Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

ECG and PPG signal-based real-time blood pressure estimation with deep learning in an embedded environment

Scientific Students' Association Report

Author:

Bálint Tóth

Advisors:

Dr. Luca Szegletes
Dr. Ákos Nagy
Szabolcs Torma

2022

Contents

Kivonat

A vérnyomás a szervezet egészségügyi állapotát jellemző egyik legfontosabb adat. Ezen fiziológia információ alapján számos jelenlegi és jövőbeli betegségre lehet következtetni, amelyek akár nagyon súlyosak is lehetnek. Ezt az a statisztikai adat is alátámasztja, miszerint napjainkban a legtöbb halálesetet a magas vérnyomással kapcsolatos szövődmények okozzák, ilyen például a szívroham vagy a stroke. Tehát a nem megfelelő vérnyomás korai észrevételével akár meg is előzhetőek olyan kóros állapotok, amelyek később súlyos tünetekkel jelentkeznének.

Mindezért rendkívül hasznos lenne egy a mindennapokban használható eszköz, amelynek segítségével valós időben, folyamatosan tudnánk monitorozni ezen élettani jelnek a pontos értékeit. Az előbb említett feltételeket jelenleg csak az invazív vérnyomásmérés teljesíti, bár ezen eljárás csak orvosi felügyelet mellett alkalmazható, így a mindennapokban nem használható. A probléma megoldására jelenlévő non-invazív módszerek közül a leggyakoribb a mandzsettás eljárás, amelynek jelentős hátránya, hogy nem alkalmas folyamatos vérnyomásmérésre, továbbá a mérés eredményét befolyásolhatja a mandzsetta okozta szorító érzés pszichológiai hatása. Kísérletek léteznek azonban külsőleg mért jelek feldolgozásán alapuló eljárásokra is, viszont ezen módszerek teljesítménye még nem kielégítő. Ebben az esetben becslési algoritmusokat szoktak használni, amelyek más, külsőleg mérhető fiziológia jelek ismeretében próbálják meghatározni az aktuális vérnyomásértéket. Ezen jelek közé tartozik a fotopletizmográfiás (PPG) jel, amely lehetőséget nyújt a vértérfogat változásának kimutatására, valamint az elektrokardiogram (EKG), amely a szív elektromos jelenségeit regisztrálja.

A dolgozat célja egy olyan vérnyomásmérő rendszer bemutatása, amely a fent leírt követelményeket képes teljesíteni. Bemenetként EKG és PPG jeleket használok fel, amelyeket egy MAX86150EVSYS hardver egység mér, a mintavételezett értékeket feldolgozza, majd Bluetooth kapcsolaton keresztül továbbítja. Az elküldött adatokat egy Raspberry Pi 4 Model B fogadja, amely a rendszer magját képezi. Itt történik a bemenetek további feldolgozása, valamint neurális hálózat segítségével a vérnyomás aktuális szisztolés, valamint diasztolés értékének becslése. Az ezen két értéket előállító mély tanulást alkalmazó algoritmus több konvolúciós és LSTM réteget tartalmaz, melyek célja az idő- és frekvenciatartománybeli minták megtanulása. A becslés eredményét egy Raspberry-hez csatlakozó monitor jeleníti meg egy grafikus felhasználói interfész segítségével.

A neurális hálózat becslési hibáját először egy előre feldolgozott adatokat tartalmazó, publikusan elérhető adathalmazon értékeltem ki, és hasonlítottam össze irodalmi munkák eredményeivel több osztályozási rendszert felhasználva. Majd az MAX86150EVSYS segítségével saját méréseket végeztem, amelyekre megismételtem a kiértékelési folyamatot. Az eredmények azt mutatják, hogy a bemutatott módszer meghaladja az összehasonítás alapjait képező publikációkban ismertetett eljárások eredményeit. A dolgozatom során bemutatom az elkészített rendszer tervezésének folyamatát, valamint azt, hogy a mély tanuláson alapuló módszerek milyen ígéretes lehetőségeket teremtenek a non-invazív vérnyomásmérés területén.

Abstract

Blood pressure is one of the most vital data characterizing the health status of a human body. Based on this physiological information, many current and future diseases can be detected, which could be very serious. Furthermore, statistics show that most deaths today are caused by hypertension-related complications, such as heart attack or stroke. So, with the early detection of an inadequate blood pressure value, it is even possible to prevent pathological conditions that would later manifest themselves with unpleasant symptoms.

Therefore, it would be extremely useful to have a device that could be used daily, and it would also enable monitoring the exact values of this vital signal continuously, in real-time. These criteria are met only by invasive blood pressure measurement, although this procedure is only allowed under medical supervision, so it is not usable in everyday life. Among the non-invasive solutions, the cuff-based method is the most common. A significant disadvantage of this is that it is not suitable for continuous blood pressure measurement, and the results can also be affected by the psychological effect of the cuff tightening. However, there are also attempts at methods based on the processing of externally measured signals, but the performance of these methods is not yet satisfactory. In this case, estimation algorithms are used to determine the current blood pressure value based on the knowledge of other externally measurable physiological signals. These signs include the photoplethysmogram (PPG), which provides an opportunity to detect changes in blood volume, and the electrocardiogram (ECG), which records the electrical phenomena of the heart.

This study aims to present a blood pressure measuring system that can fulfill the previously-mentioned conditions. As input, I use ECG and PPG signals, measured by a MAX86150EVSYS hardware unit, which collects the sampled values and then transmits them via Bluetooth. The sent information is received by a Raspberry Pi 4 Model B, which is the core unit. Here, the inputs are further processed, and the current systolic and diastolic blood pressure values are estimated using a neural network. The deep learning algorithm that produces these two values contains several convolutional and LSTM layers, which aim to learn patterns in the time and frequency domains. The results are displayed on a monitor connected to the Raspberry using a graphical user interface.

Firstly, I evaluated the estimation error of the neural network on a publicly available dataset containing pre-processed data and compared it with the results of literary works using several grading systems. Then I performed measurements using the MAX86150SYS hardware and repeated the evaluation process on the acquired data. The results show that the presented method exceeds the results of the procedures described in the publications that form the basis of the comparison. During this work, I will present the process of designing the desired system, as well as the promising possibilities that deep learning methods create in the field of non-invasive blood pressure measurement.

Chapter 1

Introduction

Blood pressure, body temperature, pulse, and respiratory rate make up the four primary vital signals. These medical signals indicate the state of the body's life-sustaining functions. [25] The acceptable range of these signals depends on several factors, such as age, gender, weight, and health. In the case of long-term inappropriate blood pressure values, unpleasant illnesses may develop. Two characteristic values of blood pressure are usually examined, namely the systolic and diastolic blood pressure values. The systolic blood pressure (SBP) is the maximum value that occurs when the heart beats, while the diastolic blood pressure (DBP) is the minimum value that can be measured when the heart rests. [26] An inappropriate value of any one of these data can indicate serious problems. The most common disorder is hypertension. High blood pressure can lead to many cardiovascular diseases, like coronary artery disease, heart failure, and atrial fibrillation. These pathological conditions are the leading causes of death worldwide. For example, in 2017, an estimated 10.4 million people died due to elevated systolic blood pressure. [11] In statistics, these figures might be reducible if inappropriate blood pressure values could be indicated before the diseases reach a more serious stage.

For all of this, a daily wearable device that could continuously measure blood pressure in real time would be vital and even life-saving in some cases. However, the accuracy of today's wearable devices is not satisfactory. The invasive method has the highest accuracy, but unusable in everyday life, as medical supervision is mandatory during the process. The most common non-invasive solutions are discontinuous, such as cuff-based measuring. However, in the case of this method, the tight feeling caused by the cuff can have a psychological effect that could negatively influence the measurement. Furthermore, white coat syndrome [74] can similarly distort the results. It can also be very uncomfortable, especially with many consecutive uses. These disadvantages could be avoided with the help of a non-invasive solution that can produce the current SBP and DBP values using external signals.

Neural networks are used to find the non-linear relationship between these physiological signals. These networks process external signals that are easily accessible and measurable. Photoplethysmogram (PPG) and electrocardiogram (ECG) meet these criteria. ECG is a well-measurable signal that records the electrical activity of the heart. [45] Since blood pressure is partly the result of the heart's work, a significant relationship is observable with ECG. PPG is also significantly related to blood pressure, as PPG allows the detection of changes in blood volume, which greatly affects blood pressure. [59] Although both signals are related to blood pressure, it is a common approach to derive SBP and DBP values from PPG because these two waveforms are very similar, whereas the ECG waveform differs

significantly from blood pressure. In the case of a PPG-based solution, in contrary to comfort and portability, the disadvantage is that its accuracy is not adequate. However, some approaches use ECG in addition to the PPG signal to try to achieve better accuracy, but this solution mostly appears only on a research level.

Based on the above-mentioned information, I consider it an exciting challenge and an engaging task to create a system that can sample the input signals and based on these values estimate the SBP and DBP values with a sufficiently small error and display them.

Nowadays, the use of neural networks is becoming more and more widespread in healthcare as well. These solutions can facilitate the work of doctors and create many new opportunities to produce automated solutions to various problems.

A prominent member of fields that utilize neural networks is the processing of EEG (Electroencephalogram) signals, which, e.g., provides an opportunity to detect seizures [32] or to create a brain-computer interface [81]. Furthermore, the field of medical image processing is also dominant, within which several solutions can be found, for example, MRI image-based tumor classification [67] or chest CT image-based COVID-19 identification [66]. Finally, the topic of physiological signal processing, which is also examined in this study, has great significance. This area includes, e.g., ECG classification [101] and blood pressure estimation based on PPG signal [46].

The neural networks implementing the previously-mentioned solutions are often composed of several types of layers. Among these layers, the convolutional layers should be emphasized, which are often used at the beginning of the model for feature extrusion. Furthermore, layers based on RNN (Recurrent neural network) are utilized to process time series data. Solutions based on the attention layer can be considered a new approach, which enables fast and efficient processing of large sequences of data.

During my work, I created a neural network that contains three types of layers, namely convolutional layers, LSTM layers, and fully connected layers, which aim to find patterns in the time and frequency domains in the input signals. The network is trained, validated, and tested on a publicly available dataset. The desired system, which is visualized in Figure 5.1, consists of two chief hardware units. One is the MAX86150EVSYS [2] signal measurement unit with a client application allowing to save measured data, which I utilized for transfer learning. The other hardware is the Raspberry Pi 4 Model B [72], which runs an application responsible for Bluetooth communication with the data acquisition unit, running the neural network, displaying the obtained results on the monitor, and handling events generated by the mouse. In addition, as a supplement to the problem, I also compute the heart rate from the ECG signal.

This work is structured as follows. In Chapter 2, the background of blood pressure measurement and the related signals used in this study are presented. In addition, the neural network knowledge related to my work is detailed. In chapter 3, two articles are presented, which deal with a similar problem as this study. These papers also contain information that helped me get to know the current state of deep learning-based blood pressure measurement better. Chapter 4 shows the pre-processing methods used in the implemented system and the structure of the created networks. After that, in Chapter 5, I introduce the desired system and its components, as well as the application I created. Chapter 6 details the data used, the way the networks were trained, the evaluation methods, as well as the performance of the networks based on them, and the limitations of the final system. Finally, in Chapter 7, I summarize this work and then mention the future development opportunities according to this task.

Chapter 2

Background

Within the framework of this chapter, the information, which is closely related to this study, is presented. The topics include all the signals utilized and intended to be produced, as well as the methods used as the building blocks of implementation.

2.1 Arterial blood pressure

Blood pressure is the pressure difference between the central and surrounding parts of the bloodstream. Furthermore, it can also be defined as the pressure of circulating blood against the walls of blood vessels. Most of the pressure in the vessels results from the pumping of the heart. [6] The importance of pressure is the fact that it makes blood circulation possible, and this blood flow delivers oxygen, nutrients, and regulatory substances to the organs and removes the end products of metabolism. [9]

Therefore, blood pressure is critical for us. This is also why doctors often check this physiological signal because it is easy to infer the health status of the given person from these values.

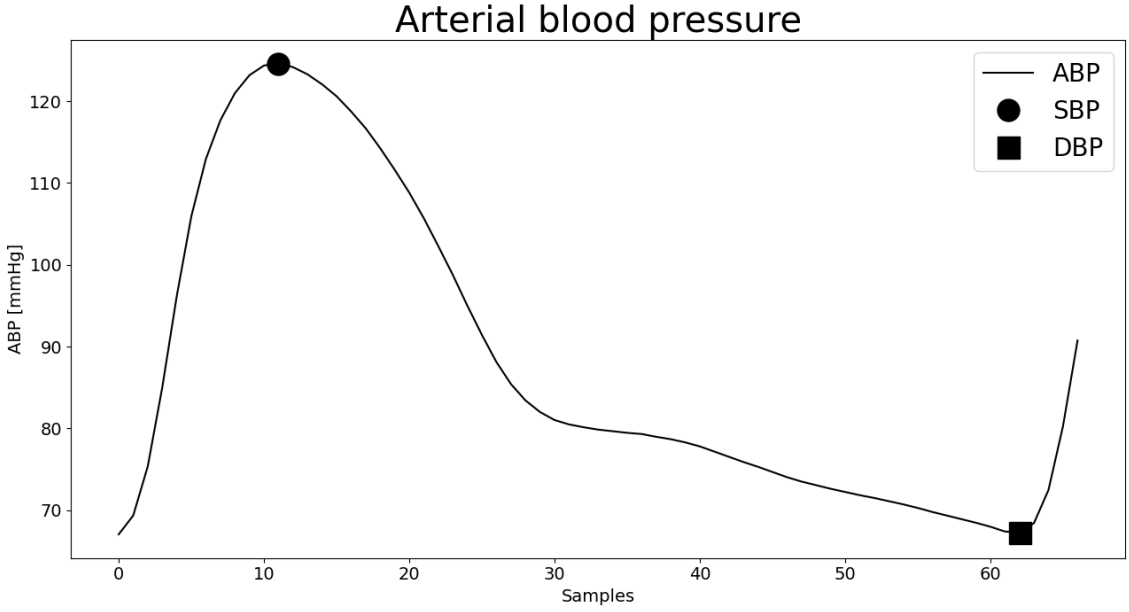

Figure 2.1: Arterial blood pressure waveform

If the exact place in the body is not defined for blood pressure measurement, then we are talking about blood pressure that can be measured in the large arteries, and this value decreases with the division of the arteries into smaller branches. Within the cardiac cycle, which is defined as the interval between two heartbeats, arterial blood pressure (ABP) contains two notable values. The first is systolic blood pressure (SBP), which is the maximum pressure during a cycle, and the second is diastolic blood pressure (DBP), which is the minimum pressure between two heartbeats. These quantities are measured in millimeters of mercury (mmHg) above surrounding atmospheric pressure. Figure 2.1 aims to visualize the above-mentioned values.

By knowing the systolic and diastolic blood pressure, the normal and abnormal levels can be determined. In addition, the blood pressure values differ depending on gender and age, however, I present the evaluation applied to general adults. Since there are many classes, the most common ones are highlighted. Table 2.1 indicates the exact values and limits for the different blood pressure levels in Europe according to the European Society of Cardiology (ESC) and the European Society of Hypertension (ESH) [100].

Category	Systolic (mmHg)	Diastolic(mmHg)
Hypotension	< 90	< 60
Optimal	90–119	60–79
Normal	120–129	80–84
High normal	130–139	85–89
Grade 1 hypertension	140–159	90-99
Grade 2 hypertension	160-179	100-109
Grade 3 hypertension	≥ 180	≥ 110
Isolated systolic hypertension	≥ 140	< 90
Isolated diastolic hypertension	< 140	≥ 90

Table 2.1: Adult blood pressure classification [100][28][39]

Table 2.1 clearly shows that we distinguish several abnormal conditions of arterial blood pressure. If the values are too low, we are talking about hypotension. [28] However, if the blood pressure is too high, we are talking about hypertension. [13] These cases can be pending for varying periods. High blood pressure is much more common than low blood pressure, this is the reason why we divide hypertension into several parts.

The primary symptoms of hypotension are usually lightheadedness and dizziness. Other symptoms include fatigue, shortness of breath, headache, tremors, increased thirst, irregular heartbeat, chest pain, and confusion. Severely low blood pressure can deprive the brain and other vital organs of oxygen and nutrients, leading to a life-threatening condition known as shock. [28]

High blood pressure is especially dangerous because it can be asymptomatic for a long time, which can even mean years. [27] However, long-term high blood pressure is a major risk factor for stroke, heart failure, atrial fibrillation, peripheral artery disease, chronic kidney disease, and dementia. [13] In addition, hypertension is one of the leading causes of premature death worldwide. Furthermore, blood pressure is a dynamically varying parameter, therefore it may happen that intermittent measurements e.g., a medical visit do not provide the full picture, and as a result, hypertensive periods and repeated temporary hypertensive states may remain unrecognized for a long time. For all these reasons, it would be extremely important to constantly measure blood pressure. With the help of this, the problem could be treated in the initial phase of hypertension, avoiding fatal consequences.

In detailing high blood pressure, "high normal" blood pressure is on the border between normal and hypertension, but is not yet considered a high value. It can rarely cause dizziness, facial flushing, and blood spots in the eyes. [29] At this blood pressure level, it would be most optimal to stop the development of high blood pressure. Hypertension is divided into three stages, in which the third is the most critical, and in all the cases, the serious diseases mentioned above can occur. Isolated systolic hypertension occurs when the systolic value is high, while the diastolic value is average. This is more typical for older people. Over time, it can increase the risk of stroke, heart disease, and chronic kidney disease. [88] Isolated diastolic hypertension is defined by a high diastolic blood pressure value with an average systolic value. It can increase the risk of cardiovascular disease, and this case is more typical for young and middle-aged people. [39]

Based on the knowledge mentioned above and the information provided by the elements of Table 2.1, it is clear that measuring systolic and diastolic blood pressure values is extremely important. Accurate knowledge of these values reveals a lot of information about a person's current state of health. In addition, the development of many serious diseases can be prevented.

2.2 Classic measurement methods

There are many ways to measure blood pressure. We distinguish between invasive and non-invasive measurement methods. The choice of the exact measurement method often depends on the person being examined, different methods are used for, e.g., infants, children, the elderly, and pregnant women. In the following I present the classical most frequently used methods. [69]

1. Invasive blood pressure monitoring

   Invasive (intra-arterial) blood pressure (IBP) monitoring is a commonly used technique in the intensive care unit (ICU) and also during surgery. This technique consists of direct measurement of arterial pressure by inserting a cannula into the corresponding artery. The cannula is attached to a sterile, fluid-filled system that is in connection with an electronic monitor. [31]

   This measurement method has several advantages. The most significant one is that we can continuously measure blood pressure in real time. Furthermore, it is the most accurate blood pressure measurement method. [31]

   However, invasive monitoring also has disadvantages. This method requires expertise and is very expensive. It is essential, for example, to keep in mind proper sterilization. Moreover, compared to non-invasive techniques, it can cause serious complications. [90]

2. The oscillometric technique

   During this measurement, the cuff, placed on the upper arm, is inflated and then deflated while the pressure inside the cuff is measured. The rises and falls show small oscillations indicating the pulsatile blood volume in the artery below the cuff. The amplitude of these oscillations varies with the applied cuff pressure. Blood pressure is then estimated from oscillation amplitudes and cuff pressure. [63]

   The strengths of the method is the possibility of blood pressure measurement in case of a weak signal. It does not necessarily require specialist knowledge. Furthermore,

we get rid of the many inconveniences caused by invasive measurements. [70]

However, we can talk about several weaknesses. For example, oscillometry is very sensitive to movements due to the bandwidth of the signals, so the arm must be stationary. Also, the accuracy of systolic and diastolic blood pressure depends on the estimation algorithm used. Furthermore, it cannot be considered comfortable, especially in the long term. Last but not least, one of the disadvantages is that it cannot measure continuously, only at specific intervals.

3. The auscultatory method

   It is also important to mention the auscultation method, as it can be called the gold standard of blood pressure measurement. During the measurement, the bell of the stethoscope must be held above the brachial artery and the blood pressure cuff must be inflated to a level higher than the systolic pressure determined by palpation, then deflate it continuously. After that, record the systolic and diastolic pressure based on the Korotkoff sounds[1]. [8]

   The advantages and disadvantages of this method are very similar to those I mentioned regarding the oscillometric technique, so they are not detailed here.

The methods mentioned above are the most well-known and have been used for centuries[2]. They were developed over time, but the basic principle remained. By learning about these methods, we can gain insights into the basic idea of blood pressure measurement. According to the previously clarified details, we can try to develop new methods that will get rid of the above-mentioned disadvantages while containing as many advantages as possible.

In the following chapters, a new measurement approach is presented, which is intended to combine the advantages of invasive and non-invasive measurement methods.

2.3 Photoplethysmogram

Photoplethysmography (PPG) is an optically acquired signal, which allows the detection of changes in blood volume in the microvascular bed of the tissue. This optical signal is easily measurable and accessible. PPG is often used due to the simplicity and cheapness of the sensor needed to measure it. The construction of the PPG sensor is very simple. It contains an LED and a photodetector. For a smoother understanding, Figure 2.2 illustrates the measurement methods of the PPG sensor.

