photoplethysmography (PPG) and Physiological Parameters of Dengue Patients". In: (2021).

[7] SmartCare. *An open pulse oximeter to support healthcare research.* URL: `http://devices.smartcareanalytics.co.uk/`. (last accessed: 31.01.2022).

[8] C. for Disease Control *et al. Areas with Risk of Dengue.* URL: `https://www.cdc.gov/dengue/areaswithrisk/index.html#:~:text=The\%20disease\%20is\%20common\%20in,humid\%20climates\%20and\%20Aedes\%20mosquitoes.`. (last accessed: 30.01.2022).

[9] C. for Disease Control *et al. Dengue Around the World.* URL: `https://www.cdc.gov/dengue/areaswithrisk/around-the-world.html`. (last accessed: 30.01.2022).

[10] Amboss. *Dengue.* URL: `https://next.amboss.com/us/article/350SPg?q=dengue#Z84d1d83616dbfa563e6b5afcd13` (last accessed: 30.01.2022).

[11] L. C. Katzelnick *et al.* "Dengue viruses cluster antigenically but not as discrete serotypes". In: *Science* 349.6254 (2015), pp. 1338–1343.

[12] S. B. Halstead. "Dengue hemorrhagic fever: two infections and antibody dependent enhancement, a brief history and personal memoir". In: *Revista cubana de medicina tropical* 54.3 (2002), pp. 171–179.

[13] L. C. Katzelnick *et al.* "Antibody-dependent enhancement of severe dengue disease in humans". In: *Science* 358.6365 (2017), pp. 929–932.

[14] W. H. O. (WHO). "Dengue vaccine: WHO position paper, September 2018 - Recommendations". In: *Vaccine* 37.35 (Aug. 2019), pp. 4848–4849.

[15] S. Sridhar *et al.* "Effect of dengue serostatus on dengue vaccine safety and efficacy". In: *New England Journal of Medicine* 379.4 (2018), pp. 327–340.

[16] A. Challoner *et al.* "A photoelectric plethysmograph for the measurement of cutaneous blood flow". In: *Physics in Medicine & Biology* 19.3 (1974), p. 317.

[17] T. Tamura *et al.* "Wearable photoplethysmographic sensors—past and present". In: *Electronics* 3.2 (2014), pp. 282–302.

[18] Empatica. *E4 data - BVP expected signal.* URL: `https://support.empatica.com/hc/en-us/articles/360029719792-E4-data-BVP-expected-signal`. (last accessed: 31.01.2022).

[19] J. Přibil *et al.* "Comparative Measurement of the PPG Signal on Different Human Body Positions by Sensors Working in Reflection and Transmission Modes". In: *Engineering Proceedings.* Vol. 2. 1. Multidisciplinary Digital Publishing Institute. 2020, p. 69.

[20] Y. Liang *et al.* “An optimal filter for short photoplethysmogram signals”. In: *Scientific data* 5.1 (2018), pp. 1–12.

[21] J. L. Moraes *et al.* “Advances in photopletysmography signal analysis for biomedical applications”. In: *Sensors* 18.6 (2018), p. 1894.

[22] F. Rundo *et al.* “An advanced bio-inspired photoplethysmography (PPG) and ECG pattern recognition system for medical assessment”. In: *Sensors* 18.2 (2018), p. 405.

[23] J Allen *et al.* “Effects of filtering on multisite photoplethysmography pulse waveform characteristics”. In: *Computers in Cardiology, 2004.* IEEE. 2004, pp. 485–488.

[24] W. Waugh *et al.* “Novel signal noise reduction method through cluster analysis, applied to photoplethysmography”. In: *Computational and mathematical methods in medicine* 2018 (2018).

[25] M. Elgendi. “Optimal signal quality index for photoplethysmogram signals”. In: *Bioengineering* 3.4 (2016), p. 21.

[26] C. Orphanidou. “Quality Assessment for the Photoplethysmogram (PPG)”. In: *Signal Quality Assessment in Physiological Monitoring.* Springer, 2018, pp. 41–63.

[27] Q. Li *et al.* “Dynamic time warping and machine learning for signal quality assessment of pulsatile signals”. In: *Physiological measurement* 33.9 (2012), p. 1491.