It is clear from Figure 2.2 that PPG has two modes, transmission, and reflection. In transmission mode, the transmitted light is detected by a photodetector placed in front of the LED. In contrast, in reflection mode, the sensor detects light reflected from tissue, bone, or blood vessels. [91]

The method of LED lighting is also an important part. The choice of light wavelength depends on the application and the designer's priorities. The most frequently used colors

[1]The sounds detected by the stethoscope during auscultatory blood pressure measurement are called Korotkoff sounds. The external pressure made by the cuff causes deformation of the underlying arteries and jetting of the flowing blood. It can be shown both analytically and experimentally that the sounds are caused by the instability of the vessel wall. [37]

[2]The first recorded invasive blood pressure measurement was in the 18th century. The oscillometric technique method was first demonstrated in the 19th century and the auscultatory method at the beginning of the 20th century. [77]

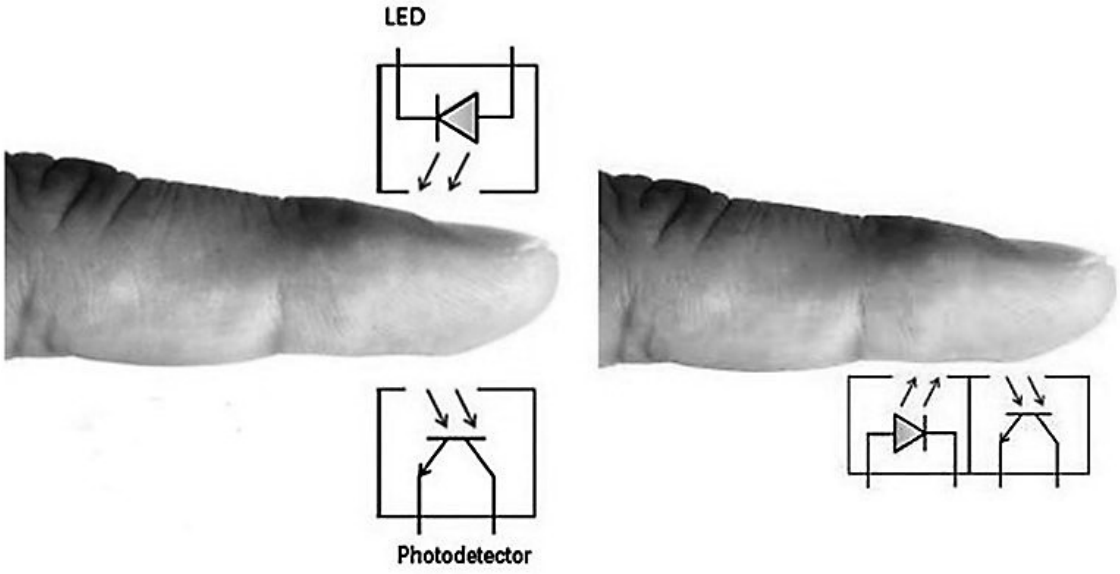

Figure 2.2: Visualization of PPG signal measurement [91]

are red and green. Although, sometimes yellow LEDs are also used. Longer wavelength light penetrates deeper into the tissue, however, Denisse Castaneda et al. stated that infrared light is more susceptible to motion artifacts[3]. [36]

As Figure 2.2 demonstrates, the measurement is usually done on the fingertip, although, this signal can also be measured in other places. It can be detected, for example, on the forehead, earlobe, wrist, torso, or ankle, but the most common place is the fingertip. We can find PPG sensors in our everyday life, e.g., smartwatches (on the wrist) or finger pulse oximeters (on the fingertip). [36]

Lower and higher frequency components of the PPG can be separated based on their sources. The lower frequencies are caused by breathing, sympathetic nervous system activity, and thermoregulation. The higher frequency components represent changes in blood volume caused by heart action, which is dependent on the systolic and diastolic phases. So we can talk about two phases of the signal. The systolic phase begins with a valley and ends with the systolic peak of the pulse wave. The end of the pulse wave is marked by another valley at the end of the diastolic phase. Characteristics such as rise time, amplitude, and shape can indicate vascular changes in blood flow in advance. [36]

PPG shows a great similarity with the blood pressure signal, which is illustrated in Figure 2.3. This characteristic is crucial for the presented study. The blue line represents the PPG and the red represents the aortic (aorta is an artery) blood pressure signal.

2.4 Electrocardiogram

Electrocardiography is a non-invasive diagnostic procedure resulting in an electrocardiogram, which contains the electrical signals of the heart, recorded with the help of electrodes placed on the skin. These electrodes detect the small electrical activities during each car-

[3]Motion artifacts are one of the most common disturbances in signal processing. The most significant phenomenon that creates movement artifacts is the change in the position of the electrodes relative to the skin. Therefore, significant electrical signal distortion occurs. So, during our everyday activities, motion artifacts are created. [49]

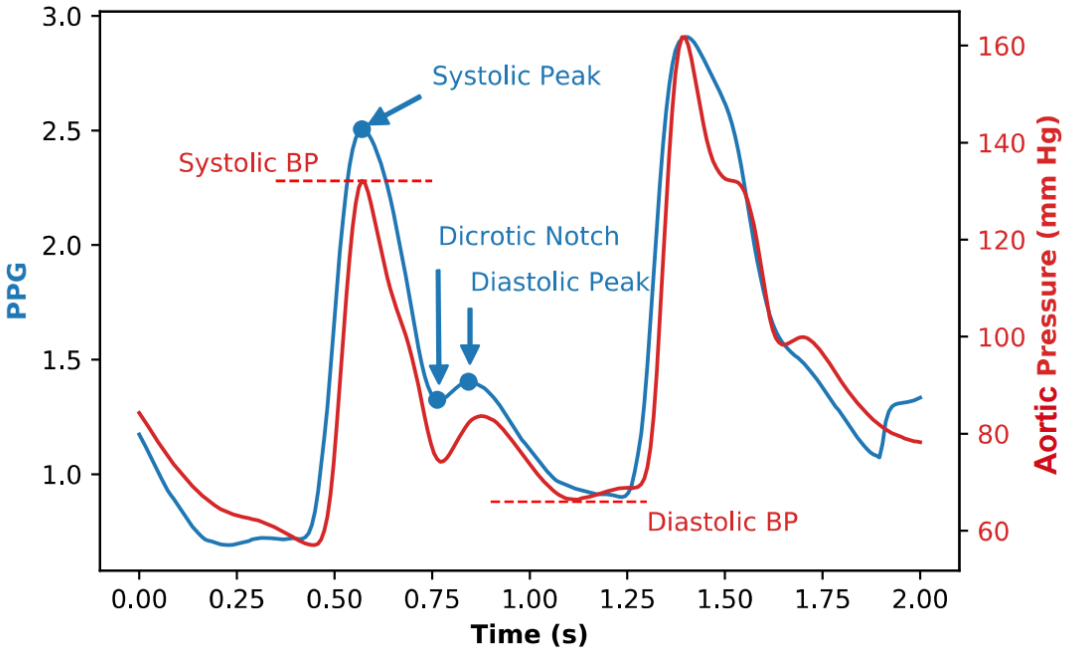

Figure 2.3: Comparison of PPG and blood pressure signals [92]

diac cycle, which is the result of the depolarization and followed by the repolarization of the heart muscle. [45]

The effect of many heart diseases is to change the signal from the original ECG pattern. Examples of such diseases are atrial fibrillation [65] and myocardial ischemia [34]. Therefore, when examining the ECG signal, stress testing is recommended, because some abnormal phenomena cannot be detected at rest, but their presences are recognizable under stress. It can be observed from the previously-mentioned information, that the continuous measurement of this physiological signal is essential since the health state of a person can be inferred from it.

There are several ways to measure an ECG. The conventional measurement method is the 12-lead ECG, which is recorded in a supine position. During this procedure ten electrodes are placed on the patient's limbs and the surface of the chest. The total magnitude of the heart's electrical potential is then measured at twelve different angles. In this way, the overall magnitude and direction of electrical depolarization of the heart are recorded at each instant of the cardiac cycle. Although this method provides the most information, other methods can also be used to measure ECG. In terms of this work, the single-lead method is the most important, because these means of measurement is most commonly used by wearable health devices such as smartwatches. Although this method provides less information than its counterpart with more measurement points, it is suitable for detecting many diseases, such as atrial fibrillation. Furthermore, it is much more comfortable than the others and makes long-term measurements possible. Perhaps the most ergonomic solution for this is to place the two electrodes on one fingertip of each of our two index fingers.

It is worth examining the shape of the ECG signal in detail. Several waves appear in the signal. Each represents the depolarization or repolarization of a specific part of the heart. These waves can be observed in Figure 2.4.

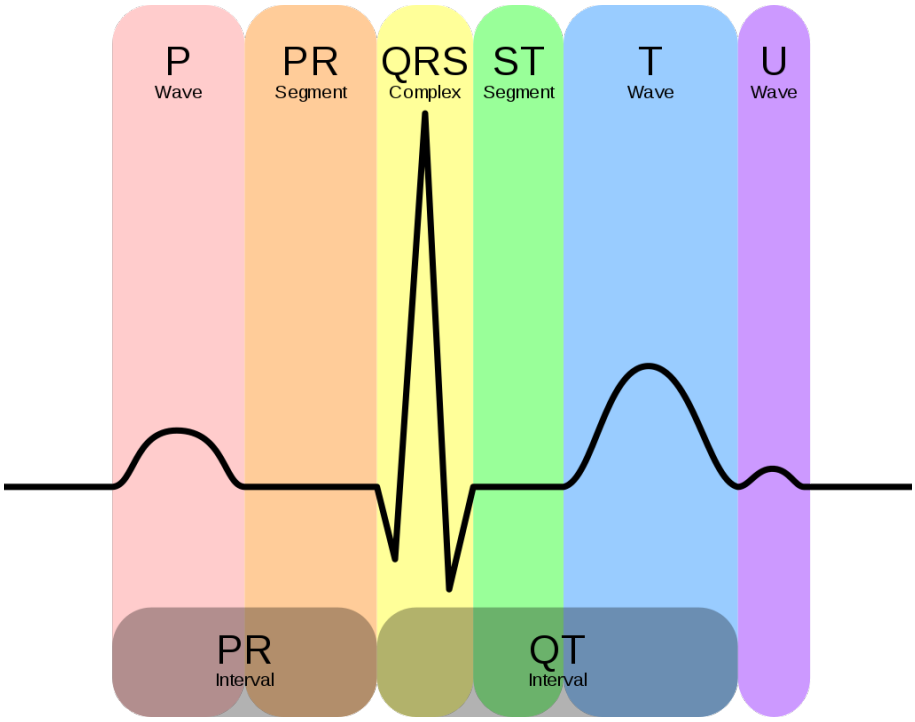

Figure 2.4: ECG waves [99]

Interpreting Figure 2.4, the P wave represents atrial depolarization, while the cause of the QRS complex is ventricular depolarization. In contrast, the T wave indicates ventricular repolarization. And the U wave represents the repolarization of the papillary muscle.

Based on the above, the ECG contains a large amount of information about the condition and functioning of the heart, from which we can conclude that it is closely related to blood pressure. However, it does not contain enough information to make an accurate estimation only by itself. Nevertheless, by supplementing ECG with the PPG signal, we can obtain sufficiently more information. We can consider this approach as the improvement of the photoplethysmography-based method that only uses the PPG signal for the estimation.

2.4.1 Heart rate

Heart rate is an outstandingly crucial physiological value. Together with blood pressure, it is part of the four primary vital signs. [25] It is also known as pulse rate. This physiological information shows the frequency of the heart's contractions per minute, which is usually indicated in the unit of measure, beats per minute (bpm). Many factors can influence this value, including, e.g., age, stress, hormonal status, and the presence of diseases. [103] Therefore, the knowledge of these values can significantly facilitate the establishment of a proper diagnosis for a given subject.

Like most data in physiology, it has healthy and unhealthy ranges. According to the American Heart Association, the usual resting adult human heart rate is 60 to 100 bpm. However, these limits are not as sharp as in the case of blood pressure since this value for athletes can be even 30 bpm. If the individual's heart rate is higher than this during rest, we speak of tachycardia. If it is below this range, we speak of bradycardia. It is also worth mentioning that 40-50 bpm is healthy during sleep.

Heart rate is measured by listening to or palpating the heartbeat. A measurement point can be found in any part of the body. If we want to determine this value accurately, we need to use an electrocardiogram. Based on the ECG, the instantaneous heart rate is calculated based on the R-wave to R-wave interval, which can be observed in Figure 2.5. In the following section, a method based on the ECG signal that implements the calculation of this value is described.

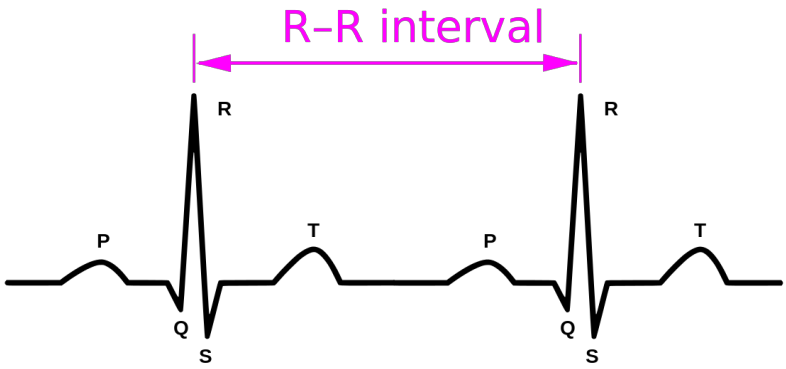

Figure 2.5: R wave-to-R wave interval in ECG waveform [98]

2.4.2 Pan–Tompkins algorithm

During my work, I measured the heart rate using an ECG signal. This task requires the accurate detection of R peaks. I used the Pan–Tompkins algorithm [71] to solve this problem.

The algorithm was proposed by *Jiapu Pan* and *Willis J. Tompkins* in [71], in 1985. The Pan–Tompkins algorithm detects the QRS complexes, presented in the previous section, in electrocardiographic signals. The procedure uses a series of filters to emphasize the frequencies related to rapid cardiac depolarization and removes noise. Then after applying these filters, it detects peaks using threshold values. [71] Figure 2.6 shows the series of filters.

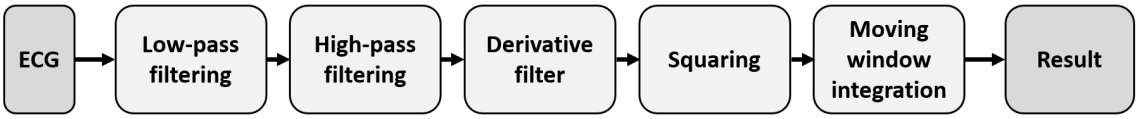

Figure 2.6: Series of filters in Pan–Tompkins algorithm

The steps of the algorithm are detailed below:

1. As a first step, to increase the signal-to-noise ratio, it is recommended to apply a band-pass filter (this is the combination of the low-pass and the high-pass filter from the block diagram above). This filter aims to highlight the desired peaks and suppress muscle noise and other disturbances.

2. As the next step, a derivative filter is applied, which emphasizes the parts of the signal where faster changes occur, thereby highlighting the steep rise and fall of the QRS.

3. The signal obtained as the output of the previous step is then squared to highlight the already outstanding R peak, thus reducing the possibility of a T-wave being detected as an R peak.

4. Finally, a moving average filter is applied to provide information on the duration of the QRS complex, and also functions as a low-pass filter on the resulting waveform, which may contain several outliers due to derivation and squaring.

Even though the PPG waveform differs significantly from the ECG waveform, the algorithm can also be used for PPG if its parameters are modified slightly. This is possible because the steepest part of the PPG signal is just before the systolic peak, just like around the R peak in ECG.

2.5 Neural networks

Neural networks utilize and imitate many ideas we learned from the signal transmission between biological neurons. Artificial neural networks are graph-based models in which artificial neurons communicate with each other. The neurons form the vertices, and the connections represent the edges.

In the simplest case, the models can be divided into three main parts. The first is the input layer, which forwards the input data to the rest of the network. The second part consists of hidden layers, whose task is to transform and transcode the data and create intermediate representations. The last layer is the output layer, the implementation differs depending on whether we are talking about a classification problem or a regression problem. In the case of classification, we produce the same amount of output as the number of classes. The values obtained for each output node contain information characterizing the probability of belonging to the given class, from which it can be determined which class the input belongs. However, in the case of regression problems, a continuous value is predicted based on the input variables, which will serve as the output. So the principal goal of these problems is to estimate a function utilizing inputs and outputs. [54]

Figure 2.7 visualizes the above-mentioned information with a simple example.

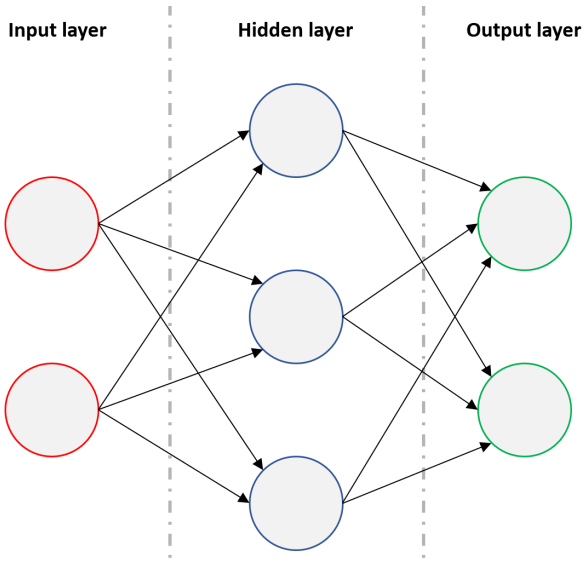

Figure 2.7: Artificial neural network

Neural networks are becoming more and more widespread in all areas of life. The use of these models is preferred for many problems that cannot be solved by other methods or would take a very long time. Examples of such tasks are signal processing([52][80][105]), image processing([44][64][62]), or natural language processing([50][89][40]). In the following, models are presented that are important in terms of the current work.

2.5.1 Deep learning

Deep learning models are neural networks that contain several hidden layers. The task of these layers between the output and input is to learn important patterns and details. As a result, deep learning networks do not require manual feature engineering as most machine learning applications do, because they are able to learn the important characteristics of the input data. [35]

The algorithms created during my work use architectures based on deep learning, namely recurrent neural networks, convolutional neural networks, and a part of Transformers called as attention layer. These models are preferred for signal processing problems, so analyzing and using them was reasonable. In the following, these are presented in more detail.

2.5.2 Fully-connected layer

Concerning deep learning, we often come across a so-called fully-connected neural network (FCNN) [94]. They are used as the last layer in all of the networks presented in this work since they produce the output with the correct dimensions. Concerning the FFCN, the structure and function of a neuron are presented, however, it works identically in other types of neural networks. The neurons generate the dot product of the weights(w) and the inputs(x) and then add a bias. The value obtained as a result of completing these operations is entered into a non-linear activation function[4] and this will give the output(y) of the neuron. The above is illustrated in Figure 2.8.

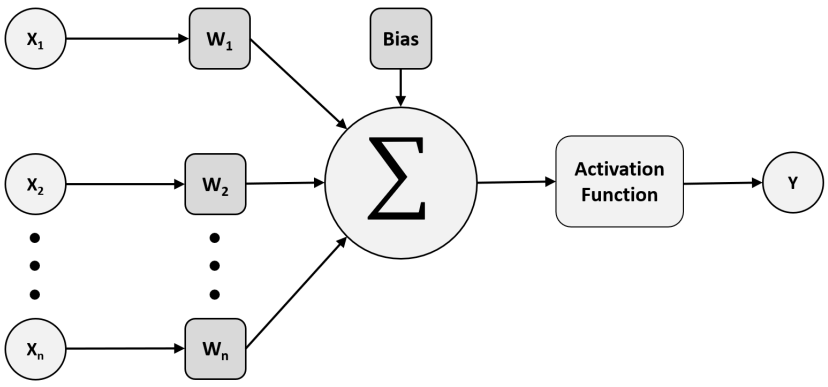

Figure 2.8: Operations done by a neuron in FCNN

Figure 2.8 can also be written in the form of equation, which is shown by Equation 2.1, where φ denotes the activation function and b denotes the bias.

[4]In artificial neural networks, the activation function of a graph vertex gives the output for a given input. Non-linear activation is called non-linearity because they allow networks to learn not just linear relations. [87]

$$y = \varphi(\sum_{k=1}^{n} x_k \cdot w_k + b) \tag{2.1}$$

The advantage of this neural network is that it is simple and fast, as it is based on the calculation of linear combinations. The disadvantage is that, due to its network topology, it is expensive in terms of the number of weights. It is also significantly sensitive to vanishing[5] and exploding gradient problems[6].

2.5.3 Convolutional neural network

A prominent type of artificial neural network is the convolutional neural network, which is based on the cross-correlation operation. These convolutional layers contain filters (or kernels) with weights that slide along the input features and produce feature maps as outputs corresponding to the filters. This type of network is used with preference in image processing, recommender systems, natural language processing, physiological signal processing, financial data processing, and other type of time series processing. This network learns the optimal values of the filters in an automated way. The advantage of a convolutional network is that it can search for local patterns in the input signal and allows learning much more complex ones by combining them. Furthermore, the use of kernels make the network independent of local pattern positions. Also, since not every data point is associated with a neuron, but examines the data with kernels, this is a much more efficient method, which many articles call weight sharing. [83]

Convolutional layer: The convolutional layer can be created with many different properties. I present the most important of these parameters below. [75]

1. Kernel size: The size of the kernel determines the number of inputs that participate in the same filtering step.
2. Input channels: This determines the number of input channels.
3. Output channels: Defines the number of feature maps to be created.
4. Padding: Defines the method that determines the way the filters handle the edges of the input samples.
5. Stride: Defines the step size when a filter passes through the input samples.
6. Dilation rate: This defines a specific spacing between kernel elements.

Furthermore, it is important to mention the equation of the convolution layer, which is shown by Equation 2.2, where N is the batch size[7], C is the number of channels, L is the

[5]We encounter the vanishing gradient problem when we train neural networks with backpropagation. The problem is that in some cases the gradient will be too small in the course of the gradients' backward flow, which significantly prevents the values of the weight to be changed. In the worst case, learning can stop completely. [96]

[6]We encounter the exploding gradient problem when neural networks are trained by backpropagation. This phenomenon occurs when the error gradients are large and cause enormous updates of the neural network model weights during training. Proper training, on the other hand, would be effective if these updates were small and controlled. When the magnitude of the gradients increases, an unstable network is likely to occur, which in the worst case results in a model that fails to do its task. [38]

[7]Batch size is the number of samples processed simultaneously while training the model.

length of the sequence, b is the bias, w is weights, x is the input, y is the output, and $*$ is the cross-correlation operator. Assuming a 1-dimensional convolution, the size of the input is (N, C_{in}, L), and the size of the output is (N, C_{out}, L_{out}).

$$y(N_i, C_{out,j}) = b(C_{out,j}) + \sum_{k=1}^{C_{in}-1} w(C_{out,j}, k) * x(N_i, k) \tag{2.2}$$

Batch normalization layer: Batch normalization layers are often called directly after convolution layers. This layer implements a method, called Batch normalization, which enables faster and more stable training of neural networks by normalizing (re-centering and re-scaling) its inputs. [53] The operation of this layer is shown by Equation 2.3, where x denotes the input, y denotes the output, μ is the mean of the input data, σ is the standard deviation of the input data, so σ^2 is the variance of the input data, ε is 10^{-5} by default, and γ and β are learnable parameters.

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \cdot \gamma + \beta \tag{2.3}$$

Furthermore, this layer not only enables effective training but can be used to avoid overfitting[8] the model. This method was proposed by Sergey Ioffe and Christian Szegedy[9] in 2015 [53].

Activation function: After normalization, the activation functions are used, which are most often implemented with ReLU layers in the case of convolutional networks. The ReLU means rectified linear unit, which uses the following non-linear activation function: $f(x) = max(0, x)$. It is clear from the formula that it keeps the positive inputs, while the negative ones are set to zero. So, with the help of this layer, we inject non-linearity into the network, which makes it able to learn non-linear relationships between the input and the output. In addition, in the past, sigmoid [87] was used instead of ReLU as an activation function whose formula is shown in Equation 2.4.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

However, the most recently proposed functions are Sigmoid-weighted Linear Unit (SiLU) and Swish. The operation of SiLU is described by the following equation: $f(x) = x \cdot \sigma(x)$, while Swish, which complements SiLU with parameter β, is described by this equation: $f(x) = x \cdot \sigma(\beta \cdot x)$. [78]

Pooling layer: After the ReLU, pooling layers are sometimes used. This performs non-linear downsampling of the input data. This layer reduces the spatial size of the representation to reduce the number of parameters and helps to avoid overfitting. The most common implementation of this layer is max pooling and average pooling, during

[8] One of the goals of deep learning models is to get adequate generalization ability by learning the training data. However, there are occasions when the model tries to learn too many details from the training data as well as the noise. As a result, the performance of the model will not be satisfactory for unseen data. This phenomenon is called overfitting. [48]

[9] Christian Szegedy completed his high school studies in Budapest and obtained his first degree at Eötvös Loránd University in the field of Mathematics and Computer Science.

which the pools containing the input data of a given size are replaced with the maximum or the average of the values contained in them. [79]

Dropout layer: Finally, the Dropout layer is occasionally used, which is a mask that cancels the contribution of some neurons to the next layer. Dropout layers are of particular importance when training convolutional layers because they prevent the phenomenon of overfitting and help each neuron to learn adequately at the proper rate. [33]

So, the advantage of convolutional layers is that they can be applied to several types of problems, using their skill of learning the important features in the input data without any human supervision. However, its disadvantage is that it does not encode the location of the objects, it needs a large amount of training data, and since it does not contain memory elements, therefore, it cannot model the temporality of time series signals by itself, which is often indispensable information.

The convolutional layers play a significant role in the structure of my desired neural network, for which I used all the above-mentioned layers to achieve proper functioning during the preparation.

2.5.4 Recurrent neural network

Recurrent Neural Network (RNN) creates a method for efficient time series processing. This network has internal memory elements to monitor temporal dependencies, so it has internal states. [42] During the operation of the RNN, the current value of its internal state is calculated based on the current input and the value of the internal state one step earlier. Furthermore, the output of the cell is equal to the current internal state. Such a cell is shown in the Figure 2.9, which visualizes that an RNN cell contains three linear layers and an activation function, which is most often a hyperbolic tangent activation function (tanh) [102], whose operation is described by Equation 2.5:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.5}$$

In Figure 2.9, h represents the internal state, x represents the input, y represents the output, and t represents the time index.

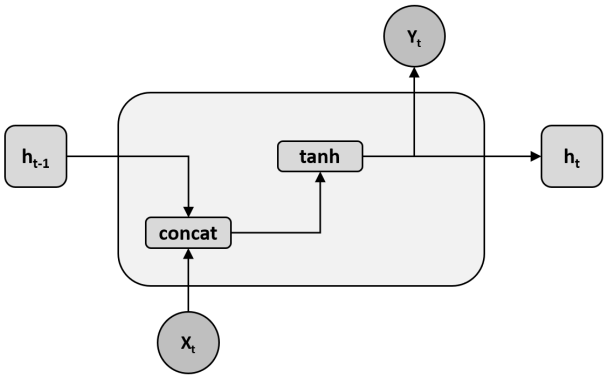

Figure 2.9: Cell of Recurrent Neural Network

This type of neural network offers many possibilities because it has internal states. However, it also has several disadvantages. For example, its training process is outstandingly

slow and is significantly affected by vanishing and exploding gradient problems. In the following, I will present the developments of RNN that eliminate these negatives. I investigated both improved versions (LSTM and GRU) during my work.

2.5.4.1 LSTM

In the case of RNN, vanishing and exploding gradient problems can often occur. The probability of occurrence of this issue is reduced by LSTM, i.e. Long short-term memory, which contains feedback connections that allow them to process entire data sets because they have access to necessary information about previous data. As a result, LSTMs are especially useful in processing data sequences such as text, speech, signals, and general time series. [41]

This special RNN layer uses a series of gates. The task of the gates is to control how data enters the cell, how it is stored in the network, and how it leaves the cell. So the three gates are the forget gate, the input gate, and the output gate, however, the input gate is often divided into input gate and cell gate. The structure of the LSTM cell is shown in Figure 2.10, where the cell is separated according to the three main gates, and f, g, i, and o denote the more detailed version of the gates, which respectively mean forget, cell, input, and output gate. Furthermore, c is the cells state, h is the hidden state, x is the input, y is the output, and t is the time index.

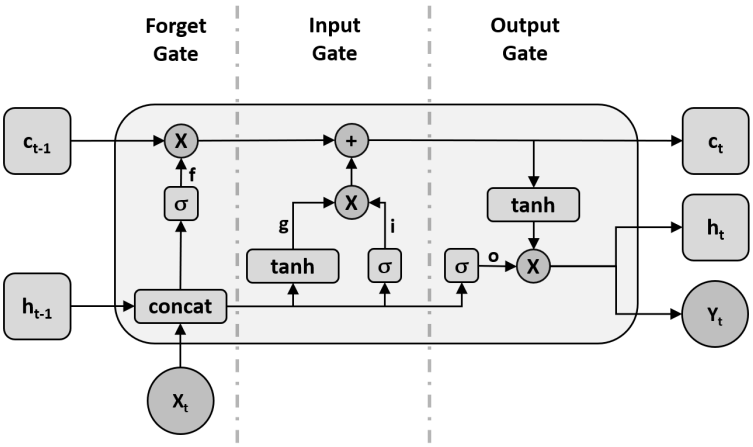

Figure 2.10: Cell of Long short-term memory

Two types of activation functions can be observed in the figure. One is the hyperbolic tangent and the other is the sigmoid function. The latter function converts the input value to an interval between 0 and 1, while the tangent hyperbolic maps it between -1 and 1.

Figure 2.10 can also be written in the form of equations, which is shown by Equation 2.6, in which the marks correspond to those seen in Figure 2.10, $*$ denotes the element-wise multiplication, t is the time index, W denotes the weights, and b denotes the biases corresponding to the given gate.

$$
\begin{aligned}
f_t &= \sigma(W_{xf} \cdot x_t + b_{xf} + W_{hf} \cdot h_{t-1} + b_{hf}) \\
g_t &= tanh(W_{xg} \cdot x_t + b_{xg} + W_{hg} \cdot h_{t-1} + b_{hg}) \\
i_t &= \sigma(W_{xi} \cdot x_t + b_{xi} + W_{hi} \cdot h_{t-1} + b_{hi}) \\
o_t &= \sigma(W_{xo} \cdot x_t + b_{xo} + W_{ho} \cdot h_{t-1} + b_{ho}) \\
c_t &= c_{t-1} * f_t + i * g_t \\
h_t &= o_t * tanh(c_t)
\end{aligned}
\tag{2.6}
$$

The tasks of the three main gates, shown in Figure 2.10, are different. The forget gate decides which parts of the long-term memory should be forgotten at a given time, using the current input and the hidden state of the previous cell. The next gate is the input gate, which aims to determine what new information should be added to the cell state, which is the long-term memory of the network. This gate utilizes the previous hidden state and the new input data. The last unit is the output gate, which decides the new hidden state, which also serves as the output of the given cell. The newly updated cell state, the previous hidden state, and the new input data are used to solve this task.

As I mentioned earlier, LSTM has the advantage of reducing the probability of vanishing and exploding gradient problems, however, its learning speed is even slower than RNN, which is its main disadvantage.

2.5.4.2 GRU

GRU stands for gated recurrent units. Like LSTM, it is constructed by gates. Compared to RNN and LSTM, it has a significant benefit in terms of the speed of the training process. The GRU is similar to the LSTM, however, it does not have an output gate, and therefore has fewer parameters. GRU approaches the performance of LSTM in many tasks, such as polyphonic music modeling, speech signal modeling, and natural language processing. Furthermore, on smaller datasets, GRU outperforms LSTM, which performs better on longer ones. [57]

2.5.5 Attention layer

The attention layer became widely known with the spread of transformers [95], which were mostly invented for natural language processing tasks, but can also be used to process any type of sequence. For me, the encoder part of the transformer is important, which can be observed in Figure 2.11.

According to Figue 2.11, it can be observed that the input sequence goes through the input embedding, which pre-processes the given data to create a hidden representation of it. The next step is positional encoding, which records the meaning and position of the individual elements of its input.

The resulting data is directed into vectors called "query", "key", and "value", which form the inputs of the upcoming layer. The next layer is the multi-head self-attention layer, which has the previously-mentioned three inputs having the same values. This layer is the most important among the other elements that make up the neural network. The attention module performs parallel calculations. Elements running in parallel are called attention heads. The module divides the three input vectors according to the number of attention heads and then passes each part through its corresponding parallel element. Then the result acquired in parallel are merged to generate the Attenuation score. This

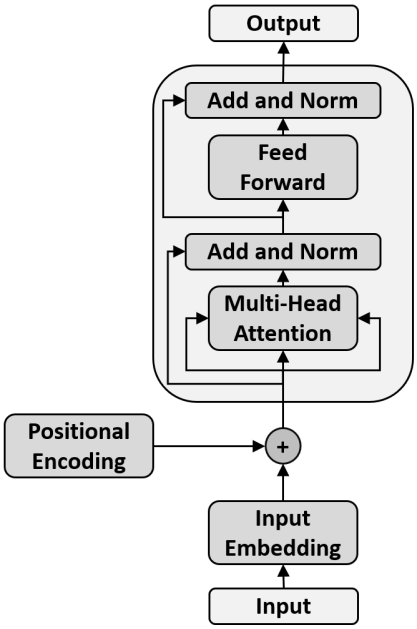

Figure 2.11: Neural network based on attention layer

method allows the detection of several types of relationships in the input. The attention layer is followed by a fully-connected feed-forward network applied position-wise.

Residual connections[10] are performed around the attention and the feed-forward layer. The output of these residual connections goes to ReLUs, and the output of the last ReLU layer forms the result of this neural network architecture.

2.5.6 Training

A neural network model can be written in the form of a parametric function $y = f(x, \vartheta)$, in which y is the output, x is the input, and ϑ are the parameters. The training process aims to find a parameter set of this function that provides the smallest possible difference between the output given by the function and the expected output. In the following, essential elements of this training process are presented.

Loss function During the training, loss functions are used, which is also called cost function or error function or objective function. This method quantifies how well our algorithm can model the desired function. [93]

Optimization Optimization aims to minimize the previously-mentioned loss function. Gradient-based methods can be used for optimization, of which I present the relatively simple and the improved versions as well.

In the case of gradient-based optimization, we can take advantage of the fact that both the model and the loss function can be derived. Based on this, we can calculate the gradient of the error function according to each weight. If we turn these gradients in the

[10]The residual connection provides an alternative path for the flow of data and gradients, during which the information flowing on this branch bypasses a few layers. This helps to avoid the difficulties arising from the problem of gradient flow that often occurs in the case of deep neural networks. Furthermore, networks with residual connections converge faster than networks without such layers. [51]

opposite direction, we get the direction of the steepest decrease. Since this method is slow, we divide the training dataset into equal-sized, randomly selected subsets(minibatches), and after each minibatch, an optimization step is performed. The previously-mentioned method is called Stochastic Gradient Descent. The formula of this method is shown in Equation 2.7, where E denotes the error, W the weights, and α the learning rate, which is a hyperparameter[11].

$$W_{k+1} = W_k - \alpha \cdot \frac{\partial ||E||^2}{\partial W} \tag{2.7}$$

This method can be further developed by adding a kind of momentum to the gradient method. According to this method, the step in a given time is the weighted average of the step taken in the direction of the negative gradient and the step taken one step earlier. This development of the gradient-based method can accelerate the optimization process and helps to avoid getting stuck in local minima.

As an improvement of the previously-mentioned method, higher derivatives can be used, which make the process faster and also helps to avoid oscillating near the optimum. During this method, the cost function is approximated by a quadratic Taylor polynomial, whose minimum point can be calculated analytically. However, since the function is not quadratic, we apply this procedure iteratively. The resulting method is shown in Equation 2.8, where H_w is the Hesse matrix containing second-order partial derivatives, and W contains the weights.

$$W_{k+1} = W_k - H_w^{-1} \nabla_w f(W_k) \tag{2.8}$$

Backpropagation It is important to mention backpropagation, as it creates the basis for training neural networks. This term strictly refers only to the algorithm for computing the gradient, on the other hand, this term is often used to refer to the entire learning algorithm. During the adjustment of the neural network's weights, backpropagation calculates the gradient of the loss function regarding the network weights for a single input-output example. This procedure is very efficient and therefore provides an opportunity to train multi-layer networks. This algorithm utilizes the chain rule [84] during the gradient flow. The gradient is calculated layer by layer, iterating from back to front, during which process the gradient of the loss function is calculated for each weight. [58]

[11] Hyperparameters are the neural network parameters whose values are not learned during training but can be adjusted manually. An example of this is the kernel size in convolutional layers.

Chapter 3

Related work

Continuous measurement of blood pressure is a great task these days, because it would allow us to detect abnormal blood pressure quickly and make early treatment. This would provide an opportunity to prevent serious diseases associated with inadequate blood pressure. The best way to measure real-time values would be with a device that could be worn the whole day and utilizes externally measurable signals to determine blood pressure values.

The works in the literature presented below show possible solutions to this problem. These articles describe blood pressure estimation algorithms using different input signals and methods. In addition to their procedures, they present several evaluation methods and the results obtained based on them. A large amount of information about the current state-of-the-art solution to this problem, found in these studies, has a great impact on the current work.

3.1 Approach based on PPG

Due to the similarity between PPG and ABP signals, some researchers have attempted to create methods that continuously estimate systolic and diastolic blood pressure based on photoplethysmography.

Approach For example, *Ali Tazarv* and *Marco Levorato* [92] attempted such a solution at the University of California. In their article, they present an approach based on deep learning. The neural network they developed has one input, which is an 8 second time window with a step of 2 second that contains the PPG signal. Its outputs are the systolic and diastolic blood pressure values for the given time frame. The network was trained with the maximum and minimum values of the given time window, which correspond to the systolic and diastolic values.

Model The first block of their deep learning model contains a CNN layer. The CNN layer starts with a 1-D filter with filter size 15, followed by a Rectified Linear Unit (RELU) as an activation function, a batch normalization layer, a max-pooling layer, and finally a dropout layer with a dropout rate = 0.1. In the max-pooling layer, the pooling size is set to 4. The purpose of this block is to find informative features efficiently in the PPG signal.

The next block is an LSTM network consisting of two identical consecutive LSTM modules with 64 units. The role of this block is to capture long-term temporal inter-dependencies in an automated way.

At the end of the network, there is a Multi-Layer-Perceptron (MLP) layer, which consists of three fully connected layers. The first is the input layer, followed by the hidden layer, and finally the output layer. The role of this MPL is to produce the appropriate outputs.

The above-mentioned model was trained using the Adam optimizer with the batch size set to 20.

Dataset Training and evaluation were performed on two datasets. One was the MIMIC-II [60] dataset and the other was The University of Queensland Vital Signs Dataset(UQVSD) [82]. Filtering was performed on both datasets. In the case of the PPG signal, signal components with frequencies outside of 0.1 and 8 Hz were removed using a bandpass filter. The arterial blood pressure signals were filtered with a low-pass filter with a cutoff frequency of 5 Hz. Finally, these signals were resampled at 20 Hz and normalized to zero mean and unit variance.

A one-window-out validation is performed for evaluation. So, in the case of a subject, an 8-second time window is kept separately for validation and one for the test. Training is done on the other data. The estimation errors of the obtained results are illustrated in Figure 3.1.

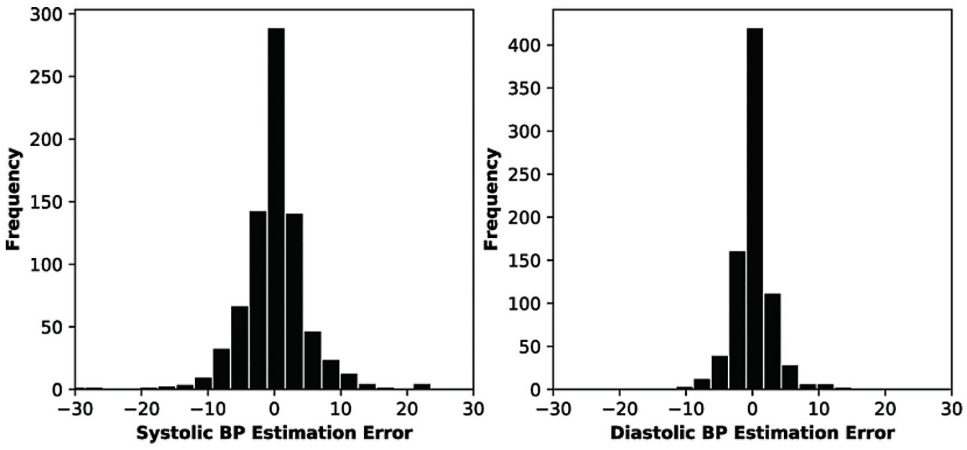

Figure 3.1: Prediction Error for SBP, and DBP [92]

Evaluation The performance of the network was evaluated based on three metrics, all of which are detailed later in Section 6.2. The first metric utilizes the absolute value of errors, which I called Statistical in the previously-mentioned section. Afterward, the British Hypertension Society standard was used for evaluation, which can be used to classify neural networks based on the absolute value of their estimation errors. Based on this metric, this model meets the criteria of the best possible class. Finally, the authors evaluated their model based on the US Association for the Advancement of Medical Instrumentation (AAMI) standard, which criteria are fulfilled by the current network. Based on these metrics, the results of this network are compared in Section 6.4.1 with the results of the approach, presented in the next section, as well as with the results of the model I created.

Conclusion In conclusion, this article is very useful and forward-looking. For my work, I collected several ideas from this article, which I supplemented and thought about further. This information includes the idea of 8 second windows, the extraction of the SBP and DBP, the convolutional layer at the beginning of a neural network to extract features from the input signal, the use of the LSTM layer to properly handle long-term temporal inter-dependencies, and finally the metrics that help evaluate the result produced by the network.

3.2 Approach based on both PPG and ECG

The measurement method with only PPG can be supplemented with ECG, as the ECG contains vital information about the functioning of the heart. By itself, the ECG does not contain a sufficient amount of information, but by combining these two signals, we can obtain significantly more data compared to the approach based only on photoplethysmography. However, the methods that utilize both ECG and PPG signals appear only as research subjects. In the following, I present the work of Yung-Hui Li et al. [61], that deals with this topic in detail.

Background The pulse transit time (PTT) is one of the most important parts of this paper. It can be defined as the time taken by the arterial impulse to travel from the heart to the peripheral location. This value can be obtained by knowing the PPG and ECG signals. PTT can be calculated as the time interval between the ECG peak and the maximum slope (first derivative) of the PPG. This is illustrated in Figure 3.2.

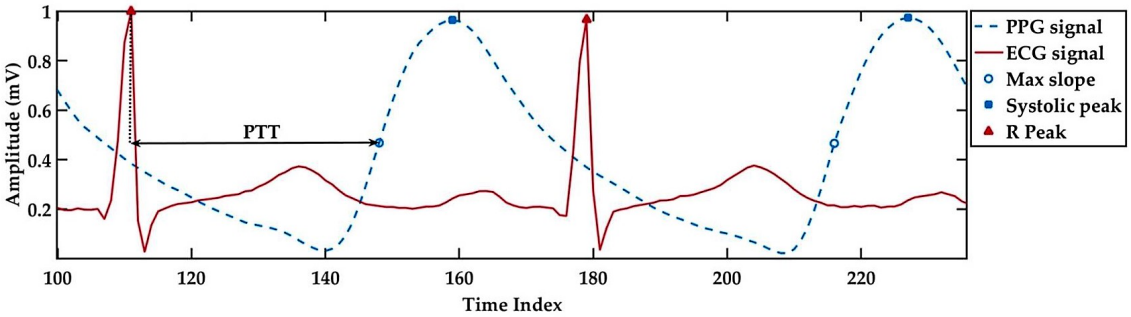

Figure 3.2: Visualization of pulse transit time (PTT) [61]

A derivation can be found in the detailed presentation of the PPT. This shows that the value of the PPT and the value of the blood pressure have a non-linear (more precisely, logarithmic) connection with each other. This implies that the current value of blood pressure can be estimated from the current pulse transit time. Based on this information, it can be concluded that this method provides a great approach to solving the assigned task (continuous blood pressure estimation). However, this is still not straightforward, since many variables are dependent on the specific arterial characteristics of the person.

Equation 3.1 shows the non-linear relationship between PPT and blood pressure.

$$P = -\frac{2}{\alpha} \cdot \ln(PPT) + \frac{1}{\alpha} \cdot \ln\left(\frac{2 \cdot r \cdot \rho}{E_0 \cdot h} \cdot \mathrm{D}^2\right), \tag{3.1}$$

where the radius of the artery is indicated with r (in unit m), and h is the thickness of the artery (also in unit m). E_0 is the modulus of elasticity of the arterial wall at 0 mmHg (in unit $mmHg$), and ρ is the density of the blood in the artery (in unit kg/m^3). Furthermore, if the pulse wave is detected by two sensors, the distance between them is D, measured in meters. Real-valued parameter α is greater than zero and closely related to arterial stiffness. For completeness, P is blood pressure and measured in $mmHg$, and PPT is pulse transmit time and measured in seconds.

From these variables, we can see how complex the task is, because these values are different for each person, and even in the case of a single person these values can vary quickly with changes in the circumstances and time.