[28] S.-H. Liu *et al.* “Classification of photoplethysmographic signal quality with fuzzy neural network for improvement of stroke volume measurement”. In: *Applied Sciences* 10.4 (2020), p. 1476.

[29] S.-H. Liu *et al.* “Classification of photoplethysmographic signal quality with deep convolution neural networks for accurate measurement of cardiac stroke volume”. In: *Applied Sciences* 10.13 (2020), p. 4612.

[30] N. Pradhan *et al.* “Evaluation of the signal quality of wrist-based photoplethysmography”. In: *Physiological Measurement* 40.6 (2019), p. 065008.

[31] M. Elgendi *et al.* “Toward generating more diagnostic features from photoplethysmogram waveforms”. In: *Diseases* 6.1 (2018), p. 20.

[32] M. Elgendi. “On the analysis of fingertip photoplethysmogram signals”. In: *Current cardiology reviews* 8.1 (2012), pp. 14–25.

[33] K. Takazawa *et al.* “Assessment of vasoactive agents and vascular aging by the second derivative of photoplethysmogram waveform”. In: *Hypertension* 32.2 (1998), pp. 365–370.

[34] D. Sarma *et al.* “Dengue Prediction using Machine Learning Algorithms”. In: *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC).* IEEE. 2020, pp. 1–6.

[35] A. K. Chattopadhyay *et al.* “VIRDOCD: A VIRtual DOCtor to predict dengue fatality”. In: *Expert Systems* (2021), e12796.

[36] S. Chattopadhyay *et al.* “Predicting Case Fatality of Dengue Epidemic: Statistical Machine Learning Towards a Virtual Doctor”. In: *Journal of Nanotechnology in Diagnosis and Treatment* 7 (2021), pp. 10–24.

[37] A. Parchani *et al.* “Electrocardiographic Changes in Dengue Fever: A Review of Literature”. In: *International Journal of General Medicine* 14 (2021), p. 5607.

[38] A. Lateef *et al.* “Dengue and relative bradycardia”. In: *Emerging infectious diseases* 13.4 (2007), p. 650.

[39] A. Reiss *et al.* “Deep PPG: large-scale heart rate estimation with convolutional neural networks”. In: *Sensors* 19.14 (2019), p. 3079.

[40] P. S. Addison. “Slope transit time (STT): A pulse transit time proxy requiring only a single signal fiducial point”. In: *IEEE Transactions on Biomedical Engineering* 63.11 (2016), pp. 2441–2444.

[41] S Yang *et al.* “Cuff-less blood pressure measurement using fingertip photoplethysmogram signals and physiological characteristics”. In: *Optics in Health Care and Biomedical Optics VIII.* Vol. 10820. International Society for Optics and Photonics. 2018, p. 1082036.

[42] G. Martínez *et al.* “Can photoplethysmography replace arterial blood pressure in the assessment of blood pressure?” In: *Journal of clinical medicine* 7.10 (2018), p. 316.

[43] K. Pilt *et al.* “New photoplethysmographic signal analysis algorithm for arterial stiffness estimation”. In: *The scientific world journal* 2013 (2013).

[44] R. R. de Almeida *et al.* “Dengue hemorrhagic fever: a state-of-the-art review focused in pulmonary involvement”. In: *Lung* 195.4 (2017), pp. 389–395.

[45] E. Marchiori *et al.* “Pulmonary manifestations of dengue”. In: *Jornal Brasileiro de Pneumologia* 46 (2020).

[46] L. Nilsson *et al.* “Respiration can be monitored by photoplethysmography with high sensitivity and specificity regardless of anaesthesia and ventilatory mode”. In: *Acta anaesthesiologica scandinavica* 49.8 (2005), pp. 1157–1162.

[47] D. Jarchi *et al.* “Validation of instantaneous respiratory rate using reflectance PPG from different body positions”. In: *Sensors* 18.11 (2018), p. 3705.

[48] T. Tamura. “Current progress of photoplethysmography and SPO2 for health monitoring”. In: *Biomedical engineering letters* 9.1 (2019), pp. 21–36.

[49] J. Chaloemwong *et al.* “Useful clinical features and hematological parameters for the diagnosis of dengue infection in patients with acute febrile illness: a retrospective study”. In: *BMC hematology* 18.1 (2018), pp. 1–10.