Furthermore, the PTT-based blood pressure estimation technique has not been widely accepted yet for cuff-less and continuous BP monitoring [104]. This is because no solution has been created that has satisfactorily low estimation errors and passed all authentication and verification procedures. So in my opinion, it is a really exciting and challenging task to create a solution with adequate accuracy using ECG and PPG simultaneously, which has a huge potential.

Dataset and pre-processing In this article, they used the same database as *Ali Tazarv*, and *Marco Levorato* [92], which is the MIMIC-II [60]. So this data set contains the input and output data for the neural network that will be detailed later. The input signals were filtered before entering the network. Here, the deletion of low-frequency artifacts was emphasized, which means frequencies below 0.1 Hz. After filtering, the next step was normalization, which was done according to Equation 3.2 where x_i is an input sample and x'_i is a normalized one. [61]

$$x'_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3.2}$$

Models and methods Unlike the solution that only estimates from the PPG signal, feature extraction is not done here with the help of convolution layers, but the authors have done it manually. First, they perform ECG R peak detection using the Pan–Tompkins algorithm [71]. Based on this, a time frame can be defined, which is an interval that starts from an R peak followed by a systolic peak and the next R peak and finishes with the following R peak. They define seven features within one window, and the mathematical form and detailed description of them are presented in the article [61].

These features are:

- Pulse Transit Time
- Heart Rate
- Reflection Index
- Systolic Timespan
- Up Time
- Systolic Volume
- Diastolic Volume

These features will be the input for the following networks. Four LSTM models are presented in this paper. Each model starts with a Bi-LSTM (Bidirectional-LSTM) layer. This is followed by a series of LSTM layers, then a fully-connected layer and finally a regression layer as output. Residual connection (marked with arrows in the following figure) is applied between the LSTM layers. The output consists of two scalar, which are the systolic and diastolic blood pressure values. The structures of these four networks are illustrated in Figure 3.3.

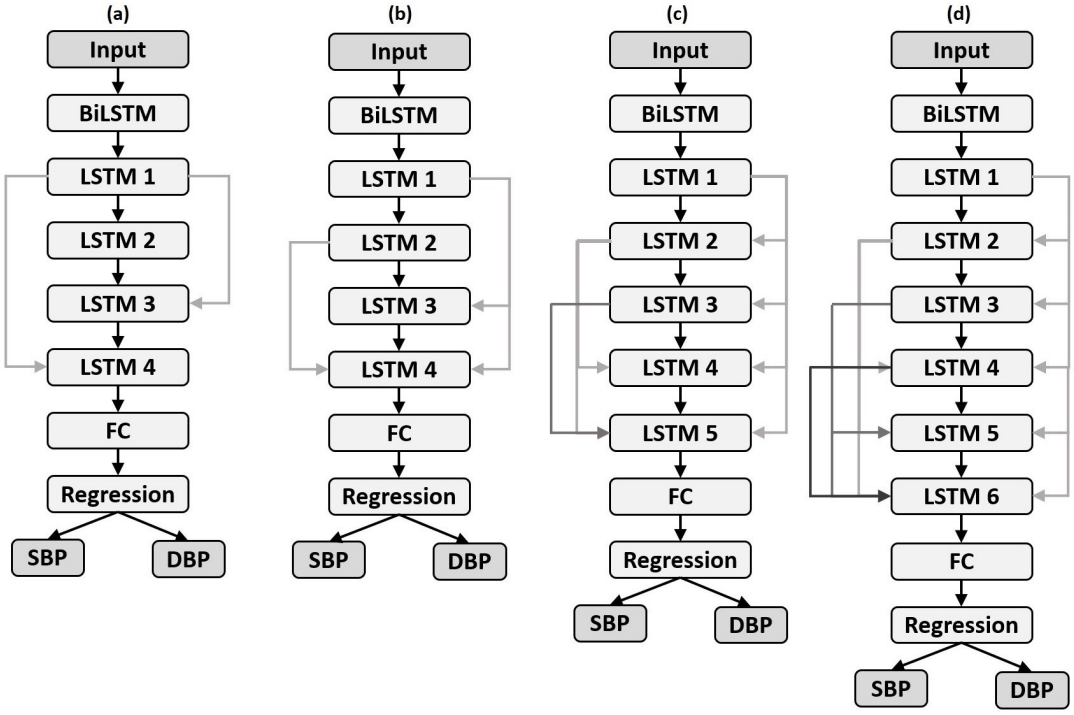

Figure 3.3: The four proposed LSTM models

After filtering the used data set, 1,113,634 cycle records remain from 3,000 randomly selected subjects. SBP and DBP values were extracted from the ABP signals as ground truth for training. Furthermore, parts with some outliers were removed (e.g. SBP 180 mmHg, DBP 130 mmHg, SBP 80 mmHg, DBP 60 mmHg). The final data sets consist of 678,202 records. They keep 80% of them as a training set and the rest for the test set without overlap. So, in the end, 542,561 records are for training and 135,641 for testing.

Evaluation As a first approach, not the entire data set was used, but only 50 subjects. This meant 5482 training and 1370 body records. The authors compared the results with three traditional machine learning methods, Linear Regression [86], Random Forest [76], and Least Squares Boost [47]. The mean value and the standard deviation of the errors obtained during the evaluation revealed that all four LSTM-based solutions outperformed the machine learning-based methods. Table 3.1 shows the results.

Model (b) can be considered as the best network among the models shown in Figure 3.3. Therefore, further evaluations and tests were made on this LSTM model.

After that, using all the available test records, in a similar way to the previous article, Model (b) was evaluated according to three metrics. These metrics were the Statistical, and the procedures based on the British Hypertension Society standard and the US Association for the Advancement of Medical Instrumentation standard. These metrics are

Method	Systolic Blood Pressure MAE	SD	Diastolic Blood Pressure MAE	SD
Linear Regression	9.14	11.50	2.98	1.21
Random Forest	2.60	3.36	3.02	1.39
Least Squares Boost	4.87	6.68	3.45	1.39
Model (a)	1.17	1.40	0.75	0.83
Model (b)	0.74	0.96	0.56	0.51
Model (c)	1.79	0.81	0.75	0.48
Model (d)	1.24	0.73	1.03	0.46

Table 3.1: Performance according to absolute error (units are measured in mmHg)

detailed later in Section 6.2. Based on the BHS standard, this model received the second-best grade for systolic blood pressure values and the best for diastolic blood pressure values. Furthermore, it could not fulfill all of the AAMI criteria. The results of Model (b) are compared with the results of the model in the previously-mentioned article and with my approach in Section 6.4.1.

Conclusion This article contained a lot of useful information that I later used in my work, for example, the combined utilization of ECG and PPG. It also helped me to learn about the nature of signals and the possibilities offered by deep learning. Maybe the authors could have achieved better results, for example, if they did a longer training or did not perform the feature extraction by themselves. But in my opinion the aim of the article was not to achieve a very high result but to show the potential of this approach.

Chapter 4

Methodology

This chapter presents the pre-processing methods required for the different tasks of the implemented system, as well as the created neural networks, with particular regard to the model utilized in the implemented system.

4.1 Pre-processing

In the implemented system, we have two input signals, the ECG and the PPG. The board used for measurement has passive RC low-pass filters with a cutoff frequency of 319 kHz at both of the sensor unit's ECG inputs. The purpose of this analog filter is to reject high-frequency electromagnetic interference (EMI). The low-frequency filtering is performed by the internal filters of the MAX86150. [3] Also, the PPG signal is filtered by the sensor unit, which includes a discrete time filter to reject 50Hz/60Hz interference and slow-moving residual ambient noise. [4]

According to the previously-mentioned information, the input signals are received by the core unit in an already pre-processed form, however, their subsequent use requires further processing, which is presented in the following.

4.1.1 Producing the proper number format

As the first step of the pre-processing, these received signals must be converted to a proper number format from a bit array, which is the integer, because they are ADC values. This conversion had to be done according to the information that the received ECG and PPG value is represented in two's complement. Then I calculated the resulting values into physical units of measure, which help the displayed data to become more understandable.

In the case of PPG, this can be done using Equation 4.1, which will give the result in nA.

$$PPG = \frac{ADC_{int}}{2^{19} - 1} \cdot 32768 \; [nA] \tag{4.1}$$

The 32768, shown in Equation 4.1, is the full-scale value of the ADC belonging to the PPG in nA units, and the 19 in the exponent of 2 is the resolution of the ADC in bits. These data can be found in the datasheet of the sensor [4].

In the case of ECG, according to the datasheet [4], Equation 4.2 can be used to convert the ADC value, obtained in integer, into mV units.

$$ECG = ADC_{int} \cdot \frac{12.247}{8 \cdot 9.5 \cdot 1000} \, [mV] \tag{4.2}$$

The value 12.247 shown in Equation 4.2 is given in μV, however, since the result is needed in mV, it still needs to be divided by 1000, which is observable in the denominator. Furthermore, the value 8 is the PGA (Programmable Gain Amplifier) ECG Gain, and the value 9.5 is the IA Gain (Instrumentation Amplifier Gain), both of which have units of $\frac{mV}{mV}$. All the previously-mentioned values can be found in the datasheet [4].

4.1.2 Pre-processing for the neural network

As soon as we have at least 1,600 pieces of data in a suitable format, the neural network can operate using these data as input. However, before we direct these data to the input of the network, certain pre-processing must be performed. In the following two Numpy [16] arrays of 1600 elements are processed, which contain the PPG and ECG signals.

The neural network was created for the publicly available database, presented in Section 6.1.1, in which the signals were sampled at 125 Hz. Since the settings of MAX86150EVSYS allow a minimum sample rate of 200 Hz, the measured signals needed to be resampled at 125 Hz to produce the correct input for the neural network. The resample() [20] function of the Scipy [22] program library helped me solve this task.

After re-sampling, the signals are filtered with a Butterworth [43] band-pass filter. I chose this type of filter because I didn't want to emphasize specific frequencies because of the waves in some other types of filters, and the exact cutoff frequencies were not known, so there was no need for an abrupt cutoff in the amplitude characteristic. The so-called maximally flat magnitude filter, i.e. the Butterworth filter, meets these criteria. This method accomplishes this task with the help of the butter() [7] function provided by the Scipy [22] Python library. I gave this function the following inputs:

- The order of the filter: N = 3
- The cutoff frequencies at which the gain drops to $1/sqrt(2)$ of the passband also known as, the -3 dB points: Wn_{ECG} = [1, 50], Wn_{PPG} = [0.8, 12] (Units are measured is Hz.)
- Sampling frequency of the digital system: fs = 125, due to the previously mentioned resampling

The amplitude characteristic of the transfer function of the filters obtained with these parameters can be seen in Figure 4.1, where the x-axis is represented with a logarithmic scale, while the values on the y-axis are expressed in decibels with a linear scale.

The signals are filtered with Scipy's filtfilt() [12] function, which performs the filtering with the previously-mentioned Butterwort filter in such a way that the filter moves from the beginning to the end of the signal and back. This method implements a zero-phase filter with twice the order of the base filter. The zero phase is important since the relative location of the values of the two signals provides important information, so if either signal suffers a shift, the estimation ability of the network may deteriorate.

After that, I only converted the PPG signal. This is necessary because the training data of the pre-trained network was measured in a different measurement arrangement compared to MAX86150EVSYS. As a result, the waveform of the current, acquired during one

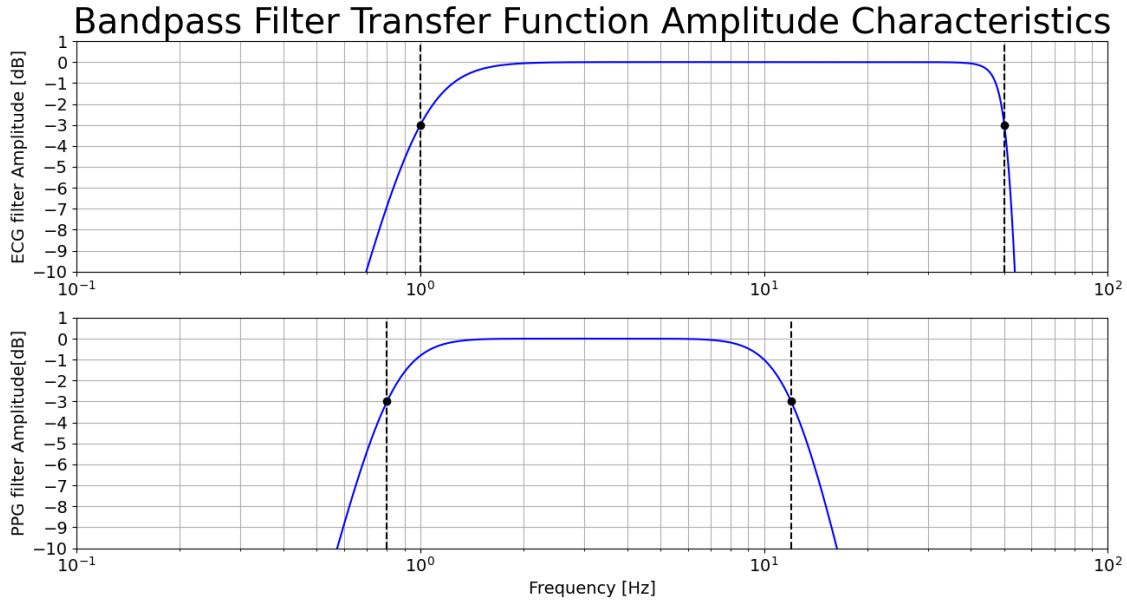

Figure 4.1: Amplitude characteristics of the used band-pass filters

measurement, can be obtained by reflecting the other on the x-axis. So, while according to one case the largest current is generated, in the other it means the smallest. Based on this information, I made the conversion of the PPG.

Finally, I standardized the signals in such a way that I subtracted the average of the signals from all the elements of the signal and then divided this difference by the standard deviation of the signals. The method of standardization for one element is shown by Equation 4.3, where μ denotes the mean of the input values, σ denotes the standard deviation of the input values, x is one element of the input, and z is its standardized version of this element.

$$z = \frac{x - \mu}{\sigma} \tag{4.3}$$

Following the sequential execution of the previously mentioned procedures, we obtained ECG and PPG values in a suitable form for feeding them to the neural network.

4.1.3 Pre-processing for the heart rate computation

To determine the heart rate, it is necessary to process the time frame containing the ECG values. During the processing, I want to perform operations on the ECG that result in a signal that can be easily used to determine the number of heartbeats within the given time frame. The heartbeats in the ECG signal are identified by their corresponding R peaks, from the number of which the heart rate can be calculated. I made this pre-processing using the Pan–Tompkins algorithm.

I implemented the algorithm as follows:

- Band-pass filter: Here, I utilized the Butterworth filter presented in Section 4.1.2, with the difference that I set the sampling frequency to 200 Hz since resampling was done here. Similar to Section 4.1.2, I performed this filtering with the Butterworth filter using the filtfilt() function.

- Derivative filter: To implement this, I used a Savitzky–Golay filter[1] with the help of the savgol_filter() [21] function from the Scipy Python library. Here I used the following parameters:
  - The length of the window, which goes through the entire signal, similar to the convolution operation: 16.
  - Order of the filter: 6.
  - Order of the derivative: 1, because the first derivative is needed.
  - The type of extension to use for the padding of the signal: 'Nearest', which means that the padding contains the nearest input value.
- As the next step, I squared the elements of the signal obtained as the output of the previous step.
- Moving window integration: to implement this, I used Scipy's convolve1d() [10] function, which can implement one-dimensional convolution along a one-dimension array. The filter, given as an argument of the function, was a fifteen-element array containing only ones to implement integration operation.

I found these parameters and settings to be the most suitable for this task, the result of which is illustrated in Figure 4.2. The upper part of the figure shows the input ECG signal before these operations. And the lower part is the processed version of this, on which the red dots indicate the detected peaks.

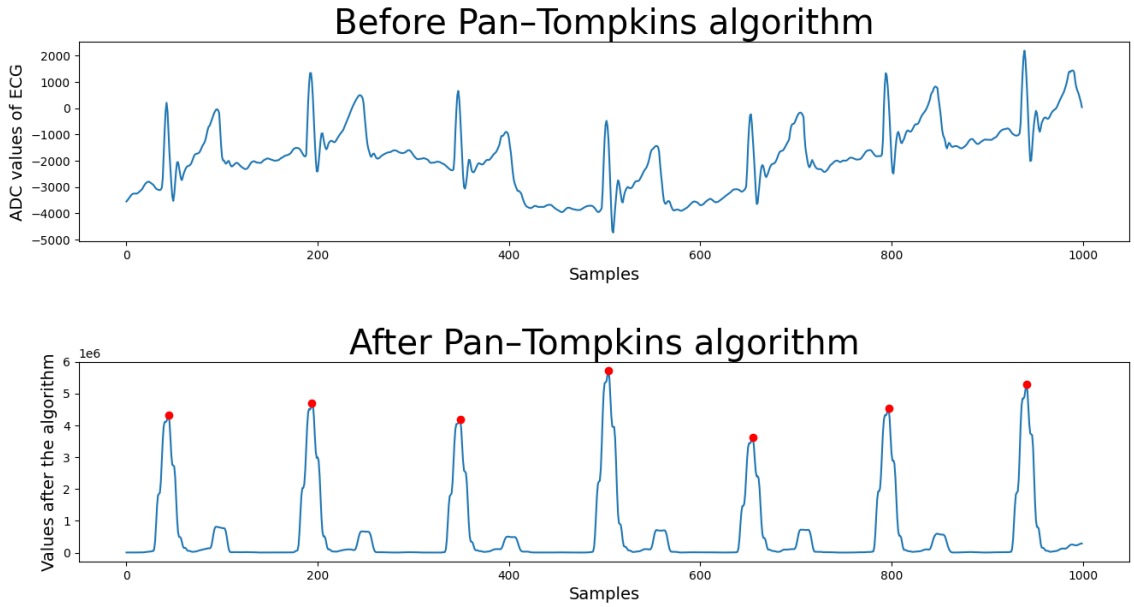

Figure 4.2: Result of the Pan–Tompkins algorithm on ECG

After performing these transformations, the R-peaks can be detected in the resulting ECG signal and based on these information, the heart rate can be calculated.

[1]The Savitzky–Golay filter is a digital filter that, in the case of equally spaced data points, is capable of smoothing the signal and calculating the derivative of the smoothed signal without distorting the signal tendency. [85]

4.1.4 Pre-processing to verify compliance

To determine whether the ECG and PPG signals are appropriate, they must be properly pre-processed. If the number of R peaks and systolic peaks are known, then the adequacy of the input signals can be determined based on the relationship between these values.

The pre-processing described in Section 4.1.3 creates the opportunity to transform the ECG signal in such a way that the R peaks can be easily detected. However, I also applied this algorithm to the PPG signal, the parameters of which I only changed that I moved the cutoff frequencies to 0.8 Hz and 12 Hz, due to the different characteristics of the signal.

Figure 4.3 shows the result of the algorithm. The upper part of the figure shows the input PPG signal before these operations. And the lower part is the processed version of this, on which the red dots indicate the detected peaks.

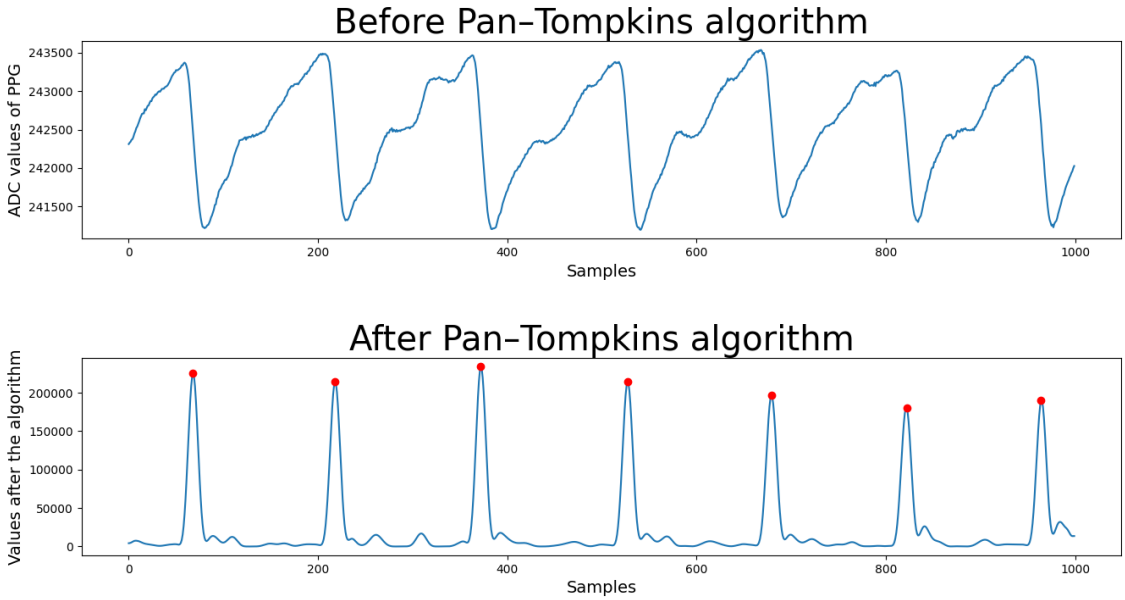

Figure 4.3: Result of the Pan–Tompkins algorithm on PPG

A slightly different application of the Pan–Tompkins algorithm on ECG and PPG results in signals in which the main peaks can be easily detected and counted. Based on this information, it is possible to decide whether the input data is correct, the method of which is detailed in Section 5.4.5.

4.2 Neural network design

During the preparation of this study, I created several deep learning models to examine which one is the most suitable for solving the given task. Three of the implemented models are presented in this chapter, as the others were made with minor changes from them. These three neural networks are illustrated in Figure 4.4.

All of these networks were pre-trained with data from the public database available on Kaggle. In the following, the models and their layers, shown in Figure 4.4, are detailed.

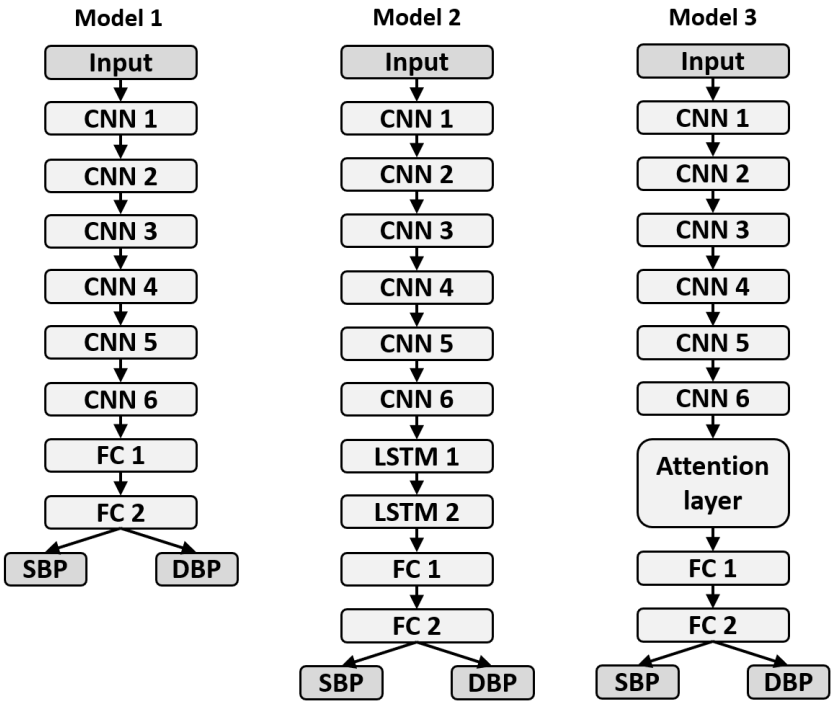

Figure 4.4: Structure of the implemented models

Model 1 The model has six convolutional layers. After the CNN layers, shown in Figure 4.4, some layers are not visualized for the sake of conciseness, however, there is a batch normalization layer, a RelU layer, and a dropout layer with 10% masking probability after each convolution layer. Furthermore, after the first and second convolution layers, I apply average pooling with a kernel size of 2. In Table 4.1, the number of filters, the kernel size, and the stride are presented according to these layers.

Layer	Number of filters	Kernel size	Stride
CNN 1	512	8	2
CNN 2	256	4	1
CNN 3	128	4	1
CNN 4	64	4	1
CNN 5	32	4	1
CNN 6	16	4	1

Table 4.1: Parameters of the CNN layers

In addition, in CNN 1 padding is not applied, while in the case of the other convolutional layers I used the so-called "same" padding, due to which the length of the output signal sequence is the same as the input length.

At the end of the model I used two fully-connected layers. The first produces 512 output features, followed by a ReLU and a dropout layer, which are not shown in Figure 4.4 for the sake of conciseness. And the FC 2 has two outputs, which correspond to systolic and diastolic blood pressure values.

Model 2 This model contains exactly the same network of six convolutional layers at the beginning that I presented according to Model 1. The purpose of these layers is feature extraction, i.e. to learn patterns in the time and frequency domain, thus highlighting the

important information of the input signal. After these layers, I used two identical LSTM layers. Together, these layers form a so-called stacked LSTM, in which the input of the second layer is given directly by the output of the first layer. Furthermore, the number of features in the hidden state is 128 in both layers, and I also used a dropout of 0.1 for the training process. These LSTM layers aim to learn the patterns found in the temporal sequence. At the end of the model, I used the same two fully-connected layers that I mentioned according to Model 1.

It is worth mentioning that I also created this model with residual connections across convolutional layers, using bidirectional LSTM[2] instead of LSTM, using GRU and bidirectional GRU[3] instead of LSTM, but since these did not perform better than Model 2, I kept it.

Model 3 This model started with the same six convolution layers and ended with the same two fully-connected layers as the other two models. However, instead of the two LSTMs, attention layers, described in Section 2.5.5, are used here. The attention layer aims to examine the entire input sequence to find patterns and relationships. The number of input features of this layer is 16, the number of heads is 8, and the dimensions of the feed-forward network are 1024. Furthermore, I set a dropout to 0.1 for the training process. I adjusted the number of layers to 4, which means this model uses 4 attention layers with the previously mentioned parameters, which is not visualized in Figure 4.4 for the sake of brevity.

In the case of this model as well, I attempted to create residual connections across convolutional layers, but I could not achieve a significant increase in performance with them.

Conclusion After evaluating the performance of all three networks, I came to the decision to use Model 2 in the created system. The basis for this decision was that Model 2 exceeded the results of Model 1 in terms of estimation errors, so I chose Model 2 from these two models, which can be understood as an extended version of Model 1. Model 3 produced similar results as Model 2 according to estimation errors, however, Model 3 had higher values both in the number of parameters and in the network's runtime, so in terms of the final implementation, Model 2 proved to be more appropriate.

In conclusion, Model 2 has the best performance among the described models, and as a result, this algorithm is used in the implemented system to estimate blood pressure values. However, to learn the characteristics of the data measured by the MAX86150EVSYS, I applied transfer learning[4] on this pre-trained model using the data recorded by the previously-mentioned device. As a result, it can estimate the systolic and diastolic blood pressure from the signal provided by the MAX86150 sensor with a low estimation error.

[2]The bidirectional LSTM supplements the unidirectional LSTM with another LSTM layer, in which the direction of the information flow is reversed, i.e., the input sequence flows backward. The outputs of the whole layer are calculated with the combination (e.g., sum, average) of the outputs of the two layers. [107]

[3]Analogous to the bidirectional LSTM.

[4]Transfer learning is a machine learning methodology designed to transfer knowledge across domains. This allows a model to utilize knowledge of more general information to solve more specific problems. [106]

Chapter 5

Implementation

In this chapter, I describe the implementation of my task. The hardware units used and the system resulting from their connection, as well as the software running on the central unit are presented in detail.

5.1 System plan

The central unit of the system is the Raspberry Pi 4 Model B, on which an application runs, which has four tasks. The first is pre-processing, which converts the data, received from the MAX86150EVSYS, into a suitable form for computation and display. Next is the communication, which is responsible for the Bluetooth connection with the MAX86150EVSYS. The third is computation, within the framework of which the systolic and diastolic blood pressure values are estimated, and the heart rate is computed. Finally, the graphical user interface (GUI) is responsible for the appropriate display of the received and calculated data. This core unit is connected to all other elements of the system. The MAX86150EVSYS sends and receives messages via Bluetooth, so this communication is two-way. Furthermore, it receives data from the computer mouse, while it sends data to the monitor. For easier understanding, the block diagram of the system is shown in Figure 5.1.

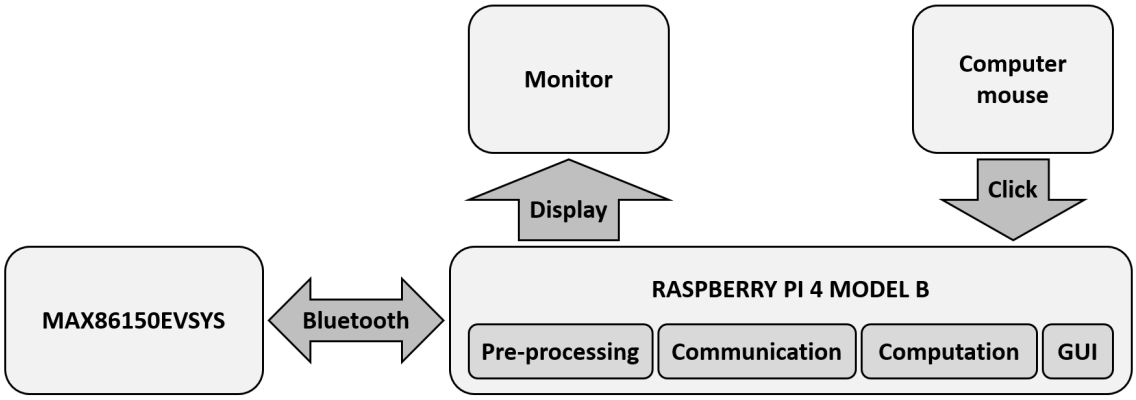

Figure 5.1: Block diagram of the system

5.2 Hardware components

Besides the peripherals, the system contains two hardware units. These components provide the proper functioning of the system. In the following points, these hardware units are presented in detail.

5.2.1 Raspberry Pi 4 Model B

This hardware component forms the central unit of the system, and all the other hardware units communicate with it. The Raspberry runs the application, created in this work, which controls the graphic display, the running of the neural network that estimates blood pressure, the determination of the heart rate (HR), and the communication with MAX86150EVSYS via Bluetooth.

This Raspberry Pi 4 Model B is a small-sized, low-power, but powerful computer. This hardware is used for different applications like smart home hub, media center, factory controller, and much more. [72] The task-related technical specifications can be found in Table 5.1.

CPU	Broadcom BCM2711, quad-core ARM Cortex-A72 (ARM v8) 64bit SoC @ 1.5GHz
Memory	2GB LPDDR4-3200 SDRAM
Ports	micro-HDMI, USB 3.0, USB 2.0, Bluetooth 5.0

Table 5.1: Technical specifications of the Raspberry Pi 4 [73]

5.2.2 MAX86150EVSYS

The MAX86150EVSYS [2] (evaluation system) provides a platform for using the MAX86150 integrated PPG and 1-lead ECG sensor module. The EV system consists of two boards, a MAX32630FTHR microcontroller board, and a MAX86150 evaluation kit. The first board contains an embedded processor called Cortex-M4F [1], which combines high-efficiency signal processing functionality with the advantages of low power consumption, low cost, and ease of use. This unit is responsible for Bluetooth communication and power management. The sensor board contains the MAX86150 module and two stainless steel dry electrodes for ECG measurement. Furthermore, this system is powered by a lithium-ion battery. Figure 5.2 shows this system.

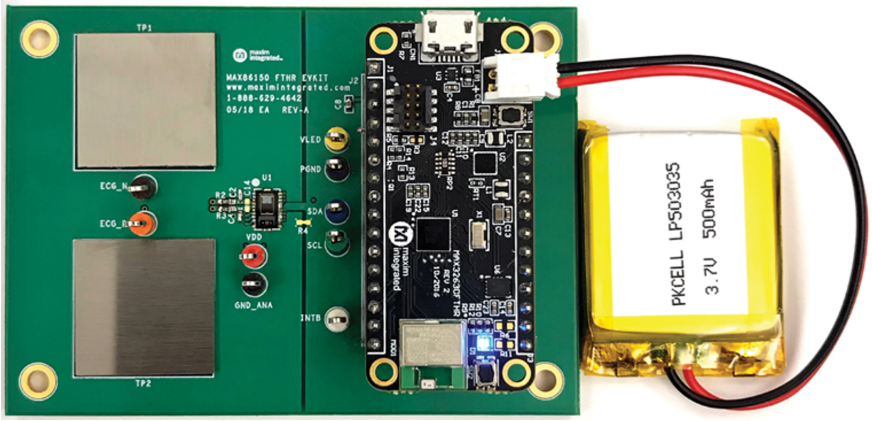

Figure 5.2: MAX86150 Evaluation System [2]

It is important to highlight that the MAX86150 [30] sensor module is the most important part of the system, as it contains an integrated electrocardiogram and photoplethysmogram sensor. It also includes internal LEDs, a photodetector, and low-noise electronics with ambient light rejection. This evaluation kit operates from a 1.8V supply with a separate power supply for the internal LEDs. Communication with the microcontroller is via a standard I^2C-compatible interface. Figure 5.3 shows the block diagram of this sensor module.

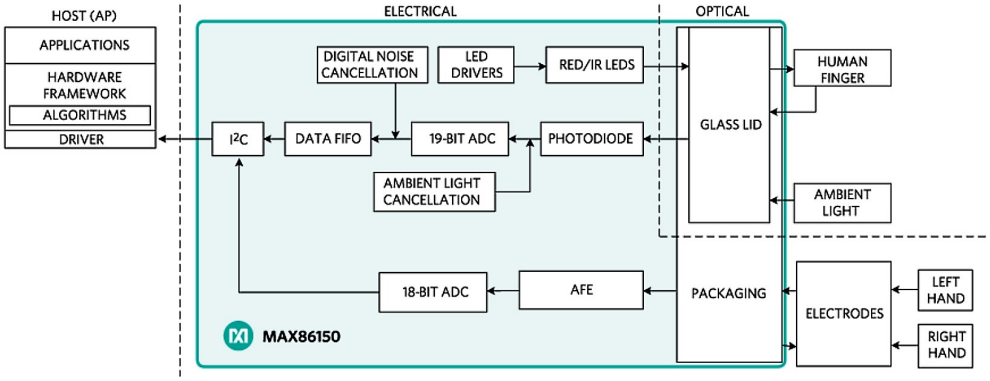

Figure 5.3: MAX86150: Simplified Block Diagram [30]

During the solution I implemented, I did not use the built-in pair of electrodes. The reason for this was that it was probably a manufacturing defect since when measuring the resistance of one electrode, a value of around 50 Ohm was obtained, while according to the datasheet, it should work as a short circuit, as the other electrode does. This error was noticed in such a way that the received waveforms did not resemble the ECG signal. After this observation, the process of troubleshooting began, during which the cause of the inappropriate waveforms was found. I solved this problem by soldering two electrode wires used for ECG measurement to the soldering of the built-in electrodes. The resulting ECG waveform has become much more appropriate.

In the case of this hardware component, it is necessary to mention the associated client application, which displays the waveforms of the measured data and also allows several settings. For instance, in terms of ECG, it is possible to adjust the adaptive filter, as well as the cutoff and notch frequency, or in the case of PPG, the AGC (Automatic Gain Control). These settings affect the measurement via software. The main view of the client application is shown in Figure 5.4.

One of the most practical features of this client application is the register map, which shows all the registers of the MAX86150 and allows you to modify the writable registers through this interface. Another mentionable function is that it allows saving the measured data, which was extremely useful when I created my dataset.

5.3 User interface

Monitor The monitor connects to the Raspberry Pi 4 Model B. Its job is to display the graphical user interface that runs on the Raspberry.

Figure 5.4: MAX86150EVSYS clinet application

Computer mouse A computer mouse enables user interactions. It connects and gives information to the Raspberry Pi 4 Model B. Thus, the user has the opportunity to start and stop the measurement, as well as to open and close the application.

5.4 Application

The application running on the Raspberry Pi 4 Model B was created in Python programming language [18]. The reason for this is that the Raspberry runs a Raspberry Pi OS (previously called Raspbian) that is capable of running Python programs, and also because it is a high-level programming language that enables rapid development. The most important parts in the structure of the application are illustrated in Figure 5.5 using the UML (Unified Modeling Language) class diagram. In the following, the program is detailed from the point of view of functionality.

5.4.1 Graphical user interface

The Graphical user interface is the component of the program that runs on the main thread. It does not perform calculations, its task is only to display the graphical elements (e.g. buttons), the waveforms, and the proper output values correctly. The Python file that implements this component contains a class, called App, that holds the components together. It can be called the main component since it contains as attributes the class of the communication component, the neural network model, and the heart rate computation component. When the App is instantiated, the constructor creates its attributes with their corresponding initializer values. It also creates a graphical interface with graphical elements.

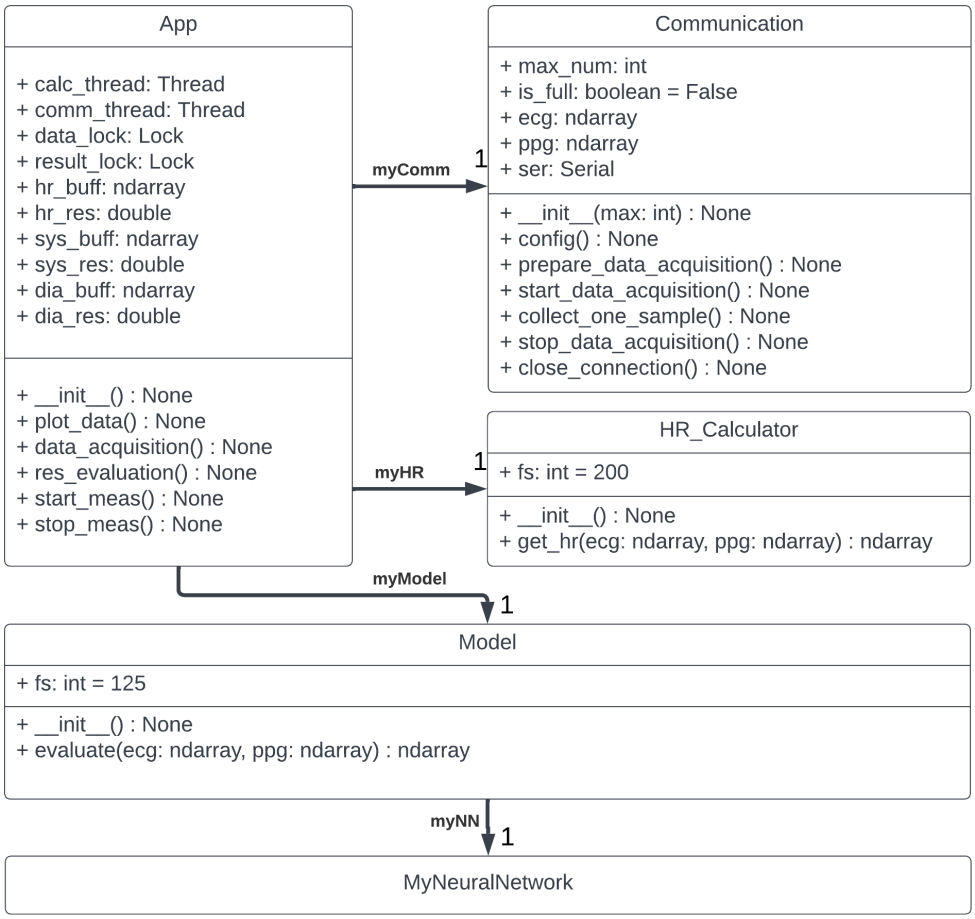

Figure 5.5: UML class diagram of the application

To implement this graphical part of the task, I used the tkinter [24] Python library, which simplified the graphic display with several built-in functions and graphical elements. The resulting graphical interface is shown in Figure 5.6, which contains several graphical elements, including:

- The central element of the application is a figure that displays the ECG and PPG input signal without further filtering. Before display, only the ADC values are converted into the appropriate numerical format and correct physical units, which is detailed in Section 4.1.1. This figure consists of two subplots, the upper one shows the ECG signal in the mV unit, and the lower one shows the PPG signal in the nA unit. In both cases, the sample indexes are found on the x-axis, and the signals are displayed as a function of the samples. The figures show 8 seconds of data at a time. As soon as the measurement exceeds this value, the oldest input sample will be discarded, so the data storage procedure works like FIFO (First In First Out).
- Two buttons can also be observed, the first one indicates the start, and the other indicates the stop of the measurement.
- There are also three panels on the right side of the display, which I implemented using label elements. These labels display the systolic blood pressure, the diastolic blood pressure, and the heart rate.

- There is a title element showing the title at the top of the canvas.
- In addition, the x in the upper right corner allows you to exit the application.

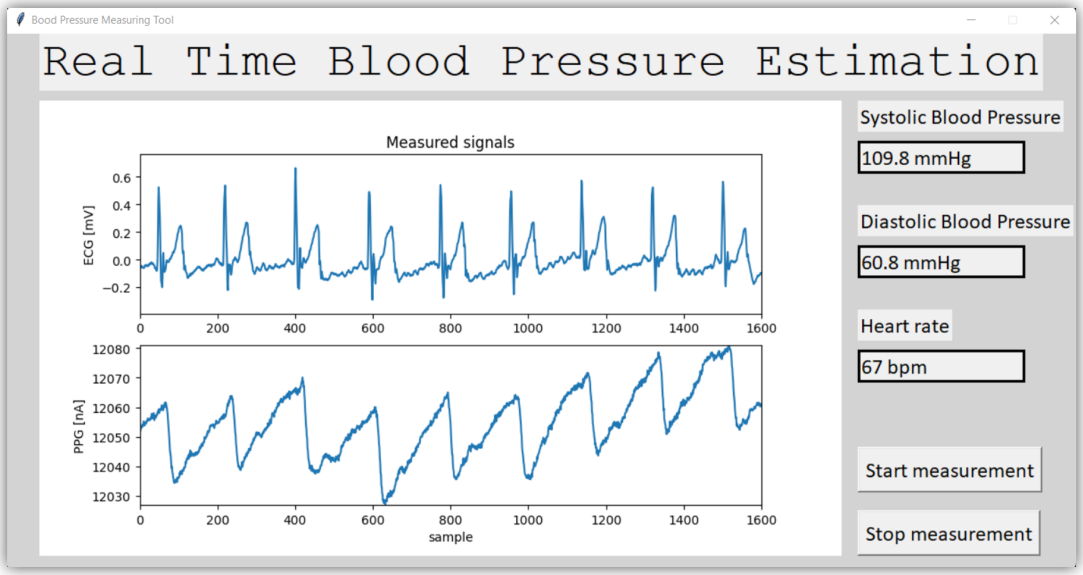

Figure 5.6: The graphical user interface

5.4.2 Logical part

The App class has many attributes. For instance, one of the attributes is the class that enables communication (*myComm*), and other attributes are the class that implements a neural network (*myModel*) or the one that computes the heart rate (*myHR*). The App also contains Numpy's ndarrays for the computed results (hr_buff, sys_buff, dia_buff), the average of which gives the attributes containing the displayed double values (hr_res, sys_res, dia_res). It also creates two threads, which I implemented using a Python program library called Threading [23]. The first thread (*comm_thread*) is responsible for the continuous collection of data with the help of the communication component. The second (*calc_thread*) is responsible for calculating the results utilizing the neural network and the HR computation components. Since there is also communication between the threads, lock elements are needed to protect variables against possible problems caused by simultaneous access of the same variable by different threads. The first lock (*data_lock*) aims to protect the variables that store the measured input ECG or PPG samples in the class responsible for communication. This lock is necessary because, in the case of active measurement, the communication component continuously writes these blocks, and the thread, responsible for the graphical display, also reads them periodically. To prevent possible simultaneous reading and writing, I use a lock to make these reading and writing operations atomic. The other lock (*result_lock*) protects the variables containing results, written by the thread responsible for result computation. The App also reads these variables periodically, so the lock is necessary in this case.

The app class has several methods in addition to the constructor, which are the following:

- *plot_data()*: This method is responsible for writing and plotting: Depending on the adequacy of the input, it displays the outputs and the input signals. If the inputs

are incorrect, a line is displayed in place of the outputs to indicate that there is no output to display.

- *data_acquisation()*: This method runs on a separate thread. If the measurement has already started, then continuously read the data with the help of the communication component.