[50] U. Ralapanawa *et al.* “Value of peripheral blood count for dengue severity prediction”. In: *BMC research notes* 11.1 (2018), pp. 1–6.

[51] A. R. Kavsaoğlu *et al.* “Non-invasive prediction of hemoglobin level using machine learning techniques with the PPG signal’s characteristics features”. In: *Applied Soft Computing* 37 (2015), pp. 983–991.

[52] G. Yoon *et al.* “Multiple diagnosis based on photoplethysmography: Hematocrit, SpO2, pulse, and respiration”. In: *Optics in Health Care and Biomedical optics: Diagnostics and Treatment.* Vol. 4916. SPIE. 2002, pp. 185–188.

[53] M. Azarnoosh *et al.* “Increasing the Accuracy of Blood Hematocrit Measurement by Triplicate Wavelength Photoplethysmography Method”. In: *Jorjani Biomedicine Journal* 6.4 (2018), pp. 19–28.

[54] K. Wirsing. “Time Frequency Analysis of Wavelet and Fourier Transform”. In: *Wavelet Theory.* IntechOpen London, UK, 2020.

[55] A. Jansen. “Modal Identification of Lightweight Pedestrian Bridges based on Time-Frequency Analysis”. PhD thesis. Mar. 2016. DOI: `10.13140/RG.2.2.17116.54407`.

[56] H. Tjahjadi *et al.* “Noninvasive classification of blood pressure based on photoplethysmography signals using bidirectional long short-term memory and time-frequency analysis”. In: *IEEE Access* 8 (2020), pp. 20735–20748.

[57] J. Allen *et al.* “Deep learning-based photoplethysmography classification for peripheral arterial disease detection: a proof-of-concept study”. In: *Physiological Measurement* 42.5 (2021), p. 054002.

[58] H. Wang *et al.* “A high resolution approach to estimating time-frequency spectra and their amplitudes”. In: *Annals of biomedical engineering* 34.2 (2006), pp. 326–338.

[59] K. H. Chon *et al.* “Estimation of respiratory rate from photoplethysmogram data using time–frequency spectral estimation”. In: *IEEE Transactions on Biomedical Engineering* 56.8 (2009), pp. 2054–2063.

[60] Q. Zhao. *Neuron Weights.* URL: `https://qichaozhao.github.io/potato-lemon-2/`. (last accessed: 06.02.2022).

[61] F. Bre *et al.* “Prediction of wind pressure coefficients on building surfaces using artificial neural networks”. In: *Energy and Buildings* 158 (2018), pp. 1429–1441.

[62] A. L. Barroso *et al. Match-Lab Imperial: Deep Learning Course.* URL: `https://github.com/MatchLab-Imperial/deep-learning-course`. (last accessed: 06.02.2022).

[63] K. Simonyan *et al.* “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[64] R. Thakur. *Step by step VGG16 implementation in Keras for beginners.* URL: `https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c`. (last accessed: 07.02.2022).

[65] C. S. Wiki. *Max-pooling / Pooling.* URL: `https://computersciencewiki.org/index.php/Max-pooling_/_Pooling`. (last accessed: 07.02.2022).

[66] S. L. Moulton *et al.* “State-of-the-art monitoring in treatment of dengue shock syndrome: a case series”. In: *Journal of medical case reports* 10.1 (2016), pp. 1–7.

[67] F. Esgalhado *et al.* “The Application of Deep Learning Algorithms for PPG Signal Processing and Classification”. In: *Computers* 10.12 (2021), p. 158.

[68] D. K. Ming *et al.* “Applied machine learning for the risk-stratification and clinical decision support of hospitalised patients with dengue in Vietnam”. In: *PLOS Digital Health* 1.1 (2022), e0000005.

[69] T. Srivastava. *11 Important Model Evaluation Metrics for Machine Learning Everyone should know.* URL: `https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/`. (last accessed: 07.02.2022).

[70] Google. *Machine Learning Crash Course - Classification: ROC Curve and AUC.* URL: `https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc`. (last accessed: 07.02.2022).

[71] S. Hicks *et al.* “On evaluation metrics for medical applications of artificial intelligence”. In: *medRxiv* (2021).