- *res_evaluation()*: This also runs on a separate thread. This method is responsible for evaluating the heart rate (HR), determining the quality of the inputs, estimating the blood pressure, and writing the results into the appropriate attributes.

- *start_meas()*: This is the event assigned to the start button, so if someone presses the start button, this function starts running. The task of this method is to reset the buffers and start data collection.

- *stop_meas()*: This is the event assigned to the stop button, so if someone presses that button, this function starts running. The task of this method is to stop the measurement.

The App class descends from a class defined in tkinter, which represents the main window of an application. Several useful functions are available thanks to inheritance. For example, the after() method, one of whose arguments is a time specified in milliseconds, and the other is a function that is called after the specified time. With this function, I resolved to call the *plot_data()* method 1 *ms* after it finished its previous running. Furthermore, inheritance also gives access to the method that starts the execution of the graphical interface, called mainloop().

As soon as we exit the graphical interface, the program breaks the connection using the component responsible for communication, then waits for the other threads to finish running using the join() operation.

5.4.3 Neural network model

This software component is responsible for producing the systolic and diastolic values based on the input data. The measurement can be started via the graphical user interface. As soon as the first 8-second sample arrives, the execution of the neural network begins on a separate thread. After estimation, it returns the obtained results to the graphical interface. Then, this thread repeats this process again and again until the measurement is stopped.

So there are two classes in the python file that implements this component:

- The first class implements the chosen neural network model (Model 2) described in Section 4.2 using the Python library called Pytroch [19]. In this class, there are attributes representing individual layers of the network, as well as an additional method beside the constructor, the so-called forward function, which returns the estimated values if the input is adequate according to the dimensions. For the sake of brevity, this class is not detailed in the UML diagram shown in Figure 5.5.

- The second is a class whose attribute is the previously-mentioned class that implements the neural network. Furthermore, it has an evaluation method, which expects two Numpy [16] arrays of 1600 elements that contain the PPG and ECG signals. Inside this function, the pre-processing described in Section 4.1.2 takes place. The

resulting signals are suitable for feeding them to the neural network, which part runs after the pre-processing. So essentially this class provides a method that evaluates the neural network given the appropriate inputs and returns an array whose first element is the systolic blood pressure value and the second element is the diastolic blood pressure value.

In the thread that runs the network, the heart rate is also evaluated by a function provided by the Heart rate computation component. This method also provides information that determines whether the input data is correct. If it is, then the thread runs the neural network on this data, and places the resulting output in a 5-element array. Then the average of these array's elements is copied to the variable that will later be read and then shown on the screen by the graphic interface. This is necessary to smooth the output and thereby increase the user experience.

5.4.4 Communication

This software component enables communication with the MAX86150EVSYS via Bluetooth. Therefore it establishes and maintains this Bluetooth connection, as well as performs all tasks related to communication. To achieve an encapsulated implementation of these functionalities, I implemented all communication-related parts of the program in one class, called Communication.

After the connection is established, it sends configuration commands to the evaluation system, with which I set several parameters that significantly affect the measurement. Among other things, these are the most remarkable adjustment according to ECG:

- IA Gain (Instrumentation Amplifier Gain) = 9.5 mV/mV, which adjusts the gain of the differential input chopping instrumentation amplifier
- PGA (Programmable Gain Amplifier) ECG Gain = 8 mV/mV
- Sample Rate = 200 Hz, which is the output data rate of the delta-sigma ADC
- Adaptive filter: ON, which enables an adaptive hybrid-time-domain software filter
- Notch Freq = 50/60 Hz, which sets the software notch filter
- Cutoff Freq = 50 Hz, which can set the cutoff frequency in the software low-pass filter

The most relevant PPG settings are the following:

- AGC (Automatic Gain Control): ON, which enables automatic gain control, which allows the evaluation system software to dynamically adjust LED currents and ADC range
- IR PA (Pulse Amplitude) = 10 mA, which sets the amplitude of the infrared LED current
- ADC Range = 32768 nA, which sets the full-scale range of the photodiode
- ALC + FDM: ON, which provides ambient light cancellation
- Pulse Width = 400 μs, which sets the pulse width of the LED current

- IR Range = 51 mA, which sets the infrared LED current range
- The FIFO, which contains the sampled data, can also be parameterized

The above-mentioned parameter set was selected according to the recommendations of the datasheet [4].

As soon as the measurement starts via the graphic interface, it sends a command to the sensor card so that it can start sending data. After that, the program continuously reads the incoming data on a separate thread. The data frame received from the evaluation system is in Figure 5.7.

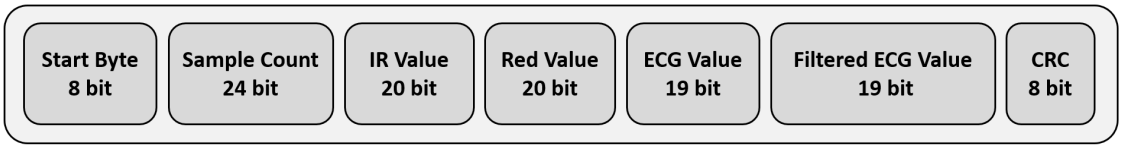

Figure 5.7: Data frame received from MAX86150EVSYS

In this frame, the start byte is an especially helpful section, because it indices the start of the frame. Therefore, its arrival is continuously monitored by the component after the beginning of the measurement. After the program finds this byte among the incoming data, the physiological signal-related sections will be collected. So only the filtered ECG and IR are needed because they need to be displayed and used by other components to calculate the HR and the blood pressure. After selecting these values, using the procedures presented in Section 4.1.1, I converted the obtained values to the appropriate format and unit.

Following that, the received values are stored in two 1600 elements long Numpy array [15], which means eight seconds of stored data because the sampling rate is 200 Hz. These arrays read later by the components managing the graphical interface and the neural network and heart rate calculation.

As soon as the measurement stops, this component sends a command to stop receiving data via Bluetooth. If the application shuts down, it breaks the Bluetooth connection with the MAX86150EVSYS device before the program ends.

5.4.5 Heart rate computation

The class implements this component contains an essential function. This function is called in the thread in which the blood pressure values are estimated. Similar to the neural network evaluation method, this function has two inputs, the ECG and PPG signals. Using these inputs, it performs two tasks. One is to evaluate the heart rate (HR), and the other is to judge the adequacy of the input data.

To determine the heart rate, the input signals are pre-processed using the methods described in Section 4.1.3. After performing these transformations, the R-peaks can be easily detected in the resulting ECG signal. Based on this information, the heart rate can be produced. I calculated it by dividing the difference between the indices indicating the position of the first and last found peak of the given frame by the number of sections between these peaks, which quantity is one less than the number of these peaks, so I get the average distance between the neighboring ones. For a better understanding, Equation 5.1 represents the above-mentioned operation.

$$ECG_{interval} = \frac{Peak_{last} - Peak_{first}}{NumberOfPeaks - 1} \tag{5.1}$$

The result can be calculated in time units by multiplying the result obtained in the number of indexes by the sampling time. Then, divide 60 by the average time between R-peaks obtained earlier for the input time frame, which gives the desired heart rate in bpm. Equation 5.2 represents this operation.

$$HR = \frac{60}{ECG_{interval} \cdot \frac{1}{200}} \; [bpm] \tag{5.2}$$

For the second task of the function, the input signals are first pre-processed according to the methods presented in Section 4.1.4, as a result of which the main peaks of the input data can be easily detected and counted. Based on these information, the function can decide whether the input data is correct. This information is needed because the MAX86150EVSYS does not provide information about whether a particular sensor is being used properly, so I had to solve this task myself. Therefore, the other value that this function returns is the suitability of the inputs for further computations. From this point of view, three possible outcomes can occur, which are detailed in the following list:

1. The first is that both inputs are suitable for calculating HR and blood pressure (BP). In this case, ECG is suitable if the heart rate calculated from its peaks is between 30 and 225 bpm. And PPG is appropriate if the difference between the number of ECG and PPG peaks is less than or equal to one, according to the fact that every ECG peak is accompanied by a PPG peak.

2. The second indicates that, according to the previously-mentioned interval, the ECG is correct, so the heart rate is computable. However, if the number of peaks in the two signals differs by more than one, and the number of ECG peaks gives a reasonable HR value, I assume that the PPG sensor is not being used properly. Therefore, blood pressure values are not producible from the input data without the correct PPG.

3. Finally, I also distinguish the case when the ECG signal is incorrect, according to the previously-mentioned interval. In this case, the user is probably not touching the ECG electrodes properly, so neither heart rate nor blood pressure can be evaluated.

So the output is a two-element array, the first element of which indicates the quality of the inputs, and the second is the value of the calculated heart rate.

As mentioned in the Neural network model, the thread contains a cycle that begins when the measurement has been started and the first 8 seconds of data has arrived. After that, the cycle runs continuously until the measurement is stopped. In the body of the cycle, the previously detailed function that calculates the HR is called, the result of which determines which calculations should be performed. If the HR is appropriate, I collect the currently received values in a 5-element array, similar to what I do in terms of blood pressure. Then I put the average of the current elements of the array into a variable that the component, responsible for the graphical interface, reads and then displays. Also, boolean-type variables are set during the cycle so that the graphical interface can obtain information about which inputs are appropriate.

Chapter 6

Experiments and results

Within the framework of this chapter, the data used in the training process and during testing will be presented. The parameters and methods of training are also mentioned. Furthermore, the used metrics are detailed, as well as the performance of the implemented networks based on these metrics. However, the results of the model used in the implemented system are also presented with different visualization methods.

6.1 Dataset

In this section, I present the sources from which I obtained data for training neural networks and evaluating their performance. Furthermore, it is described how I arranged these data into a suitable form for operations with the neural networks.

6.1.1 Publicly available dataset

I used a publicly available dataset [55][56] for the training, validating, and testing of the designed neural networks. This data can be found on Kaggle website, which contains selected data from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) II database, on which additional pre-filtering and validating were applied.

Measurements in MIMIC II were performed between 2001 and 2008 at Beth Israel Deaconess Medical Center (BIDMC) in Boston. This database contains many clinical data and physiological waveforms. However, the required data in terms of this study were the ECG, PPG, and blood pressure waveforms. These waveforms were collected from bedside monitors (Philips Intellivue MP70 Patient Monitor) at a sampling rate of 125 Hz. It is important to emphasize that blood pressure was measured invasively, therefore the most accurate blood pressure measurement method was used. It is also an important information that the PPG was measured on the fingertip.

So the dataset, found on Kaggle, contains PPG, blood pressure, and the signal from the one channel of the ECG that is least noisy and most similar to the expected waveform with the notable waves and peaks. These data total 5.28 Gbytes, which can be downloaded in .mat or .csv format, of which I chose the latter since I worked in Python throughout my work.

Then I converted the database into a suitable form for training, validating, and testing. To do this, I divided each waveform (PPG, ECG, BP) into 8-second time frames with a 2-second shift between the first element of adjacent frames, so there is an overlap of

6 seconds between these frames. Then I determined the systolic (SBP) and diastolic blood pressure (DBP) values from the blood pressure (BP) waveform as the maximum (SBP) and minimum (DBP) blood pressure values of the given time frame. Furthermore, I deleted the frames that had outlier systolic or diastolic values or were associated with an unrealistically low or high heart rate. After that, I put approximately 85% of the data into the training dataset and the remaining approximately 7.5-7.5% into the validation and testing dataset. During this sorting, I paid special attention to ensure that there was no overlap between the individual data sets, which is why I had to leave out a few time frames.

From such a frame containing 8 seconds of ECG, PPG, and the information of the SBP and DBP, I received a total of 58,234 pieces, of which 48,754 belong to the training data, 4,404 to the validation data, and 5,076 to the test data. This large amount of data provides the opportunity for proper training of neural networks.

6.1.2 Own measurement

I performed measurements using the MAX86150EVSYS device and the associated client application. Furthermore, for validation, I used an OMRON M2 automatic blood pressure measuring device, which can measure the systolic and diastolic blood pressure values with an accuracy of ± 3 mmHg. This device is shown in Figure 6.1.

Figure 6.1: Omron M2 [17]

Two subjects participated in the measurements. The cuff of the validating device was placed on the left upper arm of the subject, according to the description attached to the device. The PPG sensor on the MAX6150EVSYS device was touched by the index finger of the right hand, while the ECG electrodes were clamped between the middle finger and thumb of both hands.

A measurement lasted from the beginning of cuff inflation to the end of cuff deflation, during which the ECG and PPG waveforms were saved using MAX86150EVSYS's client application. The acquired waveforms of these measurements were supplemented with the systolic and diastolic values displayed by the Omron M2 for the given measurement. Using this method, I recorded 16 measurements, of which I kept 12, as the signals were not properly recorded during the deleted measurements due to disturbing factors. The following additional information were documented for all measurements:

- Subject ID

- Start time of the measurement
- Stop time of the measurement
- Result 1: Systolic blood pressure
- Result 2: Diastolic blood pressure
- Result 3: Heart rate
- Disturbance during the measurement, which received the following values: "None", "Minimal", and "Significant".

I also separated the signals obtained with the previously-mentioned procedure into 8-second time windows, however, to increase the number of received frames, I did not use a 2-second shift between the frames, but 0.4 seconds, which means 50 samples at a sampling frequency of 200 Hz. Each frame contains 8 seconds of PPG and ECG waveforms, as well as the corresponding systolic and diastolic values. The other registered data are not included in the frames.

I checked the appropriateness of the time frames obtained one by one and deleted those that did not prove to be appropriate. As a result, I received a total of 1621 time frames, of which 568 belong to one subject, while 1053 belong to the other subject.

6.2 Metrics

I evaluated the output of the prepared and trained neural networks on test data according to several methods, with the help of which the different networks become comparable. For the metrics, the error is interpreted as shown by Equation 6.1. However, in the case of the first two metric (Staistical and BHS), the absolute value of this error is used.

$$Error = Output_{target} - Output_{estimated} \tag{6.1}$$

Statistical In the case of this metric, the mean value and standard deviation of the previously described absolute error are calculated. A model can be considered a strong predictor if these two statistical values of the absolute error values are small.

British Hypertension Society (BHS) standard This metric applies the classification system published by the British Hypertension Society (BHS) that is commonly used to characterize the performance of blood pressure estimation algorithms. [68] In this case, the algorithms are classified based on the proportion of the outputs that have an error corresponding to a given interval. The requirements and rules are shown in Table 6.1, with the cumulative percentage values.

Grades	$\leq 5\ mmHg$	$\leq 10\ mmHg$	$\leq 15\ mmHg$
Grade A	60%	85%	95%
Grade B	50%	75%	90%
Grade C	40%	65%	85%

Table 6.1: BHS Standard Minimum Requirements [68]

Association for the Advancement of Medical Instrumentation (AAMI) standard The last metric is the standard published by the US Association for the Advancement of Medical Instrumentation(AAMI). Based on this method, a measurement algorithm is valid if the absolute value of the mean error is less than 5 mmHg and the standard deviation of the errors is less than 8 mmHg. [97]

6.3 Training

The labels used for the training process were the SBP and DBP values for the given time frame according to both presented data sets.

To train the models presented in Section 4.2, I utilized the AdamW optimizer using a Python library called Pytorch [19], which contains a function called AdamW [5]. I set the learning rate of this optimizer to 0.0001 and left all other parameters at their default values, which are as follows: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, and $weight\_decay = 0.01$.

As a loss function, I operated mean square error, which is commonly used in regression problems. I implemented this criterion with MSELoss() [14] function, which also can be found in the Pytorch library.

Furthermore, during training, I set the batch size to 24 by tuning this hyperparameter. I found that the network trained with this parameter gave the fewest estimation errors evaluated on the validation and test data.

I completed the training with epochs[1]. After each epoch, I evaluated the network on the validation data. If the average of the errors received after evaluating the batches was the lowest so far, I saved the result as the best state of the network.

6.4 Network results

In this section, the performance of the neural network models, described in Section 4.2, are evaluated on the publicly available dataset based on the metrics explained in Section 6.2. Then the chosen model (Model 2) is also evaluated on the measured dataset.

6.4.1 Result on the public dataset

Here, not only the results of the neural networks are presented, but also based on the metrics, they are compared with the results of the procedures presented in Chapter 3.

In the case of Section 3.1, I only mention the results evaluated on the MIMIC II database in the comparison, since the results of the other procedures were also evaluated on this. While in the case of Section 3.2, the best-performing network result is shown only, which was also evaluated on all the data obtained from the MIMIC II database.

Statistical Based on this metric, the neural networks presented in Section 3.1 and Section 3.2 produced results, which can be observed in Table 6.2 alongside my results. In Table 6.2, MAE is the average mean error and SD is the standard deviation.

[1]Epoch means a training cycle, during which all of the training data are used once to train our network. The complete training of a network usually takes place over several epochs.

Model	Systolic Blood Pressure MAE	SD	Diastolic Blood Pressure MAE	SD
PPG-based (3.1)	3.70	3.07	2.02	1.76
Model (b) (3.2)	6.73	14.51	2.52	6.44
Model 1 (4.2)	3.11	3.43	1.74	1.93
Model 2 (4.2)	2.85	3.70	1.62	2.10
Model 3 (4.2)	2.87	3.71	1.80	2.12

Table 6.2: Evaluation with the statistical metric (units are measured in mmHg)

According to the results shown in Table 6.2, my neural network models outperform Model (b) based on this metric. While the results of the PPG-based model are similar to mine as this model outperforms my results in terms of standard deviation, and my models outperform the PPG-based model in terms of mean values.

The models I have created give very similar results. While Model 1 can be called the best in terms of the standard deviation, the other two models are ahead of it in terms of the mean value. However, among my models, Model 2 can be considered a compromise in terms of this metric since it is not the worst for any of the characteristics.

BHS standard Similar to the previously-presented comparison, I compared the results of the different approaches in a table, which can be seen in Table 6.3. In this table, the given letter in brackets denotes the class according to the given percentage, SBP denotes the systolic blood pressure, and DBP is the diastolic blood pressure.

Model	Signal	$\leq 5\ mmHg$	$\leq 10\ mmHg$	$\leq 15\ mmHg$
PPG-based (3.1)	SBP	77%(A)	92%(A)	96%(A)
	DBP	93%(A)	97%(A)	99%(A)
Model (b) (3.2)	SBP	59.46%(B)	79.97%(B)	88.45%(B)
	DBP	76.95%(A)	95.72%(A)	99.97%(A)
Model 1 (4.2)	SBP	83.65%(A)	95.69%(A)	98.61%(A)
	DBP	94.48%(A)	99.02%(A)	99.89%(A)
Model 2 (4.2)	SBP	87.21%(A)	95.78%(A)	98.52%(A)
	DBP	95.00%(A)	98.75%(A)	99.68%(A)
Model 3 (4.2)	SBP	86.12%(A)	96.07%(A)	98.34%(A)
	DBP	93.21%(A)	98.66%(A)	99.77%(A)

Table 6.3: Performance according to British Hypertension Society classification

As Table 6.3 shows, in terms of the systolic blood pressure value, based on this metric, the results of my models outperform the other approaches. However, in the case of the diastolic values, looking at the largest interval, the results are very similar, however, Model (b) slightly exceeds my results. In the case of the PPG-based model, this result is not known precisely enough to determine how it relates to the other results. However, examining the other diastolic intervals, my approaches outperforms the other models.

Examining the models I created, it can be noticed that they show very similar results for the two larger intervals. While for the smallest interval, Model 2 shows the best results, both for SBP and DBP.

AAMI standard According to this metric, I compared the results of the different approaches in Table 6.4, in which ME means the mean error and SD means the standard deviation.

Model	Systolic Blood Pressure |ME|	Systolic Blood Pressure SD	Diastolic Blood Pressure |ME|	Diastolic Blood Pressure SD
PPG-based (3.1)	0.21	6.27	0.24	3.40
Model (b) (3.2)	4.64	14.51	3.16	6.44
Model 1 (4.2)	0.30	4.63	0.17	2.58
Model 2 (4.2)	0.14	4.66	0.12	2.65
Model 3 (4.2)	0.09	4.69	0.70	2.69

Table 6.4: Evaluation with AAMI metric (units are measured in mmHg)

It is clear from Table 6.4 that the PPG-based model and my models fulfill the AAMI criteria for all values. However, in the case of Model (b), the standard deviation of the systolic blood pressure value is greater than 8, so it does not meet this criterion.

Furthermore, the PPG-based model is worse than my models in terms of standard deviation but exceeds some models in terms of the mean value. The models I made are very similar in terms of standard deviation. However, in terms of mean value, Model 2 and Model 3 are outstanding. Furthermore, it is worth mentioning that Model 2 exceeds the PPG-based approach in terms of all characteristics.

Visualization of the results The results of the selected model (Model 2), evaluated on the test data, can be visualized in several ways. In the following, the results obtained with these methods are presented.

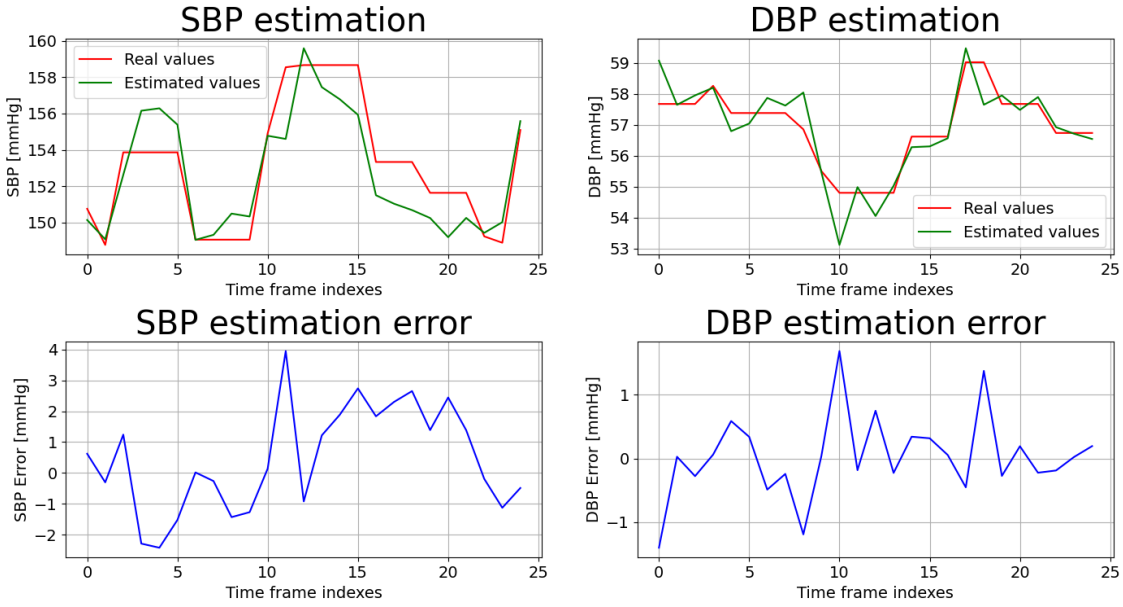

Figure 6.2: Comparison of estimated and real BP values

In Figure 6.2, the estimated values are visually compared with the genuine values presented on 25 adjacent time frames of the test dataset. On the left side, information according to the systolic blood pressure value (SBP) can be observed, while on the right, information is about the diastolic blood pressure values (DBP). In the upper plot, the real and estimated values are displayed at the same time, while in the lower plot, the error values, calculated from the difference between the real and estimated values, are shown. Each value refers to a specific time frame in the test data set.

The following visualization tool is the histogram, which approximates the distribution of the errors. Figure 6.3 was constructed based on the estimation error of systolic (SBP) and diastolic blood pressure (DBP), which I calculated as the difference between the true blood pressure value and the estimated blood pressure value.

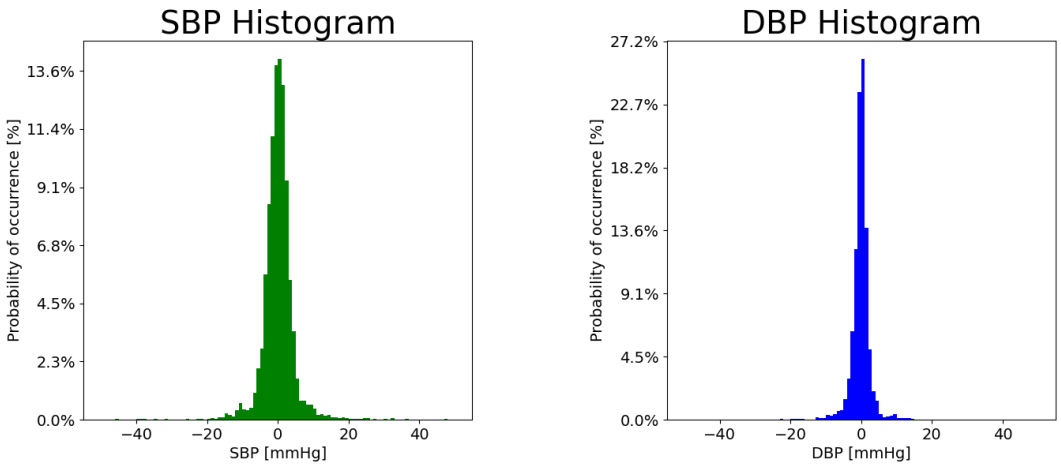

Figure 6.3: Histogram of blood pressure estimation errors

Figure 6.3 shows that both histograms approximate a normal distribution with zero mean value, which is 0.14 $mmHg$ for SBP and -0.12 $mmHg$ for DBP. The variance of this distribution can also be determined, which is 21.73 $mmHg^2$ for SBP and 7.04 $mmHg^2$ for DBP.

6.4.2 Results on self-measured dataset

The selected model (Model 2) was evaluated on the dataset obtained by my measurements. During this process, I used the same metrics and visualization tools as in the case of the evaluation of the publicly available dataset.

The results presented here were achieved with the help of transfer learning, during which I continued the training process with my self-measured data on the model pre-trained on the public dataset.

I performed the training using two types of training, validation, and test datasets. In the first case, which is called in the following the Normal dataset, I used 80% of each time frame in the training dataset, and 10-10% in the validation and test dataset. In the case of this extraction, I paid attention to ensuring that there was no overlap between the datasets. As a result, 925 time frames were included in the training dataset, 131 in the validation dataset, and 125 in the test dataset. So, in the case of these datasets, the data of both subjects can be found in each dataset. In the other case, I performed a so-called cross-validation, during which the training and validation dataset includes the frames that belong to the subject, who has more (1053) frames. And I perform the test on all the time frames belonging to the subject with 568 time frames. With this method, the training process is completely separated from testing, thus it provides useful information about the generalization ability of the resulting network. As a result, the training dataset contains 847, the validation dataset 206, and the test dataset 568 frames. In the case of this procedure, no overlapping can occur between the test and the other data sets, but it is possible between the training and validation datasets. To prevent data loss due to the overlap between frames, a completely separate measurement was added to the validation

dataset, while the results of all the other measurements on the same subject were added to the training dataset.

Statistical Here, the results of the selected model using the two types of datasets according to this metric are presented. This evaluation can be observed in Table 6.5, where MAE is the average mean error, and SD is the standard deviation.

	Systolic Blood Pressure		Diastolic Blood Pressure	
Model	MAE	SD	MAE	SD
Model 2 trained on Normal dataset	1.72	1.37	1.85	2.41
Model 2 trained with cross-validation	3.66	1.80	4.99	3.93

Table 6.5: Evaluation with the statistical metric (units are measured in mmHg)

As expected, the results obtained from the Normal dataset exceed the results of the cross-validation. In the case of several characteristics, these results exceed the results of the models evaluated on the public dataset. A significant cause of this is that I worked with orders of magnitude less data in the case of these evaluations.

BHS standard Here, similarly to the previous metric, I evaluate the model trained on the two types of data sets based on the standard defined by BHS. Table 6.6 shows the results of this evaluation, in which the given letter in brackets denotes the class according to the given percentage, SBP is the systolic blood pressure, and DBP is the diastolic blood pressure.

Model	Signal	$\leq 5\ mmHg$	$\leq 10\ mmHg$	$\leq 15\ mmHg$
Model 2 trained on Normal dataset	SBP	97.60%(A)	100.00%(A)	100.00%(A)
	DBP	92.80%(A)	100.00%(A)	100.00%(A)
Model 2 trained with cross-validation	SBP	70.07%(A)	100.00%(A)	100.00%(A)
	DBP	51.58%(B)	80.11%(B)	100.00%(A)

Table 6.6: Performance according to British Hypertension Society classification

As Table 6.6 shows, in the case of the Normal dataset, the obtained results significantly exceed the minimum limits of the best (A) category according to all intervals. In the case of cross-validation, the results for the systolic values and the largest interval of DBP reach grade A, but the other two intervals of diastolic values can only satisfy the requirements for grade B.

In terms of this metric, it is also worth mentioning that the high results obtained are significantly caused by the small amount of data.

AAMI standard Similar to the previously detailed metrics, I present the results of the chosen model on different datasets based on this metric, which can be seen in Table 6.7, where ME is the mean error and SD represents the standard deviation.

Based on the values shown in Table 6.7, the criteria of this metric are met by both approaches, since each absolute value of the mean error is less than 5 mmHg and each standard deviation of the errors is less than 8 mmHg. Although it is worth mentioning that while the result according to the Normal dataset significantly exceeds these requirements, in the case of cross-validation the results are close to the limits.

Model	Systolic Blood Pressure |ME|	Systolic Blood Pressure SD	Diastolic Blood Pressure |ME|	Diastolic Blood Pressure SD
Model 2 trained on Normal dataset	0.58	2.12	0.66	2.96
Model 2 trained with cross-validation	3.65	1.81	4.96	3.96

Table 6.7: Evaluation with AAMI metric (units are measured in mmHg)

Visualization of the results of Model 2 trained on the Normal dataset First, I visually compare the estimated and real blood pressure values and show the estimation error according to these values, as I did during the evaluation of the public dataset.

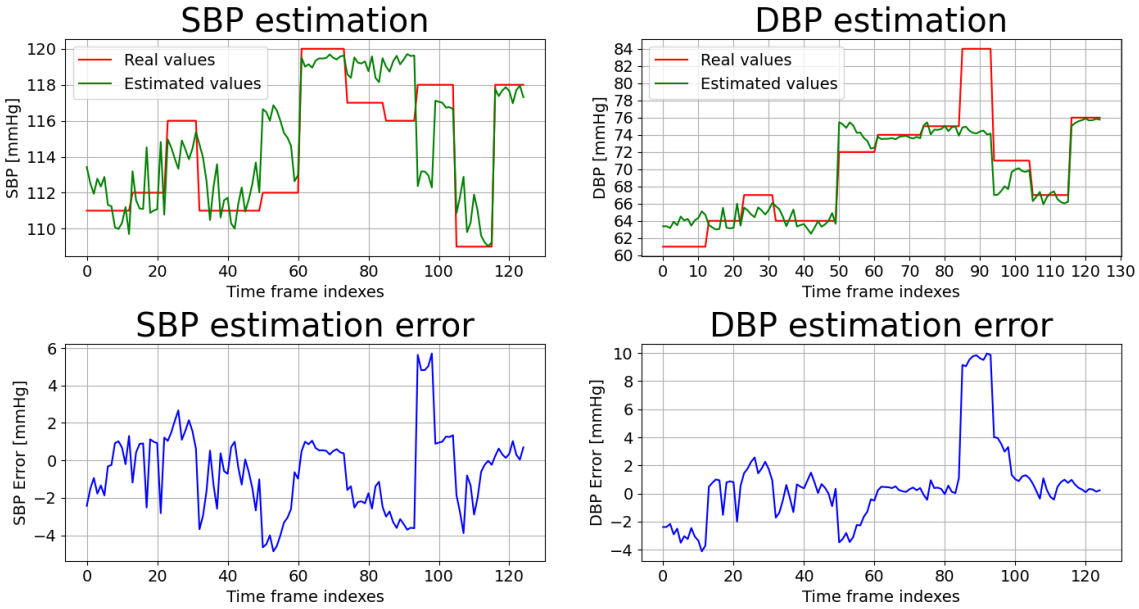

Figure 6.4: Comparison of estimated and real BP values

Figure 6.4 shows this visualization, which contains the estimated values for each time frame in the test dataset due to the small amount of data.

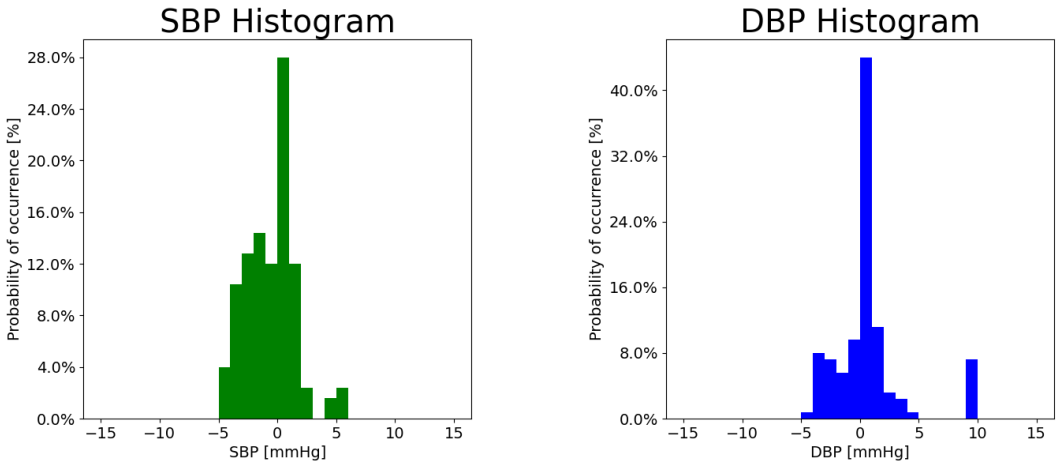

Figure 6.5: Histogram of blood pressure estimation errors

The following visualization tool is the histogram, based on the estimation error of systolic (SBP) and diastolic blood pressure (DBP), which are calculated as the difference between

the true blood pressure value and the estimated blood pressure value. The histograms are shown in Figure 6.5.

It is clear from this histogram that for such a small amount of data, it does not yet approximate the normal distribution as it did in the case of the public database.

Visualization of the results of Model 2 trained with cross-validation Here, the same visualization tools were used as in the case of Model 2, which was trained on the Normal dataset. Figure 6.6 shows the comparison of the real and the estimated blood pressure values for the time frames in the total test data set since the number of our data is small.

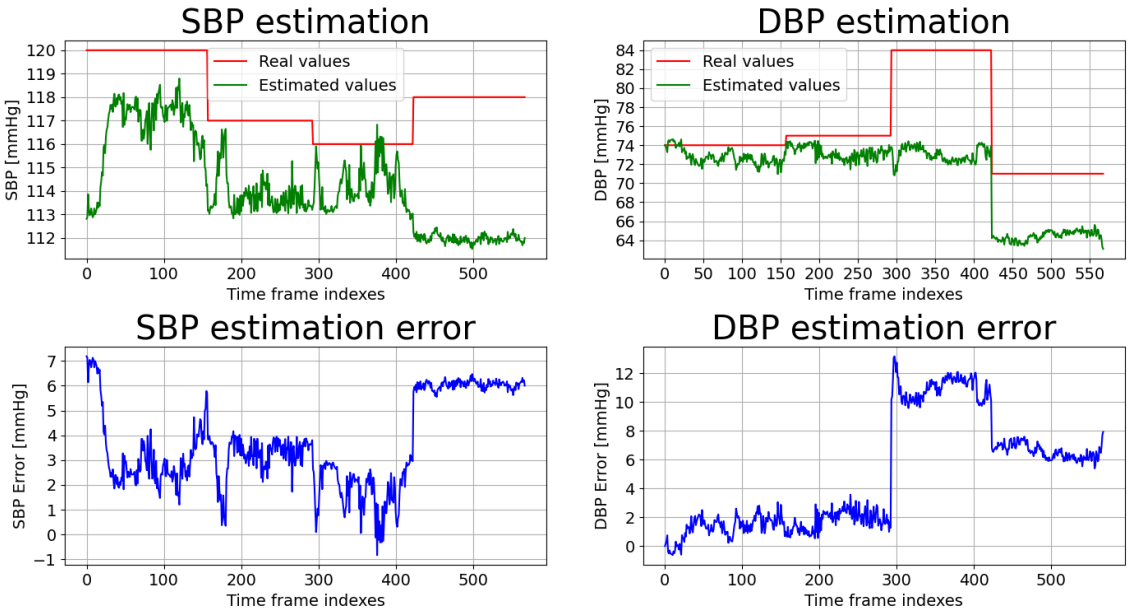

Figure 6.6: Comparison of estimated and real BP values

The following visualization method is the histogram, shown in Figure 6.7, which presents the distribution of the estimation errors.

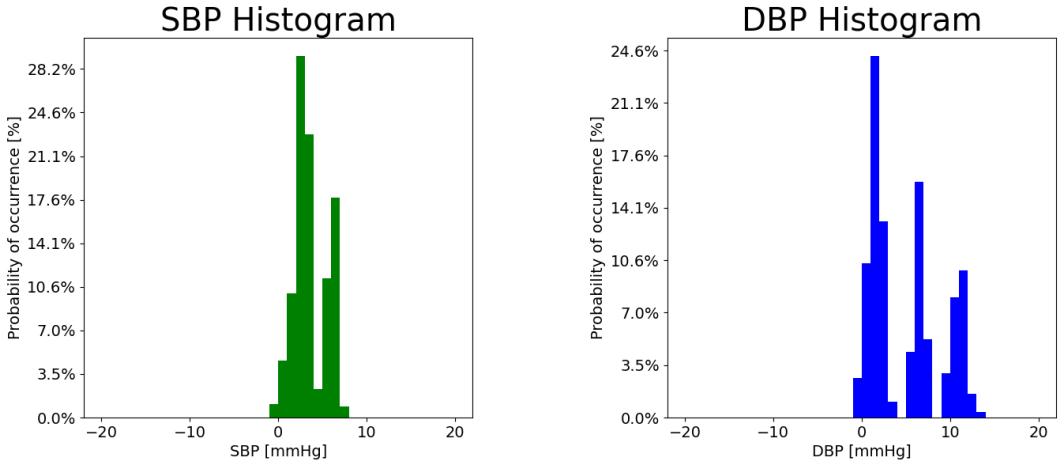

Figure 6.7: Histogram of blood pressure estimation errors

It is worth mentioning that according to both visualization methods, the means of the errors are far from zero and most of the errors are positive, so this model underestimates

the true values in general. This information matches the mean values presented in the table of the AAMI standard.

Conclusion Based on the previously presented information, the applicability of Model 2 can be evaluated in the case of my measurements. The selected model proved to be adequate for the Normal dataset, however, the results for cross-validation are also outstanding. Although in the latter case the model was not able to reach the limits of the best class in terms of all the intervals according to the BHS standard. However, the model was able to acquire a significant generalization ability, with the help of which in some cases it was able to determine the blood pressure of a completely unknown person with a sufficiently small estimation error, and in the worst case, it did not make an error of more than 15 mmHg.

Considering this information, I concluded that the version of Model 2 pre-trained on the public dataset and then further trained on the Normal dataset should be included in the implemented system. The strengths of this model are that thanks to the public dataset, it has a significant generalization ability, and thanks to transfer learning and my measurements, it was able to learn the characteristics of the measuring device. Therefore, this model can estimate the blood pressure values with a small estimation error from the pre-processed version of the data sent by MAX81650EVSYS.

6.5 Real-time results and limitations

This section presents the final state of the implemented system. Furthermore, it details the performance of the most critical part of the system and how it behaves in the case of significantly different inputs from the previously used dataset.

Real-time results The runtime of a neural network is always critical, especially in the present case, since the model used in the implemented system runs on the CPU of an embedded system, while these algorithms usually run on high-performance GPUs. So, it is worth examining how much time my neural network needs to produce an output for a given input on the central unit of the implemented system, which is the Raspberry Pi 4 Model B.

To investigate this, I measured the runtime of the network in the final state of the system in 133 cases. The slowest run was 712 ms, while the fastest was 219 ms. The average value of the measurements was 444 ms, while the standard deviation was 114 ms.

Limitations The subjects were at rest during the measurement of the data, which forms the Normal dataset. Therefore, I wanted to investigate how the network would perform on high blood pressure data, by which I mean a value of around 150 mmHg or higher for SBP. The obtained results showed that the chosen model is unable to estimate these values with sufficiently small estimation errors. However, if I further train the network with these data, then the BHS standard's criteria for class A can be met. Therefore, this problem could be solved if more data were available that were measured with the given measuring device.

Furthermore, in the case of very high pressure and heart rate, due to the design of the MAX86150EVSYS, it is difficult to record proper waveforms since the PPG sensor is very sensitive and in this physiological state, the subjects cannot touch the sensor in an

appropriate stable way because of their increased breathing. This problem could be solved by making a device that provides stable contact with the sensors.

Final state of the implemented system The use of the implemented system is shown in Figure 6.8, on which the Omron M2 blood pressure measuring device used for validation can be observed, and the cuff of which was placed on my left upper arm when the picture was taken. On the right side of Omron M2 is the Raspberry Pi 4 Model B in a black case connected to the monitor with the white HDMI cable shown in the picture. To the right of the Raspberry is the MAX86150EVSYS, which clearly shows the use of the soldered electrodes and the PPG sensor with my right index finger.

Figure 6.8: Usage of the implemented system

An enlarged and sharpened version of the GUI is displayed in the upper right corner of Figure 6.8 for better visibility. Based on this image, the results of Omron can be compared with my results.

Chapter 7

Conclusions and future work

Blood pressure is one of the vital signals, the measurement of which is essential because an inadequate value of this signal can cause serious consequences. Nowadays, it is most often measured using a cuff, which has significant disadvantages, e.g., that it is uncomfortable, it cannot measure continuously, because of this temporary hypertension can be unrecognized, and the results of which can be influenced by psychological effects. If an accurate and continuous measurement is required, invasive blood pressure measurement is used, however, this method can only be performed under medical supervision.

Within the framework of this study, I presented a novel deep learning based method utilizing ECG and PPG signals for measuring blood pressure. This method offers the possibility of continuous and convenient measurement with sufficiently small estimation errors. To implement this task, I created a complete framework that can sample and display data in addition to performing calculations. I trained and tested the neural network model, which runs on the core unit of the system, on a publicly available dataset, as well as on my self-measured dataset. Then I evaluated the obtained results using various metrics and visualization methods.

Conclusions This study aimed to create a system that can continuously estimate the current systolic and diastolic blood pressure values with a small estimation error based on externally measurable input signals. The central task of the preparation was to implement a suitable estimation algorithm that could run efficiently on an embedded system.

The results of this study transcended my expectations, as my chosen model managed to exceed the results of the procedures described in the presented publications according to several respects. Furthermore, it was achieved that the algorithm runs in an embedded environment with a proper runtime. In addition, I supplemented the given task with heart rate computation to create a more useful and versatile system.

Future work There are several opportunities for further development of the current work. First of all, the hyperparameters of the neural network could be further tuned, which is an option to improve all tasks using neural networks.

The accuracy of the neural network running on the embedded system could also be increased if more labeled data, measured by MAX81650, were available since the neural network could learn more essential features from more data. I could not make more measurements due to time constraints, but it offers an excellent opportunity for further development.

Furthermore, creating an own hardware unit that is more user-friendly than the MAX86150EVSYS, would be a significant enhancement. Although, the measuring board used in this study performed its task adequately. However, the simultaneous usability of the ECG and PPG sensor could have been easier for the users. In addition, since the ECG input signal filter circuit is right next to the PPG sensor, that circuit could easily be touched with a finger due to the absence of an enclosure, which would completely spoil the ECG waveform if touched during measurement. Therefore, with self-created hardware, it would be possible to implement an easy-to-use or even wearable device.

The knowledge and results acquired during the work provide an excellent basis for the further development of the task. Furthermore, the implemented system and its performance serve as proof of the concept I aimed to implement.

Acknowledgements

I would like to thank Szabolcs Torma for spending a lot of time and energy helping me with my study, for his contribution to every step of my task with his helpful pieces of advice, and for the excellent atmosphere he created during the preparation of the current work.

I would like to thank Dr. Luca Szegletes for introducing me to the world of deep learning and helping me get started with the topic of this study. Furthermore, I would like to say a special thanks for supervising and helping the progress of this work, as well as my mental health.

I would like to thank Dr. Ákos Nagy for helping me choose the right topic, as well as for helping me figure out the structure of the implemented system, and also for helping me select and obtain the necessary components. I am also grateful that he helped me as soon as he could in case of any problem.

I would like to thank dr. Zsombor Mátyás Papp, M.D., who graduated from the Faculty of Medicine at Semmelweis University, and Domonkos Mike, who is a 4th-year student at the Faculty of Medicine at Semmelweis University, for reviewing the sections related to their fields of expertise and helping to clarify these parts of my study.

I would like to thank Márton Jász, who graduated from The University of Manchester, for reviewing the linguistic correctness of this study and improving it with his suggestions.

I would like to thank Bertalan Pimper, who made Raspberry Pi 4 Model B available for me at the right time, thereby he greatly contributed to the completion of this study before the deadline.

Bibliography

[1] A. (n.d.). cortex-m4 – arm®. arm. URL https://www.arm.com/products/silicon-ip-cpu/cortex-m/cortex-m4. [Accessed October 31, 2022].

[2] Max86150evsys evaluation system for the max86150: Maxim integrated. evaluation system for the max86150 | maxim integrated. . URL https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX86150EVSYS.html/design-development/tb-tabs-4. [Accessed October 31, 2022].

[3] Max86150 evaluation system, . URL https://datasheets.maximintegrated.com/en/ds/MAX86150EVSYS.pdf. [Accessed October 31, 2022].

[4] Max86150 integrated photoplethysmogram and electrocardiogram bio-sensor module for mobile health, . URL https://datasheets.maximintegrated.com/en/ds/MAX86150.pdf. [Accessed October 31, 2022].

[5] Adamw. URL https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html. [Accessed October 31, 2022].

[6] Blood pressure. URL https://www.physio-pedia.com/Blood_Pressure. [Accessed October 31, 2022].

[7] Scipy.signal.butter. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html. [Accessed October 31, 2022].

[8] Check blood pressure in both arms. URL http://www.meddean.luc.edu/lumen/meded/medicine/pulmonar/pd/pstep6.htm. [Accessed October 31, 2022].

[9] Cardiovascular system. URL https://www.physio-pedia.com/Cardiovascular_System. [Accessed October 31, 2022].

[10] Scipy.ndimage.convolve1d. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.convolve1d.html. [Accessed October 31, 2022].

[11] Global, regional, and national comparative risk assessment of 84 behavioural, environmental and occupational, and metabolic risks or clusters of risks for 195 countries and territories, 1990-2017: A systematic analysis for the global burden of disease study 2017. URL https://pubmed.ncbi.nlm.nih.gov/30496105/.

[12] Scipy.signal.filtfilt. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html. [Accessed October 31, 2022].

[13] Overview-high blood pressure (hypertension). URL https://www.nhs.uk/conditions/high-blood-pressure-hypertension/. [Accessed October 31, 2022].

[14] Mseloss. URL https://pytorch.org/docs/stable/generated/torch.nn.MSELoss.html. [Accessed October 31, 2022].

[15] Numpy.array. URL https://numpy.org/doc/stable/reference/generated/numpy.array.html. [Accessed October 31, 2022].

[16] Numpy. URL https://numpy.org/. [Accessed October 31, 2022].

[17] Omron m2 intellisense felkaros vérnyomásmérő. URL https://www.pulzusmeres.hu/OMRON-M2-Intellisense-felkaros-vernyomasmero. [Accessed October 31, 2022].

[18] Python, . URL https://www.python.org/about/. [Accessed October 31, 2022].

[19] Pytorch, . URL https://pytorch.org/. [Accessed October 31, 2022].

[20] Scipy.signal.resample. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample.html. [Accessed October 31, 2022].

[21] Scipy.signal.savgol_filter. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.savgol_filter.html. [Accessed October 31, 2022].

[22] Scipy documentation. URL https://docs.scipy.org/doc/scipy/index.html. [Accessed October 31, 2022].

[23] Threading - thread-based parallelism. URL https://docs.python.org/3/library/threading.html. [Accessed October 31, 2022].

[24] Tkinter - python interface to tcl/tk. URL https://docs.python.org/3/library/tkinter.html. [Accessed October 31, 2022].

[25] Vital signs (body temperature, pulse rate, respiration rate, blood pressure). URL https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentTypeID=85&ContentID=P00866. [Accessed October 31, 2022].

[26] High blood pressure symptoms and causes, May 2021. URL https://www.cdc.gov/bloodpressure/about.htm.

[27] High blood pressure (hypertension), Sep 2022. URL https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/symptoms-causes/syc-20373410.

[28] Low blood pressure (hypotension), May 2022. URL https://www.mayoclinic.org/diseases-conditions/low-blood-pressure/symptoms-causes/syc-20355465.

[29] What are the symptoms of high blood pressure?, September 2022. URL https://www.heart.org/en/health-topics/high-blood-pressure/why-high-blood-pressure-is-a-silent-killer/what-are-the-symptoms-of-high-blood-pressure.

[30] Max86150 integrated photoplethysmogram and electrocardiogram bio-sensor module for mobile health: Maxim integrated. integrated photoplethysmogram and electrocardiogram bio-sensor module for mobile health | maxim integrated. 2022.09.30. URL https://www.maximintegrated.com/en/products/interface/signal-integrity/MAX86150.html/product-details/tabs-4. [Accessed October 31, 2022].

[31] Invasive blood pressure, Accessed October 31, 2022. URL http://www.memscap.com/applications-and-market-segments/medical-and-biomedical/invasive-blood-pressure.

[32] U. Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals. *Computers in Biology and Medicine*, 100:270–278, 2018. ISSN 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2017.09.017. URL https://www.sciencedirect.com/science/article/pii/S0010482517303153.

[33] Baeldung. How relu and dropout layers work in cnns, May 2022. URL https://www.baeldung.com/cs/ml-relu-dropout-layers.

[34] Marius Reto Bigler, Patrick Zimmermann, Athanasios Papadis, and Christian Seiler. Accuracy of intracoronary ecg parameters for myocardial ischemia detection. *Journal of Electrocardiology*, 64:50–57, 2021. ISSN 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2020.11.018. URL https://www.sciencedirect.com/science/article/pii/S0022073620306130.

[35] Ed Burns and Kate Brush. What is deep learning and how does it work?, Mar 2021. URL https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network.

[36] Denisse Castaneda, Aibhlin Esparza, Mohammad Ghamari, Cinna Soltanpur, and Homer Nazeran. A review on wearable photoplethysmography sensors and their potential future applications in health care, 2018. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/.

[37] David Cohen and Uri Dinnar. Korotkoff sounds - a phenomenon associated with parametric instability of fluid filled elastic tube. In Dov Jaron, editor, *Proceedings of the Sixth New England Bioengineering Conference*, pages 33–36. Pergamon, 1978. ISBN 978-0-08-022678-1. DOI: https://doi.org/10.1016/B978-0-08-022678-1.50014-X. URL https://www.sciencedirect.com/science/article/pii/B978008022678150014X.

[38] DeepAI. Exploding gradient problem, May 2019. URL https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem.

[39] Hüsnü Değirmenci, Eftal Murat Bakirci, Murat Çakir, and Halil İbrahim Tanrıseven. Current approach to isolated diastolic hypertension, Dec 2020. URL https://www.peertechzpublications.com/articles/ACH-6-128.php#Copyright.

[40] Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring. The dawn of quantum natural language processing. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8612–8616, 2022. DOI: 10.1109/ICASSP43922.2022.9747675.

[41] Rian Dolphin. Lstm networks: A detailed explanation, Dec 2021. URL https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[42] By: IBM Cloud Education. What are recurrent neural networks? URL https://www.ibm.com/cloud/learn/recurrent-neural-networks. [Accessed October 31, 2022].

[43] Electrical4U. Butterworth filter: What is it? (design amp; applications), Apr 2021. URL https://www.electrical4u.com/butterworth-filter/.

[44] Xiaole Fan, Xiufang Feng, Yunyun Dong, and Huichao Hou. Covid-19 ct image recognition algorithm based on transformer and cnn. *Displays*, 72:102150, 2022. ISSN 0141-9382. DOI: https://doi.org/10.1016/j.displa.2022.102150. URL https://www.sciencedirect.com/science/article/pii/S0141938222000026.

[45] Adam Feather, David Randall, and Mona Waterhouse. chapter 30. Cardiology, page 1033–1038. Elsevier, 2020.

[46] Giancarlo Fortino and Valerio Giampà. Ppg-based methods for non invasive and continuous blood pressure measurement: an overview and development issues in body sensor networks. In *2010 IEEE International Workshop on Medical Measurements and Applications*, pages 10–13, 2010. DOI: 10.1109/MEMEA.2010.5480201.

[47] Robert M. Freund, Paul Grigas, and Rahul Mazumder. A new perspective on boosting in linear regression via subgradient optimization and relatives. *The Annals of Statistics*, 45(6):2328 – 2364, 2017. DOI: 10.1214/16-AOS1505. URL https://doi.org/10.1214/16-AOS1505.

[48] Chirag Goyal. Guide to prevent overfitting in neural networks, Jun 2021. URL https://www.analyticsvidhya.com/blog/2021/06/complete-guide-to-prevent-overfitting-in-neural-networks-part-1/.

[49] Bernard Grundlehner and Vojkan Mihajlović. Ambulatory eeg monitoring. In Roger Narayan, editor, *Encyclopedia of Biomedical Engineering*, pages 223–239. Elsevier, Oxford, 2019. ISBN 978-0-12-805144-3. DOI: https://doi.org/10.1016/B978-0-12-801238-3.10885-2. URL https://www.sciencedirect.com/science/article/pii/B9780128012383108852.

[50] Sifei Han, Robert F. Zhang, Lingyun Shi, Russell Richie, Haixia Liu, Andrew Tseng, Wei Quan, Neal Ryan, David Brent, and Fuchiang R. Tsui. Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing. *Journal of Biomedical Informatics*, 127:103984, 2022. ISSN 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2021.103984. URL https://www.sciencedirect.com/science/article/pii/S1532046421003130.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[52] Bin Huang, Weihai Chen, Chun-Liang Lin, Chia-Feng Juang, and Jianhua Wang. Mlp-bp: A novel framework for cuffless blood pressure measurement with ppg and ecg signals based on mlp-mixer neural networks. *Biomedical Signal Processing and Control*, 73:103404, 2022. ISSN 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.103404. URL https://www.sciencedirect.com/science/article/pii/S1746809421010016.

[53] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

[54] About Sakshi Gupta Sakshi is a Senior Associate Editor at Springboard. She is a technology enthusiast who loves to read, write about emerging tech. She is a content marketer, has experience working in the Indian, and US markets. Regression vs. classification in machine learning: What's the difference?, Jul 2022. URL https://www.springboard.com/blog/data-science/regression-vs-classification/.

[55] Mohamad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, and Mahdi Shabany. Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1006–1009, 2015. DOI: 10.1109/ISCAS.2015.7168806.

[56] Mohammad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, and Mahdi Shabany. Cuffless blood pressure estimation algorithms for continuous health-care monitoring. *IEEE Transactions on Biomedical Engineering*, 64(4):859–869, 2017. DOI: 10.1109/TBME.2016.2580904.

[57] Simeon Kostadinov. Understanding gru networks, Nov 2019. URL https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be.

[58] Simeon Kostadinov. Understanding backpropagation algorithm, Aug 2019. URL https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd.

[59] Panicos A. Kyriacou and John Allen. *Photoplethysmography: Technology, Signal Analysis and Applications*. Elsevier, 2021.

[60] Joon Lee, Daniel J Scott, Mauricio Villarroel, Gari D Clifford, Mohammed Saeed, and Roger G Mark. Open-access mimic-ii database for intensive care research, 2011. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6339457/.

[61] Yung-Hui Li, Latifa Nabila Harfiya, Kartika Purwandari, and Yue-Der Lin. Real-time cuffless continuous blood pressure estimation using deep learning model, Sep 2020. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7584036/?fbclid=IwAR2SAa4LNQ3mgsWDTOFsnj06BrCWoapoXKNRfImjx4LWioPidaNc4ZNOIpg.

[62] Zhiyuan Li, Qinmu Wu, Shengming Yang, and Xiangping Chen. Diagnosis of rotor demagnetization and eccentricity faults for ipmsm based on deep cnn and image recognition - complex amp; intelligent systems, May 2022. URL https://link.springer.com/article/10.1007/s40747-022-00764-z#citeas.

[63] Jiankun Liu, Hao-Min Cheng, Chen-Huan Chen, Shih-Hsien Sung, Mohsen Moslehpour, Jin-Oh Hahn, and Ramakrishna Mukkamala. Patient-specific oscillometric blood pressure measurement, Jun 2016. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4907878/.

[64] Wenyu Liu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. Image-adaptive yolo for object detection in adverse weather conditions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2):1792–1800, Jun. 2022. DOI: 10.1609/aaai.v36i2.20072. URL https://ojs.aaai.org/index.php/AAAI/article/view/20072.

[65] Pavel Lyakhov, Mariya Kiladze, and Ulyana Lyakhova. System for neural network determination of atrial fibrillation on ecg signals with wavelet-based preprocessing. *Applied Sciences*, 11(16), 2021. ISSN 2076-3417. DOI: 10.3390/app11167213. URL https://www.mdpi.com/2076-3417/11/16/7213.

[66] Arnab Kumar Mishra, Sujit Kumar Das, Pinki Roy, and Sivaji Bandyopadhyay. Identifying covid19 from chest ct images: A deep convolutional neural networks based approach, Aug 2020. URL https://www.hindawi.com/journals/jhe/2020/8843664/.

[67] Khan Muhammad, Salman Khan, Javier Del Ser, and Victor Hugo C. de Albuquerque. Deep learning for multigrade brain tumor classification in smart healthcare systems: A prospective survey. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):507–522, 2021. DOI: 10.1109/TNNLS.2020.2995800.

[68] E O'Brien, B Waeber, G Parati, J Staessen, and M G Myers. Blood pressure measuring devices: Recommendations of the european society of hypertension, Mar 2001. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1119736/.

[69] Gbenga Ogedegbe and Thomas Pickering. Principles and techniques of blood pressure measurement, Nov 2010. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3639494/.

[70] Bonnafoux P;. Auscultatory and oscillometric methods of ambulatory blood pressure monitoring, advantages and limits: A technical point of view. URL https://pubmed.ncbi.nlm.nih.gov/10226223/.

[71] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985. DOI: 10.1109/TBME.1985.325532.

[72] Raspberry Pi. Raspberry pi 4 model b, . URL https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. [Accessed October 31, 2022].

[73] Raspberry Pi. Raspberry pi 4 model b specifications, . URL https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. [Accessed October 31, 2022].

[74] Mariana R Pioli, Alessandra Mv Ritter, Ana Paula de Faria, and Rodrigo Modolo. White coat syndrome and its variations: differences and clinical impact. *Integr Blood Press Control*, 11:73–79, November 2018.

[75] Paul-Louis Pröve. An introduction to different types of convolutions in deep learning, Feb 2018. URL https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d.

[76] Sruthi E R. Random forest: Introduction to random forest algorithm, Jun 2022. URL https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

[77] Florian Rader and Ronald G Victor. The slow evolution of blood pressure monitoring: But wait, not so fast! *JACC Basic Transl Sci*, 2(6):643–645, December 2017.

[78] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, Oct 2017. URL https://arxiv.org/abs/1710.05941v2.

[79] Kartikeya Rana. Pooling layer-beginner to intermediate, Feb 2022. URL https://ai.plainenglish.io/pooling-layer-beginner-to-intermediate-fa0dbdce80eb?gi=5e3b174eac88.

[80] Emna Rejaibi, Ali Komaty, Fabrice Meriaudeau, Said Agrebi, and Alice Othmani. Mfcc-based recurrent neural network for automatic clinical depression recognition and assessment from speech. *Biomedical Signal Processing and Control*, 71:103107, 2022. ISSN 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2021.103107. URL https://www.sciencedirect.com/science/article/pii/S1746809421007047.

[81] Arunabha M. Roy. Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for eeg mi multiclassification in brain–computer interface. *Engineering Applications of Artificial Intelligence*, 116:105347, 2022. ISSN 0952-1976. DOI: https://doi.org/10.1016/j.engappai.2022.105347. URL https://www.sciencedirect.com/science/article/pii/S0952197622003712.

[82] Liu D;Görges M;Jenkins SA;. University of queensland vital signs dataset: Development of an accessible repository of anesthesia patient monitoring data for research. URL https://pubmed.ncbi.nlm.nih.gov/22190558/.

[83] Sumit Saha. A comprehensive guide to convolutional neural networks-the eli5 way, Dec 2018. URL https://rb.gy/nxajwf.

[84] Anitej Banerjee says:, Math Vault says:, Pranjal says:, S.M. Zakaria Laskar says:, and Helene Says:. Chain rule for derivative - the theory, Sep 2020. URL https://mathvault.ca/chain-rule-derivative/.

[85] Ronald W. Schafer. What is a savitzky-golay filter? [lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011. DOI: 10.1109/MSP.2011.941097.

[86] Astrid Schneider, Gerhard Hommel, and Maria Blettner. Linear regression analysis: Part 14 of a series on evaluation of scientific publications, Nov 2010. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2992018/.

[87] Sagar Sharma. Activation functions in neural networks, Jul 2021. URL https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[88] M.D. Sheldon G. Sheps. Isolated systolic hypertension: A health concern?, Jun 2022. URL https://www.mayoclinic.org/diseases-conditions/high-blood-pressure/expert-answers/hypertension/faq-20058527.

[89] Sukriti Singh and Raghavan B. Sunoj. A transfer learning protocol for chemical catalysis using a recurrent neural network adapted from natural language processing. *Digital Discovery*, 1:303–312, 2022. DOI: 10.1039/D1DD00052G. URL http://dx.doi.org/10.1039/D1DD00052G.

[90] Audrey Spelde and Christopher Monahan. *Invasive Arterial Blood Pressure Monitoring*. McGraw-Hill Education, New York, NY, 2016. URL accessanesthesiology.mhmedical.com/content.aspx?aid=1135738099.

[91] Toshiyo Tamura. Current progress of photoplethysmography and spo2 for health monitoring, Feb 2019. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6431353/?fbclid=IwAR15C1dfgtjFexMneIqWWS5EUFjhflzRuHcdCNSLWSGsL_QEIDyuP64quRw.

[92] Ali Tazarv and Marco Levorato. A deep learning approach to predict blood pressure from ppg signals. In *2021 43rd Annual International Conference of the IEEE*

Engineering in Medicine Biology Society (EMBC), pages 5658–5662, 2021. DOI: 10.1109/EMBC46164.2021.9629687.

[93] Yingjie Tian, Duo Su, Stanislao Lauria, and Xiaohui Liu. Recent advances on loss functions in deep learning for computer vision. *Neurocomputing*, 497:129–158, 2022. ISSN 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2022.04.127. URL https://www.sciencedirect.com/science/article/pii/S0925231222005239.

[94] Diego Unzueta. Convolutional layers vs fully connected layers, Mar 2022. URL https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b.

[95] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762.

[96] Chi-Feng Wang. The vanishing gradient problem, Jan 2019. URL https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484.

[97] W B White, A S Berson, C Robbins, M J Jamieson, L M Prisant, E Roccella, and S G Sheps. National standard for measurement of resting and ambulatory blood pressures with automated sphygmomanometers. *Hypertension*, 21(4):504–509, 1993. DOI: 10.1161/01.HYP.21.4.504. URL https://www.ahajournals.org/doi/abs/10.1161/01.HYP.21.4.504.

[98] Wikipedia contributors. Heart rate — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Heart_rate&oldid=1114767737, 2022. [Online; accessed 9-October-2022].

[99] Wikipedia contributors. Electrocardiography — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Electrocardiography&oldid=1107952547, 2022. [Online; accessed 16-September-2022].

[100] Bryan Williams, Giuseppe Mancia, Wilko Spiering, Enrico Agabiti Rosei, Michel Azizi, Michel Burnier, Denis L Clement, Antonio Coca, Giovanni de Simone, Anna Dominiczak, Thomas Kahan, Felix Mahfoud, Josep Redon, Luis Ruilope, Alberto Zanchetti, Mary Kerins, Sverre E Kjeldsen, Reinhold Kreutz, Stephane Laurent, Gregory Y H Lip, Richard McManus, Krzysztof Narkiewicz, Frank Ruschitzka, Roland E Schmieder, Evgeny Shlyakhto, Costas Tsioufis, Victor Aboyans, Ileana Desormais, and ESC Scientific Document Group. 2018 ESC/ESH Guidelines for the management of arterial hypertension: The Task Force for the management of arterial hypertension of the European Society of Cardiology (ESC) and the European Society of Hypertension (ESH). *European Heart Journal*, 39(33):3021–3104, 08 2018. ISSN 0195-668X. DOI: 10.1093/eurheartj/ehy339. URL https://doi.org/10.1093/eurheartj/ehy339.

[101] Genshen Yan, Shen Liang, Yanchun Zhang, and Fan Liu. Fusing transformer model with temporal features for ecg heartbeat classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 898–905, 2019. DOI: 10.1109/BIBM47256.2019.8983326.

[102] Babak Zamanlooy and Mitra Mirhassani. Efficient vlsi implementation of neural networks with hyperbolic tangent activation function. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(1):39–48, 2014. DOI: 10.1109/TVLSI.2012.2232321.

[103] Gus Q. Zhang and Weiguo Zhang. Heart rate, lifespan, and mortality risk. *Ageing Research Reviews*, 8(1):52–60, 2009. ISSN 1568-1637. DOI: https://doi.org/10.1016/j.arr.2008.10.001. URL https://www.sciencedirect.com/science/article/pii/S1568163708000524.

[104] Yuan-ting Zhang, Carmen C.Y. Poon, Chun-hung Chan, Martin W.W. Tsang, and Kin-fai Wu. A health-shirt using e-textile materials for the continuous and cuffless monitoring of arterial blood pressure. In *2006 3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors*, pages 86–89, 2006. DOI: 10.1109/ISSMDBS.2006.360104.

[105] Ya Zhou and Xiaobo Jiao. Intelligent analysis system for signal processing tasks based on lstm recurrent neural network algorithm - neural computing and applications, Sep 2021. URL https://link.springer.com/article/10.1007/s00521-021-06478-6.

[106] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, Jun 2020. URL https://arxiv.org/abs/1911.02685.

[107] Enes Zvornicanin. Differences between bidirectional and unidirectional lstm, Feb 2022. URL https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm.