



ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIPARTIMENTO DI

INGEGNERIA DELL'ENERGIA ELETTRICA E DELL'INFORMAZIONE

"Guglielmo Marconi"

DEI

**CORSO DI LAUREA MAGISTRALE IN
INGEGNERIA ELETTRONICA**

TESI DI LAUREA

in

Hardware-Software Design of IoT Systems M

Blood Pressure Estimation from PPG for Wearable Devices: A Benchmark Study on Classical, Deep, and Transformer-Based Models

Relatore:

Chiar.mo Prof. Davide Rossi

Presentata da:

Margherita Rossi

Correlatori:

Prof. Simone Benatti

Pierangelo Maria Rapa

Mattia Orlandi

Sessione V

Anno Accademico 2023/2024

Alla mia famiglia

Ringraziamenti

Desidero ringraziare il mio relatore, Prof. Davide Rossi, e il mio correlatore, Prof. Simone Benatti, per avermi permesso di approfondire questo interessante argomento, per il tempo che mi hanno dedicato e i preziosi consigli ricevuti durante lo sviluppo di questo lavoro di tesi. Un sincero ringraziamento va anche ai miei correlatori, i dottorandi Pierangelo Rapa e Mattia Orlandi, per la guida che mi hanno fornito e per il loro costante supporto.

Ringrazio caldamente anche tutti i dottorandi del Micrel Lab per i loro consigli e per avermi accolta durante questi mesi.

Un sentito grazie anche ai miei amici e colleghi dell'università, con cui ho condiviso gli ultimi anni e senza i quali il mio percorso non sarebbe stato lo stesso.

Un ringraziamento speciale va infine ai miei amici e alla mia famiglia, per il loro continuo sostegno e l'incoraggiamento che non mi hanno mai fatto mancare.

Sommario

Il monitoraggio accurato della pressione arteriosa (PA) è fondamentale per la gestione e il controllo della salute cardiovascolare. I metodi tradizionali basati su bracciale, sebbene affidabili, sono poco pratici per il monitoraggio continuo.

Questo studio esplora un approccio non invasivo per la stima della PA utilizzando segnali di fotopleletismografia (PPG), sfruttando modelli di machine learning (ML) classici, deep learning (DL) e basati su transformer. Viene eseguita un'analisi comparativa su vari dataset per valutare l'efficacia di questi modelli.

Questa ricerca valuta metodi basati su features, come SVR, Decision Trees e Random Forests, insieme ad architetture profonde, inclusi modelli di reti CNN come CNN-LSTM, ResNet, UNet, ResUNet con attenzione, reti Transformer e Foundation Models.

I modelli sono verificati sul dataset VitalDB, ottenendo per SBP e DBP rispettivamente un MAE minimo di 5.33 e 3.42 mmHg nel caso di test con calibrazione e di 12.61 e 8.04 mmHg nel caso dell'approccio senza calibrazione.

I metodi proposti sono testati anche su dataset benchmark più piccoli per verificare come le prestazioni del modello cambiano quando applicato a dati provenienti da diverse fonti.

Dai risultati, si evince che l'accuratezza dei modelli dipende dalle condizioni sperimentali, tra cui il setup di misura, il dataset utilizzato e le modalità di acquisizione dei dati, quindi devono essere considerate le specifiche esigenze applicative e le caratteristiche dei dati quando si sceglie l'approccio per la stima della PA.

I risultati indicano che i modelli di DL migliorano l'accuratezza della previsione della PA, in particolare su grandi dataset, rispetto ai metodi ML tradizionali. Tuttavia, i metodi di ML ottengono risultati migliori su dataset più piccoli, evidenziando le difficoltà di addestrare adeguatamente i modelli di DL su dataset di dimensioni ridotte. Inoltre, quando vengono incorporati i segnali ECG insieme ai PPG, gli errori di previsione della PA si riducono notevolmente, evidenziando la possibilità di aumentare l'accuratezza mediante configurazioni multimodali e approcci di sensor fusion.

Nonostante i risultati siano promettenti, una soluzione esclusivamente algoritmica basata sul solo PPG o sulla sua combinazione con l'ECG non sembra garantire un livello di affidabilità adeguato per applicazioni cliniche o diagnostiche. Di conseguenza, un possibile sviluppo futuro di questo lavoro potrebbe essere l'esplorazione di configurazioni multimodali alternative, mirate a migliorare sia l'accuratezza che la robustezza della stima.

Abstract

Accurate blood pressure (BP) monitoring is crucial for cardiovascular health monitoring and management. Traditional cuff-based methods, while reliable, are impractical for continuous tracking.

This study explores a non-invasive approach to BP estimation using photoplethysmography (PPG) signals, leveraging classical machine learning (ML), deep learning (DL), and transformer-based models. A comprehensive benchmark analysis is performed across various datasets to assess the efficacy of these models.

The research evaluates feature-based methods, like SVR, Decision Trees and Random Forests, alongside deep architectures, including CNN-based models such as CNN-LSTM, ResNet, UNet, ResUNet with attention, Transformer networks and Foundation Models.

The models are tested on VitalDB dataset, obtaining for SBP and DBP respectively a lowest MAE of 5.33 and 3.42 mmHg in the case of calibration based test sets and 12.61 and 8.04 mmHg in the case of calibration free approach.

Moreover the proposed methods are tested on smaller benchmark datasets, to verify how model performances change when applied to data of different sources. The results indicate that model accuracy depends on experimental conditions, including the measurement setup, the dataset used, and data acquisition methods, therefore specific use cases and data properties must be considered when choosing the BP estimation approach. The results indicate that DL models enhance BP prediction accuracy particularly on large datasets, when compared to traditional feature based ML methods on the same data. Furthermore, traditional ML methods achieve better results for smaller datasets, denoting the difficulties of adequately training DL on datasets of reduced size. Moreover, when incorporating ECG signals together with PPG, the BP prediction errors are notably reduced, highlighting the possibility of increasing accuracy with sensor fusion.

Although the results obtained are promising, a purely algorithmic solution based solely on PPG or its combination with ECG does not appear to provide a sufficient level of reliability for clinical or diagnostic applications. Consequently, a possible future development of this work could be the exploration of alternative multimodal configurations aimed at further improving both the accuracy and robustness of the estimation.

Contents

- List of Figures** **xii**
- List of Tables** **xiii**
- List of Acronyms** **xv**
- 1 Introduction** **1**
- 2 Background** **3**
 - 2.1 Blood pressure monitoring 3
 - 2.1.1 Blood pressure measurement 4
 - 2.2 Physiological Signals 6
 - 2.2.1 ECG Signal 6
 - 2.2.2 PPG signal 7
 - 2.2.3 ABP Signal 9
 - 2.3 Machine Learning and Deep Learning 10
 - 2.3.1 Feature extraction 11
 - 2.3.2 Convolutional Neural Networks (CNNs) 14
 - 2.3.3 Temporal Convolutional Networks (TCNs) 16
 - 2.3.4 Long Short-Term Memory (LSTM) Networks 17
 - 2.3.5 Transformers 18
 - 2.3.6 Foundation Models 21
- 3 Non-Invasive Blood Pressure Estimation Methods** **23**
 - 3.1 Relationship between PPG and Blood Pressure 23
 - 3.2 PPG and ECG Signals for BP Estimation 24
 - 3.2.1 Pulse Arrival Time (PAT) 24
 - 3.2.2 Pulse Transit Time (PTT) 25
 - 3.2.3 Single PPG Signal 26

Contents

3.3	State of the Art of PPG-based BP estimation	26
3.3.1	Features to label estimation	27
3.3.2	Signal to label estimation	28
3.3.3	Signal to signal estimation	28
3.3.4	Standards for wearable cuffless blood pressure measuring devices	29
4	Materials and Methods	31
4.1	Public Physiological Signals Datasets	31
4.1.1	VitalDB Subset of PulseDB	33
4.1.2	Benchmark Datasets	34
4.2	Blood Pressure Estimation Algorithms	35
4.2.1	Feature Extraction	35
4.2.2	Neural Networks with Temporal Signal Input	39
4.3	Training and Testing	46
4.3.1	Data normalisation	46
4.3.2	Model implementation and training	47
4.4	Evaluation Metrics	48
5	Results	51
5.1	Comparison Between Models	51
5.1.1	VitalDB results	52
5.1.2	Benchmark datasets results	60
5.2	Comparison with State-of-the-Art	62
6	Conclusion and Future Work	65
	Bibliography	67

List of Figures

2.1	Diagram of the manual sphygmomanometer use and working principle	5
2.2	(Left) Reflective and (Right) transmissive operating principles of PPG [1]	7
2.3	Working principle of PPG [2]	8
2.4	Example of PPG signal [3]	9
2.5	Example of ABP signal [4]	10
2.6	Pipeline of ML models applied to features	12
2.7	Multi-Layer Perceptron architecture	12
2.8	Architecture of a 1D CNN	14
2.9	LSTM cell diagram	18
2.10	Attention layers [5]	19
2.11	Transformer model architecture [5]	20
3.1	Example of PAT measurement [6]	24
3.2	Example of PTT measurement [7]	25
3.3	PPG features extracted from the pulse waveform [8]	27
4.1	PPG waveform with detected peaks, onsets, and its second derivative showing fiducial points a and b	36
4.2	PPG waveform with the identification of the dicrotic notch	37
4.3	PPG waveform with intervals marked at 25%, 33%, 50%, 66%, and 75% of the maximum amplitude	37
4.4	Hybrid CNN-LSTM network [9]	40
4.5	Residual block of the ResNet NN	41
4.6	Original UNet architecture [10]	42
4.7	Paralleled CNN and Transformer network [11]	43
4.8	ResUNet + Attention network [12]	44
5.1	Prediction results on Vital AAMI Test subset using the Transformer-based model	55
5.2	Results on Vital AAMI Test subset using the ResUNet + Attention with ECG .	56

List of Figures

5.3 Results on Vital CalFree subset with the Transformer model 57

5.4 Results on Vital CalBased subset with the CNN-LSTM model with only the PPG 58

5.5 Results on Vital CalBased subset with the CNN-LSTM model with ECG 59

List of Tables

2.1	Blood pressure level classification	4
3.1	Summary of the SoA	29
3.2	BHS standard for BP measuring devices	30
4.1	Characteristics of public datasets containing PPG and blood pressure signals . .	31
4.2	Characteristics of VitalDB subsets of the PulseDB database	34
4.3	Characteristics of the benchmark datasets	35
4.4	PPG features summary	38
5.1	Model performances on the VitalDB subsets	52
5.2	Number of parameters and memory size for the different DL models	60
5.3	Model performances on the UCI, Sensors, BCG, and PPGBP datasets	61

List of Acronyms

AAMIAssociation for the Advancement of Medical Instrumentation
ABPArterial Blood Pressure
AIArtificial Intelligence
ANNArtificial Neural Networks
APGAcceleration Plethysmogram
BCGBallistocardiography
BHSBritish Hypertension Society
BPBlood Pressure
CNNConvolutional Neural Network
DBDatabase
DBPDiastolic Blood Pressure
DLDeep Learning
DNDicrotic Notch
ECGElectrocardiography
FCFully Connected
FMFoundation Model
GRUGated Recurrent Unit

List of Acronyms

HFHigh Frequency
HRHeart Rate
ICUIntensive Care Unit
LASILarge Artery Stiffness Index
LFLow Frequency
LRLinear Regression
LSTMLong Short-Term Memory
MAEMean Absolute Error
MLMachine Learning
MLPMulti-Layer Perceptron
MSEMean Squared Error
NNNeural Network
PATPulse Arrival Time
PCTNParallel Convolutional Transformer Network
PEPPre-Ejection Period
PPGPhotoplethysmography
PSDPower Spectral Density
PTTPulse Transit Time
ReLURectified Linear Unit
RFRandom Forest
RMSERoot Mean Square Error
RNNRecurrent Neural Network
SBPSystolic Blood Pressure
SoAState of the Art

List of Acronyms

- STDStandard Deviation
- SVMSupport Vector Machines
- SVRSupport Vector Regression
- TCNTemporal Convolutional Network
- VLFVery Low Frequency

Introduction

Blood pressure (BP) is a fundamental physiological parameter essential for assessing cardiovascular health. As a matter of fact, hypertension, or high blood pressure, is a major risk factor for severe medical conditions such as heart disease, stroke, and kidney failure.

While traditional cuff-based methods such as sphygmomanometers offer reliable measurements, their intermittent nature and discomfort limit continuous monitoring, while the intra-arterial catheter method presents risks due to invasive procedures.

These limitations have motivated the exploration of alternative methods for continuous and non-invasive BP estimation, particularly leveraging the application of novel machine learning and deep learning methods on physiological signals and wearable technology.

This thesis investigates a non-invasive approach to estimate BP using photoplethysmography (PPG) signals, evaluating and benchmarking various computational models, including classical machine learning (ML), deep learning (DL), and transformer-based architectures. In particular, feature-based methods such as Support Vector Regression (SVR), Decision Trees, and Random Forests are compared with more sophisticated deep learning models, including CNN-LSTM, ResNet, UNet, ResUNet with attention, Transformer networks and Foundation Models.

The study's main contributions include:

- A comprehensive benchmark analysis on several publicly available datasets, such as VitalDB, UCI, Sensors, BCG, and PPGBP, to evaluate the performance and generalisability of the implemented models.
- Identification of the dependency of model performance on experimental conditions, including dataset characteristics and measurement setup, highlighting the necessity of choosing estimation methods aligned with the specific target data and clinical application context.

1 Introduction

- Demonstration of significant improvements in BP estimation accuracy through calibration-based approaches, with a minimum achieved mean absolute error (MAE) of 5.33 mmHg for systolic (SBP) and 3.42 mmHg for diastolic blood pressure (DBP) using a CNN-LSTM model on the VitalDB dataset.
- Evaluation of calibration-free scenarios, showing that Transformer-based models achieved the lowest errors (12.61 mmHg SBP, 8.04 mmHg DBP) among the tested approaches.

Results demonstrate that the optimal model for BP estimation significantly depends on dataset characteristics, with deep learning models generally excelling on large datasets and traditional machine learning methods showing competitive performance on smaller datasets.

Furthermore, the thesis highlights the importance of calibration and sensor fusion, suggesting that integrating multiple physiological signals and subject-specific calibration substantially improves BP estimation accuracy, providing practical insights for future wearable device implementations.

Although the results obtained are promising, a purely algorithmic solution based solely on PPG or its combination with ECG does not appear to provide a sufficient level of reliability for clinical or diagnostic applications. Consequently, a possible future development of this work could be the exploration of alternative multimodal configurations aimed at further improving both the accuracy and robustness of the estimation.

The remainder of this thesis is structured as follows:

- Chapter 2 provides background information, describing the traditional BP measurement methods, physiological signals such as ECG, PPG and arterial blood pressure (ABP) and the use of artificial intelligence methods for regression tasks.
- Chapter 3 reviews the alternative BP estimation methods that have been developed in the past, with a focus on PPG-based methods, and gives an overview of the current State of the Art in BP estimation from PPG signals.
- Chapter 4 outlines the datasets used in this study, including public physiological signal datasets like MIMIC-III, VitalDB and benchmark datasets. Moreover it describes the implementation of the experimental methods that were used.
- Chapter 5 discusses the obtained results from the different regression methods and compares them to the State of the Art.
- Chapter 6 concludes this work with suggestions for potential areas for future research.

Background

This chapter introduces the blood pressure estimation problem, first by describing the importance of continuous BP monitoring, then by characterising the different traditional methods of BP measurement. Moreover, the most important physiological information on signals that are frequently used in innovative BP monitoring techniques is outlined.

Finally, a brief description of some of the most significant machine learning and deep learning architectures commonly applied to biomedical signals is given.

2.1 Blood pressure monitoring

Blood Pressure (BP) is one of the most relevant vital parameters which are commonly monitored in ambulatory settings, due to its importance in assessing cardiovascular status and the risk of associated diseases. Specifically, blood pressure is defined as the force that blood exerts against the wall of a blood vessel.

Systolic Blood Pressure (SBP) is identified as the pressure value measured at the contraction of the heart, which corresponds to the maximum force exerted by the blood being pumped into the circulatory system, while Diastolic Blood Pressure (DBP) is the pressure at the dilation of the heart.

BP values are measured in millimeters of mercury (mmHg), and based on the results obtained from the measurements in a clinical setting, the patient's pressure level classification is defined as shown in Table 2.1.

BP normally varies throughout the day and according to the activities performed; typically it is lower during sleep, while it tends to be higher in the morning and evening [13].

2 Background

Level	Systolic pressure (mmHg)	Diastolic pressure (mmHg)
Optimal	<120	<80
Normal	120–129	80–84
Normal - high	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

Table 2.1: Blood pressure level classification

The primary reason why BP measurement is important is due to the risks associated with high values, which can lead to a diagnosis of hypertension. According to the World Health Organization (WHO), an estimated 1.28 billion adults aged 30 to 79 suffers from hypertension; among them, it is believed that 46% are unaware of their condition, while only 42% are diagnosed and receive treatment [14].

Hypertension is also commonly referred to as the "silent killer" because it can cause severe complications such as heart disease, kidney failure, or stroke, often without presenting obvious symptoms. Therefore, BP monitoring is essential for diagnosing hypertension and assessing a patient's health status, allowing for timely preventive or therapeutic measures. Regular BP checks help detect any abnormalities early, monitor the effectiveness of medication or lifestyle changes, and reduce the risk of long-term cardiovascular complications.

2.1.1 Blood pressure measurement

Blood pressure measurement techniques can either be invasive or non invasive.

The invasive method is the most accurate and it is considered the gold standard for BP measurement; it involves inserting a cannula directly into an artery, usually the radial or femoral artery. Using a transducer, the mechanical wave of BP is converted into an electrical signal, which can be analysed in real time to monitor BP variations. This technique allows for continuous and precise BP measurement, making it useful in contexts such as Intensive Care Units (ICUs) or during complex surgical procedures, where constant monitoring of the patient's BP is essential. However, since it is an invasive procedure, it carries higher risks, such as infections or vascular damage, and it requires specialised skills and a suitable hospital environment to be performed safely.

For these reasons, various non-invasive BP measurement methods are often used, which can also be applied outside of hospital settings.

2 Background

The most widespread instrument for non-invasive and indirect BP measurement is the sphygmomanometer, which can be either manual or electronic.

The manual sphygmomanometer is a medical device which consists of the inflatable arm cuff, a bulb pump for pumping air into the cuff and a pressure dial which measures the air pressure. As shown in Figure 2.1, the working principle of the sphygmomanometer is based on the temporary interruption of blood flow in the patient's arm using the cuff, which is inflated to a pressure higher than that of the blood in the brachial artery and then gradually deflated.

In the case of a manual sphygmomanometer, blood flow is detected using a stethoscope placed below the cuff. The systolic pressure corresponds to the minimum pressure at which the cuff is inflated, where no sound is heard. As the cuff deflates, rhythmic sounds known as Korotkoff sounds can be heard for intermediate pressure values between systolic and diastolic pressure. As the cuff pressure continues to decrease, the diastolic pressure is identified as the point at which the sounds disappear, indicating that normal blood flow has resumed in the artery.

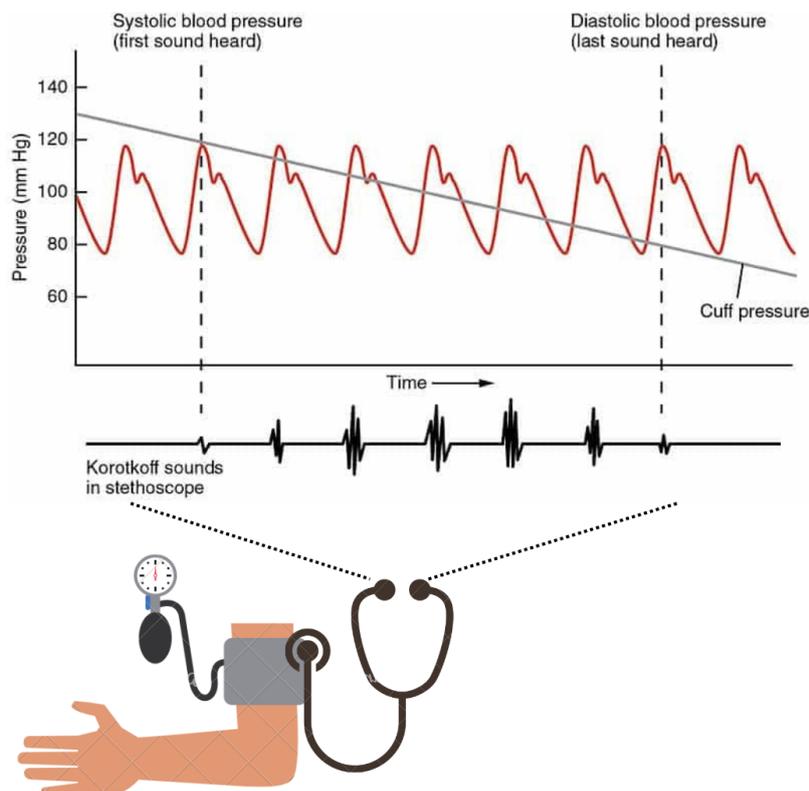


Figure 2.1: Diagram of the manual sphygmomanometer use and working principle

An alternative instrument to the manual sphygmomanometer is the electronic sphygmomanometer, which allows for automatic BP measurements, so it can also be used in non-ambulatory settings by patients themselves, since specialised medical personnel is not needed.

This type of instrument does not use the auscultatory method like the manual sphygmomanometer, but instead relies on oscillometric measurements. The oscillometric method analyses air pressure oscillations in the cuff, which are caused by pressure fluctuations in the brachial artery.

BP values are then determined using proprietary algorithms developed by the medical device manufacturers [15].

The main disadvantage of these devices is the cuff that the patient must wear to enable the measurement. This can be bulky and uncomfortable for prolonged use, since normal blood flow is interrupted. Furthermore, the cuff needs to be inflated and deflated repeatedly to obtain the measurements, making it unsuitable for continuous BP monitoring.

As a result, several researchers have sought alternative methods for non-invasive estimation of BP that do not require the use of a cuff and allow for continuous measurement for uninterrupted monitoring.

Among the various solutions, one method that stands out is the use of physiological signals such as electrocardiogram (ECG) and photoplethysmogram (PPG).

2.2 Physiological Signals

Physiological signals are signals produced by the body during its functioning, by creating changes in electrical, mechanical, chemical activity which can be measured. They are parameters that can give insight into the wellbeing of the patient and allow the medical professionals to diagnose, monitor and treat different diseases and conditions.

The main physiological signals traditionally measured to assess the status of the cardiovascular system are the arterial blood pressure signal (ABP) and the electrocardiogram (ECG).

In addition, the photoplethysmogram (PPG) has gained particular attention in recent years due to its ability to detect blood volume changes within blood vessels, with promising results for non-invasive heart rate and BP estimation.

2.2.1 ECG Signal

The electrocardiogram (ECG) measures the electrical activity of the heart during the cardiac cycle. It is performed by placing electrodes on the patient's body to measure the potential differences generated by the depolarization and repolarization of the heart muscle.

The importance of the ECG signal lies in the fact that by analysing its waveform, medical personnel and new automated technologies can non-invasively detect the onset of heart diseases, without any risks to the patient. ECG measurement in an outpatient setting for diagnostic purposes typically uses multiple electrodes, positioned to obtain 12 leads, which allow a comprehensive analysis of the heart's dipole vector variations. For wearable devices, the measurement is reduced to a single lead to ensure compactness and user comfort.

2 Background

In literature, ECG is often used in conjunction with PPG in studies for BP estimation.

In fact, ECG alone does not provide sufficient information for BP estimation. Evidence suggests that ECG characteristics are indirectly related to BP, as the morphology of the ECG is influenced by heart diseases, which, in turn, are associated with high BP values [16].

2.2.2 PPG signal

Photoplethysmography (PPG) is an optical technique that allows the measurement of volumetric variations of blood circulating in blood vessels.

The PPG signal is gaining increasing relevance in medical research due to its simplicity of non-invasive measurement and low implementation costs. Unlike other plethysmography techniques, which are based on capacitive, inductive, and piezoelectric properties, photoplethysmography relies on optical properties.

PPG measurement is performed using a light source, typically a Light Emitting Diode (LED), which illuminates the skin tissue, and a photodetector, such as a photodiode, which detects variations in light absorption or reflection caused by changes in blood volume within the tissues. PPG can be detected either in transmission or reflection mode, as shown in Figure 2.2.

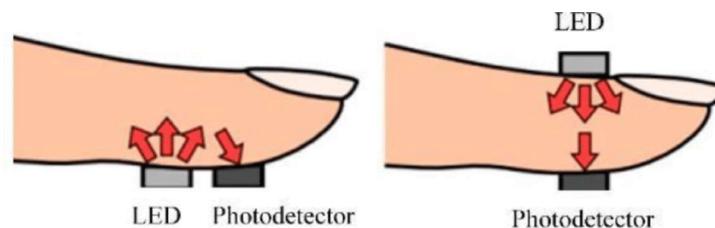


Figure 2.2: (Left) Reflective and (Right) transmissive operating principles of PPG [1]

In the transmission mode, the light source and the photodiode are placed on opposite sides of the tissue, for example, on a finger or an earlobe, so that the light emitted by the LED passes through the tissue and is detected by the photodetector. This mode is particularly effective for thin, well-vascularised tissues.

In the reflection mode, however, the light source and the photodetector are positioned on the same side of the skin surface. The light emitted by the LED penetrates the tissue and is partially reflected by blood vessels before being captured by the photodetector. This configuration is more suitable for body areas where light transmission is impractical, such as the wrist or forehead.

As shown in Figure 2.3, the PPG signal varies during the cardiac cycle due to changes in light absorption by tissues, which are determined by variations in blood volume within the vessels.

2 Background

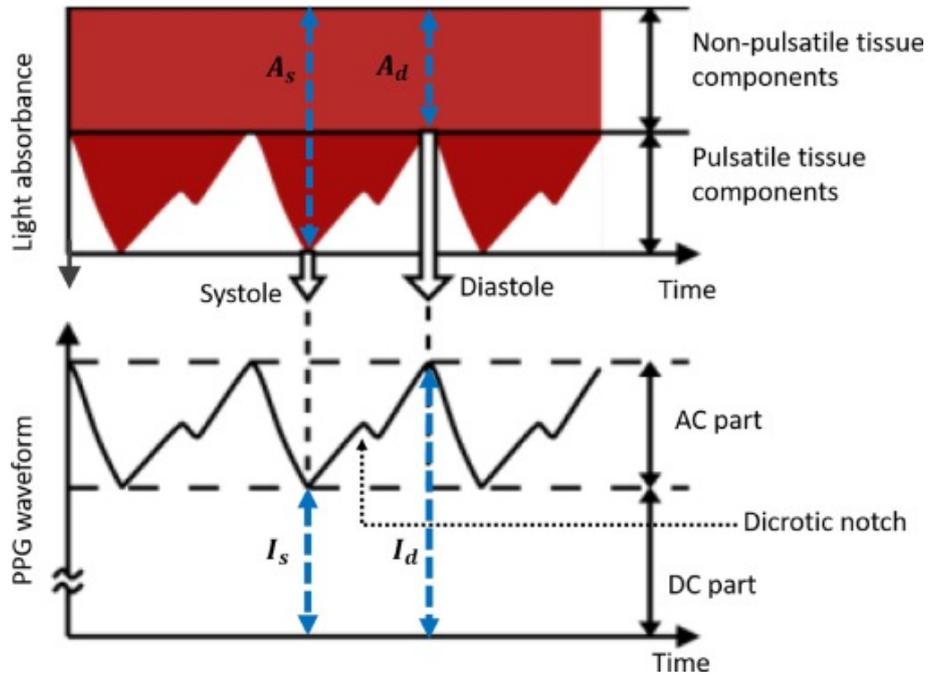


Figure 2.3: Working principle of PPG [2]

During the systolic phase of the cardiac cycle, the ejection of blood from the heart leads to an increase in blood volume within the arterial vessels, resulting in greater light absorption and a decrease in the intensity of the signal measured by the photodiode. Subsequently, during diastole, there is less light absorption and a higher detected signal intensity due to venous return and the reduction of arterial volume.

The PPG signal is continuous and quasi-periodic, as its waveform repeats with each cardiac cycle, though it may undergo variations due to physiological or environmental factors.

As highlighted in Figure 2.3, the signal is characterised by both a direct current (DC) component and an alternating current (AC) component. The DC component is due to the constant absorption of light by denser and less vascularised tissues, such as adipose tissue, connective tissue and bones. It represents the baseline component of the signal and may vary slowly over time due to factors such as respiration, the sympathetic nervous system and skin temperature. The pulsatile component of the signal is associated with BP variations caused by changes in blood volume within the vessels with each heartbeat.

An example of a PPG signal is shown in Figure 2.4. As commonly reported in the literature, the PPG signal typically exhibits a peak corresponding to the systolic phase and a trough during the diastolic phase, in contrast to what is shown in Figure 2.3. In fact, the signal is usually inverted using inverting amplifiers to obtain a waveform similar to the ABP signal during the same cardiac cycle.

2 Background

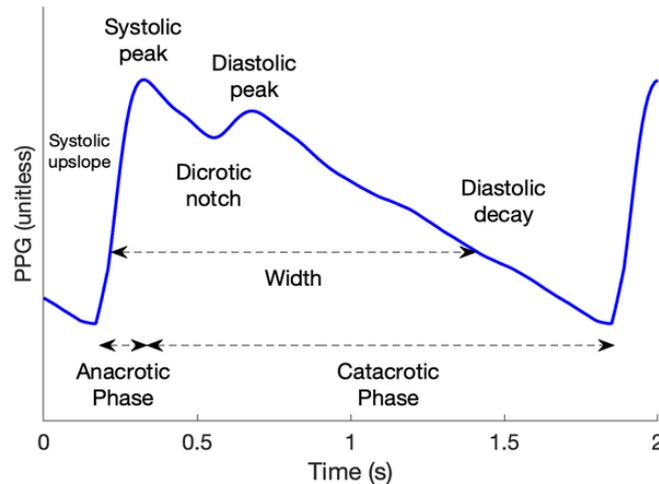


Figure 2.4: Example of PPG signal [3]

Different characteristic points and segments of the PPG signal can be identified:

- Systolic upslope, corresponding to the rapid rise of the PPG signal during the systole, caused by the increased blood volume in the blood vessels as the heart contracts;
- Systolic peak, given by the highest point of the signal due to the maximum blood volume in the arteries;
- Dicrotic notch, namely the signal deflection which can occur following the systolic peak, associated with the closure of the aortic valve;
- Diastolic peak, i.e., the secondary peak which may appear in some cases, signaling the start of the diastole;
- Diastolic decay, the decline in signal intensity caused by the reduction of blood in the vessels as the heart relaxes.

Beyond its recent applications in BP estimation, the PPG signal is most frequently used in medical devices designed for measuring blood oxygen saturation and heart rate.

2.2.3 ABP Signal

The arterial blood pressure (ABP) wave is generated by the displacement of blood following the contraction of the left ventricle, which propagates through the cardiovascular system as a pulsatile wave. This propagation depends on the elastic properties of the vascular walls and the patient's hemodynamic state.

2 Background

As previously described, BP can be measured either invasively or non-invasively. In the first case, the result is a continuous signal, as shown in Figure 2.5, while in the second case, discrete BP values are obtained, corresponding to the time intervals during which the measurements were taken.

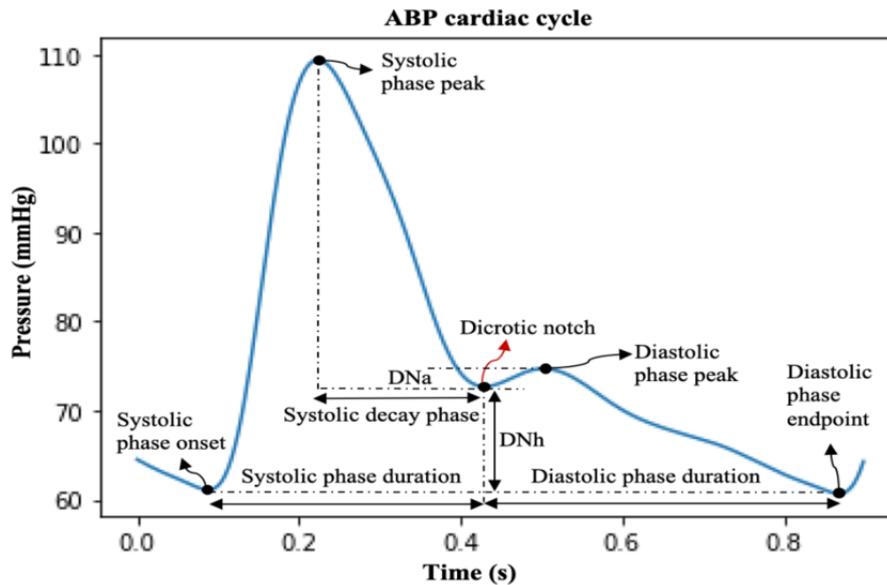


Figure 2.5: Example of ABP signal [4]

Analysing the waveform of the continuous ABP signal allows for the assessment of a patient's haemodynamic properties in intensive care settings, where continuous monitoring is necessary due to the risk of hypotensive states, hypertensive crises or potential patient instability, as well as to evaluate the effects of medications, guide diagnosis and inform therapeutic decisions.

2.3 Machine Learning and Deep Learning

In recent years, Artificial Intelligence (AI) has significantly revolutionised the healthcare field, providing support to medical professionals in diagnostics, personalised treatments, imaging and signal analysis [17]. AI, defined as the ability of a computer to mimic human behaviour in performing tasks, can be divided in the fields of Machine Learning (ML) and Deep Learning (DL).

ML consists of many algorithms that learn patterns from data in order to carry out a task, such as classification or regression. The ML approaches can be supervised or unsupervised [18].

In supervised ML, the input data is labelled with the corresponding output class or value, so that the ML model can measure its accuracy and improve its performance, while in unsupervised learning, the model learns patterns from unlabelled data, without human intervention.

2 Background

Specifically, ML allows models to improve their performance over time as they are exposed to more data. Some examples of such algorithms are Decision Trees, Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbors.

Artificial Neural Networks (ANN) are a subset of ML, inspired by biological neural networks and the way that information propagates between the neurons. An ANN consists of many nodes, or neurons, which receive input data, process it and pass it to the next neuron in the layer toward the output. An ANN typically consists of an input layer, one or more hidden layers and an output layer. If the number of hidden layers is greater than one, then the network can be defined as a deep neural network.

DL is an advanced branch of ML, which uses deep ANN in order to extract information directly from raw data in a data-driven approach [19]. In fact, one of the limitations of traditional ML is the need for preprocessed input data, for example by manual feature extraction. However, with the advent of DL, raw data can be fed directly into the network, as the model itself is capable of automatically learning and extracting the most relevant features from the data.

Some of the most frequently used deep neural networks include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Transformer models.

2.3.1 Feature extraction

Feature extraction consists in the preprocessing of raw data to extract meaningful information to reduce the computational load for a ML algorithm [20]. This procedure is useful for some ML applications, where the direct use of raw data is unfeasible.

Feature extraction is often applied to biomedical signal processing, since important characteristics of such signals can provide valuable information on an individual's health status [21].

In general, features extracted from a signal can be:

- Time domain features, which are computed from the signal time series. They can be general statistical descriptors of a signal such as mean, variance, standard deviation, skewness, kurtosis, amplitude and signal energy, or ad hoc features for a specific signal, computed by waveform analysis.
- Frequency domain features, that represent signal characteristics with respect to frequency rather than time, like band energy ratio, spectral entropy, power spectral density and dominant frequency components.
- Time-frequency domain features, that study a signal in both time and frequency domain at the same time, leveraging wavelet or spectrogram based time-frequency representations.

2 Background

After the feature extraction process, the obtained features are normalised and given as inputs to the classifier or regression ML algorithm. The approach for the use of extracted features and ML models is summarised in Figure 2.6.

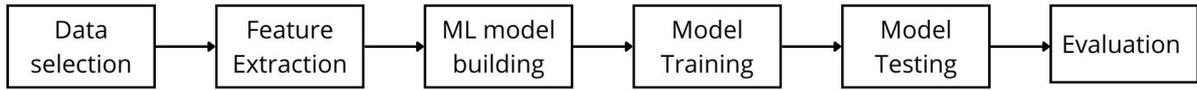


Figure 2.6: Pipeline of ML models applied to features

Regression is a type of supervised learning that has the goal to predict a continuous output variable based on input features. Various ML models can be used for regression, some are presented in the following paragraphs.

Multi-Layer Perceptron (MLP)

A Multi-Layer Perceptron (MLP) is one of the most known and used ANN. It can model non-linear relationships between features and the target variable, so it can be used for regression problems [22]. The basic architecture of MLP is depicted in Figure 2.7.

The extracted features are fed into the input layer of the MLP, which processes them in its hidden layers and generates the predicted output; it is a feed-forward NN, which means that information flows from the inputs to the outputs without loops and the output of each neuron does not affect the neuron itself.

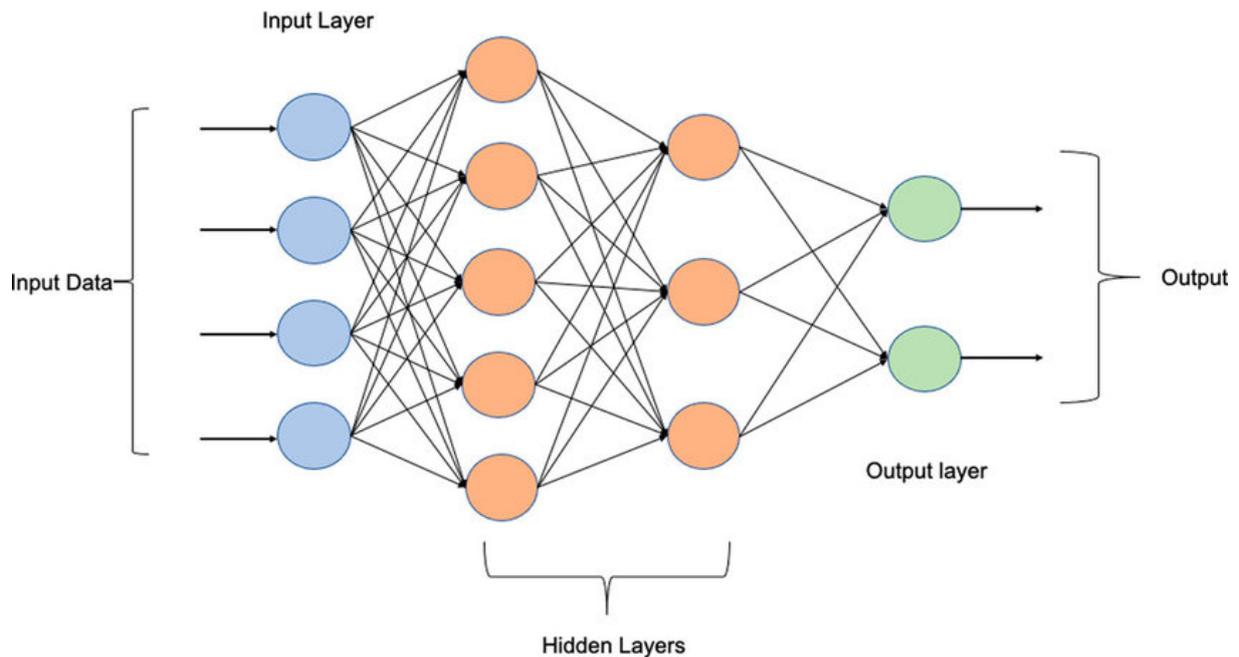


Figure 2.7: Multi-Layer Perceptron architecture

2 Background

Each hidden layer utilises a non-linear activation function that models the action potentials of biological neurons. This is important because it introduces non-linearity in the model, in order to obtain the desired output. MLP models are trained in a supervised manner with the backpropagation algorithm, which means that MLP layers' weights are iteratively updated by minimising the error between MLP outputs and the desired outputs indicated by the labeled data.

Support Vector Regression (SVR)

Support Vector Regression (SVR) is a version of Support Vector Machines (SVM) for regression that can give continuous output values instead of a discrete class. SVM are supervised classification algorithms that work by identifying an hyperplane that separates the classes of data with a maximum margin. SVR works similarly, but instead of finding a hyperplane that separates classes, it attempts to find a function that approximates the relationship between the input features and the target variable while maintaining an acceptable error margin [23].

Decision Trees

Decision Trees are a type of predictive models which recursively divide the dataset based on feature values [24]. The features extracted from the signals are used to make branching decisions in the tree structure. Each internal node of the tree represents a decision based on a specific feature, while the leaf nodes correspond to the predicted output values. The tree splits the data at each node by selecting the feature and threshold that result in the most significant reduction in variance: the goal is to find a feature and a threshold that best separates the data into more homogeneous subsets. In classification, this means groups of data points that belong to the same class, and in regression, it means groups of data points with similar numeric values.

Random Forest

Random forests or random decision forests is an ensemble learning method that works by creating multiple decision trees during training based on the input data and obtains the final prediction by averaging the outputs of all the trees. This type of approach reduces the decision trees' risk of overfitting to the data [25].

2.3.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a type of feed-forward ANN which represent one of the most significant networks in the DL field [26].

CNN's architecture is based on visual perception and it consists of layers that perform convolution operations, followed by activation functions, pooling layers, and fully connected layers. An example of CNN architecture is shown in Figure 2.8.

Key Components of CNNs

The basic elements of the CNN architecture are the following:

- Convolutional layers, the core building block of CNNs, which convolves the input tensor and gives feature maps as output. It is characterised by filter size, stride, which determines how much the filter moves at each step, and padding, that controls how the input is extended at the borders.
- Activation Functions, typically applied after a convolutional or fully connected layer, allowing non-linearity in the model.
- Pooling Layers, which reduce the dimensionality of the data in the network.
- Fully Connected Layers, also known as Dense layers, which are layers where all the neurons are connected to every neuron in the previous layer and give the final prediction based on the features extracted in the previous layers.

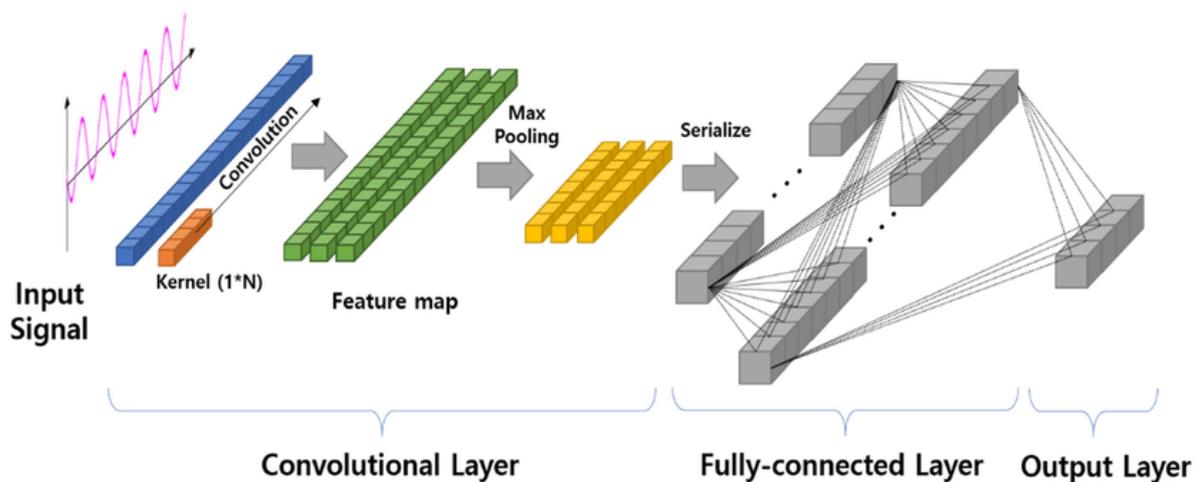


Figure 2.8: Architecture of a 1D CNN

Convolution Operation

The convolution operation is the core of a CNN. It involves applying a small filter, or kernel, over an input matrix to detect spatial or temporal patterns. This operation is used in the CNNs in order to autonomously extract features from the input signal or image, depending on the application domain. The discrete convolution operation is shown in Equation 2.1:

$$s(t) = (x * w)(t) = \sum_{i=0}^{k-1} x(t+i) \cdot w(i) \quad (2.1)$$

where:

- x is the 1D input sequence;
- w is the filter (or kernel) of length k ;
- t is the output sequence index;
- $s(t)$ is the convolution output sequence;
- $*$ is the convolution operation symbol.

Activation Functions

Activation functions are non-linear functions that are applied at the output of NN layers to add non-linearity and complexity to the computed output [27].

Some of the most used activation functions consist in:

- ReLU (Rectified Linear Unit), a fixed-shape activation function that, given an input x , computes its output as shown in Equation 2.2;

$$ReLU(x) = \max(0, x) \quad (2.2)$$

- Leaky ReLU, an extension of ReLU defined in Equation 2.3, where α is a small constant;

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases} \quad (2.3)$$

- Sigmoid, a smooth S-shaped function that maps inputs to a range between 0 and 1, as defined in Equation 2.4;

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

- Tanh (Hyperbolic Tangent), which maps inputs to a range between -1 and 1, defined in Equation 2.5.

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

Pooling Layers

Pooling layers are used to reduce the dimensionality of the data in the network, which helps in reducing the number of parameters and computation and to avoid overfitting [28].

The two most common types of pooling layers are:

- Max pooling, where the maximum value is selected among the inputs;
- Average pooling, which computes the mean of the input values.

Fully Connected Layers

Fully Connected (FC) layers are introduced at the end of the NN, to combine all the features extracted in the convolutional layers and produce the final output [29]. The output of a FC layer is given by Equation 2.6:

$$y = f(Wx + b) \tag{2.6}$$

where:

- W is the weight matrix of the layer;
- x is the input vector;
- b is the bias vector;
- f is the activation function applied to the output of the linear transformation.

2.3.3 Temporal Convolutional Networks (TCNs)

Temporal Convolutional Networks (TCNs) are built on standard Convolutional Neural Network (CNN) principles but are specifically designed for sequential data [30].

The key characteristics that distinguish TCNs from regular CNNs are:

- Causal Convolutions, namely convolutions where the output at a certain time step does not depend on future time steps. Given an input sequence $x = [x_1, x_2, \dots, x_T]$, a causal convolution ensures that at time t , the output y_t is only influenced by x_t and previous inputs, preserving the temporal order of the input data.
- Dilated Convolutions, a type of convolution where a dilation factor d is introduced to expand the receptive field without increasing the number of layers, so the kernel has gaps and skips some signal samples for one-dimensional data or image pixels for two-dimensional data. In a dilated convolution, the output at time step t is computed as:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-d \cdot i} \tag{2.7}$$

2 Background

where w_i are the convolution filter weights, k is the filter size, and d determines the spacing between sampled input values.

- Residual Connections, which consist in shortcut paths that allow data to bypass multiple layers, in order to mitigate the vanishing gradient problem and facilitate training, as described in Equation 2.8:

$$y = F(x) + x \quad (2.8)$$

where $F(x)$ represents the output of the convolutional layers and x is the input to the residual block.

2.3.4 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) Networks are a type of Recurrent Neural Networks (RNNs), specifically designed to address the limitations of traditional RNNs when dealing with long-term dependencies in sequential data [31].

RNNs are a class of neural networks designed for sequential data. Unlike feedforward neural networks, RNNs have a hidden state that allows them to maintain memory of previous inputs. While RNNs work well for short sequences, they struggle with long-range dependencies due to the vanishing and exploding gradient problem during backpropagation [32].

LSTM networks are a special type of RNN that solve the vanishing gradient problem by introducing a memory cell and a set of gates that control information flow and use sigmoid and tanh activation functions.

The gates that are used in an LSTM layer are the forget layer, which selects if the content stored in the cell is to be discarded or not, the input gate, which decides the information to be stored in the cell, and the output gate, that selects the information to give as output.

Figure 2.9 depicts the diagram of a LSTM cell.

The operations inside an LSTM cell can be described using the following equations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (\text{Forget Gate}) \quad (2.9)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (\text{Input Gate}) \quad (2.10)$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (\text{Cell Candidate}) \quad (2.11)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (\text{Cell State Update}) \quad (2.12)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (\text{Output Gate}) \quad (2.13)$$

$$h_t = o_t \odot \tanh(C_t) \quad (\text{Hidden State Update}) \quad (2.14)$$

Where:

- x_t is the input at time step t ,
- h_{t-1} is the hidden state from the previous time step,
- C_{t-1} is the previous cell state,
- W_* , U_* , and b_* are the weight matrices and biases for each gate,
- σ represents the sigmoid activation function,
- \tanh represents the hyperbolic tangent activation function,
- \odot denotes element-wise multiplication.

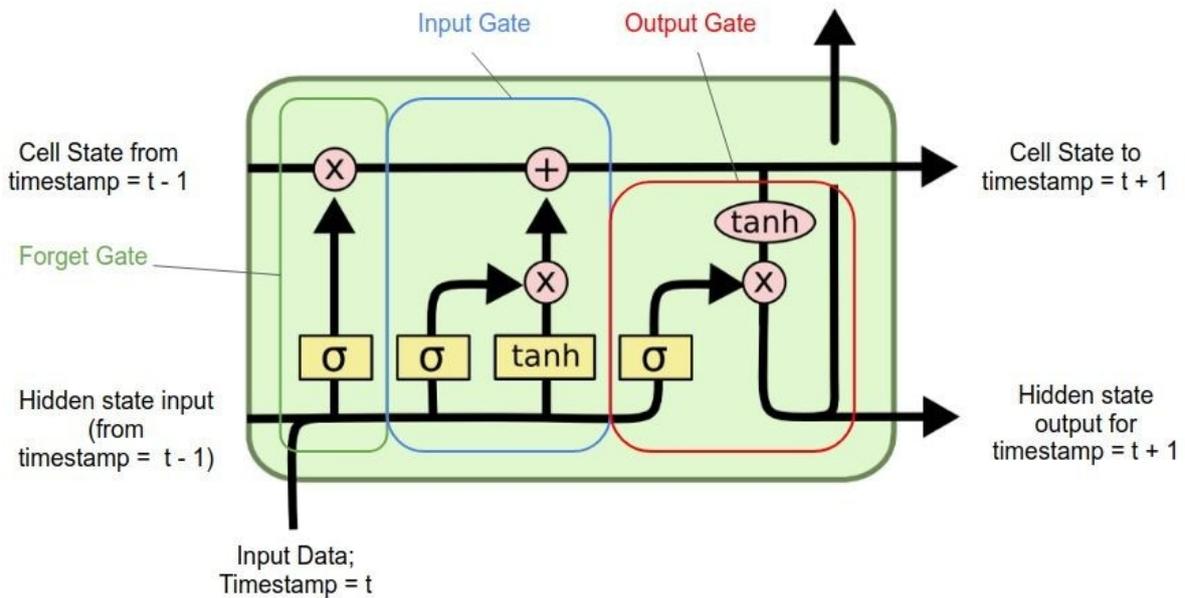


Figure 2.9: LSTM cell diagram

2.3.5 Transformers

Transformers are a type of DL model designed for processing sequential data, particularly excelling in natural language processing tasks, which were first presented in the research paper by Ashish Vaswani *et al.* called “Attention Is All You Need” [5].

Contrary to recurrent architectures such as LSTMs, transformers leverage self-attention mechanisms to process input sequences in parallel, which means that they assign different importance to various parts of the input sequence, allowing them to focus on the most significant features dynamically.

Self-Attention Mechanism

The self-attention mechanism enables the model to weigh the importance of different elements in the input sequence when making predictions.

Figure 2.10 showcases the Attention and Multi-Head Attention layers diagrams presented in [5], whose output is computed as described by the Equations 2.15, 2.16 and 2.17.

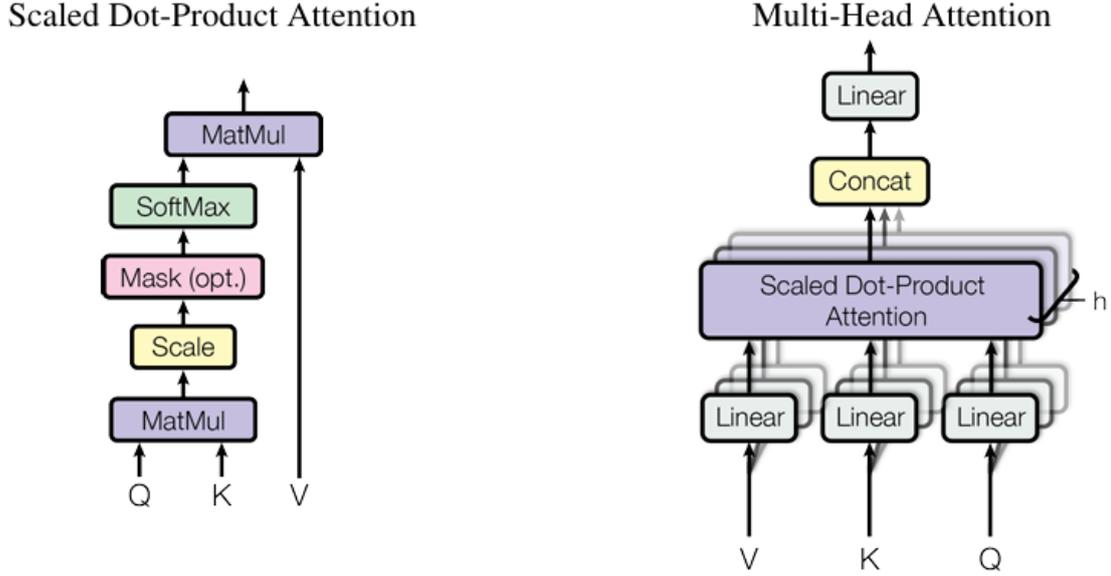


Figure 2.10: Attention layers [5]

The attention mechanism computes the output as:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.15)$$

where Q represents the queries matrix, K the keys matrix, V the values matrix and d_k is the dimension of the keys.

Multi-head attention applies multiple attention heads in parallel:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (2.16)$$

where each attention head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (2.17)$$

Key Components of Transformers

The Transformer model, as shown in Figure 2.11, consists of an Encoder-Decoder structure. Both encoder and decoder blocks are composed by the repetition of N blocks made of multiple

2 Background

layers, each incorporating the following key components:

- Multi-Head Self-Attention, the layer that extract features from the data, capturing short and long-term dependencies;
- Positional Encoding, necessary to add information about the relative position of tokens in the sequence, since the model does not contain recurrent or convolutional layers;
- Layer Normalisation and Residual Connections. Residual connections are useful to mitigate the vanishing gradient problem, while layer normalisation helps stabilise and accelerate training.

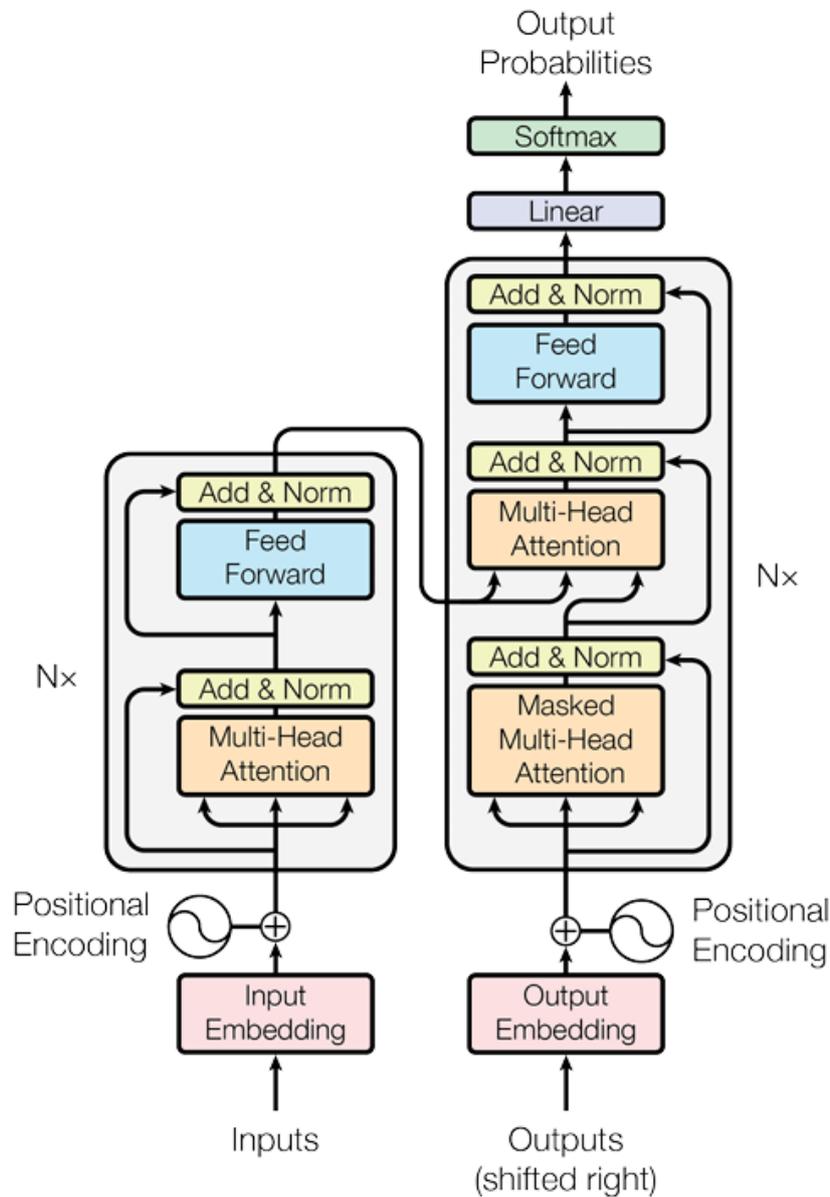


Figure 2.11: Transformer model architecture [5]

2.3.6 Foundation Models

Foundation Models (FMs) represent a recent advancement in AI, where a single large-scale model is pre-trained on massive datasets and fine-tuned for specific tasks [33].

These models, typically based on deep neural networks, have revolutionised AI by enabling transfer learning across various domains [34].

Foundation models possess the following characteristics:

- FMs are pretrained on large and diverse datasets, in order to allow the model to develop an understanding of patterns in data across different domains;
- Scalability, since FMs can process and understand large volumes of data more efficiently than traditional models;
- Generalization across tasks, given by the fact that FMs are trained on a vast quantity of data, so they can be adapted to different tasks.

Non-Invasive Blood Pressure Estimation Methods

This chapter outlines the alternative cuff-less BP measurement methods that have been designed in the last decades, with a focus on PPG and ECG based methods. Then, a brief overview of state-of-the-art methods of BP estimation from the PPG signal is given.

As an alternative to the classical BP measurement methods presented in Chapter 2, several techniques have been developed over the years for non-invasive BP estimation, some leading to the creation of commercially available devices. Some of these methodologies include:

- Peñáz's volume-clamp method [35];
- Ultrasound [36];
- Tonometry [37];
- Bioimpedance [38];
- PPG and ECG signals, which are the focus of this thesis [39, 40].

3.1 Relationship between PPG and Blood Pressure

Although, as presented in Section 2.2, the PPG and ABP signals share similarities in their waveform, such as the presence of a systolic peak, a dicrotic notch, and a diastolic peak, the relationship between the PPG signal and BP values is not straightforward.

This difficulty is due to the factors that can influence the PPG waveform, such as patients' arterial compliance and stiffness, peripheral resistance, and individual differences in arterial

properties, as well as external factors such as the PPG sensor placement, motion artifacts and temperature, which add additional variability to the signal [41].

Nevertheless, different methods were implemented for the BP estimation from PPG. This is due to the fact that researchers have identified patterns and features within the PPG waveform that correlate with blood pressure changes. Many signal processing techniques and machine learning models have been explored to extract meaningful information from the PPG signal and improve BP estimation accuracy.

3.2 PPG and ECG Signals for BP Estimation

Numerous studies have been conducted on the use of physiological signals like PPG and ECG for continuous BP estimation. These studies can be categorised into three main approaches:

- Using both types of signals to estimate BP from the Pulse Arrival Time (PAT) [6];
- Measuring PPG signals at different body locations to compute the Pulse Transit Time (PTT) for BP estimation [7];
- Estimating BP from a single PPG signal by extracting features and applying advanced machine learning and deep learning techniques [42].

3.2.1 Pulse Arrival Time (PAT)

The *Pulse Arrival Time* (PAT) is defined as the time interval between the R-wave of the ECG and a fiducial point of the PPG signal, typically the systolic peak, both associated with the same cardiac cycle, as illustrated in Figure 3.1.

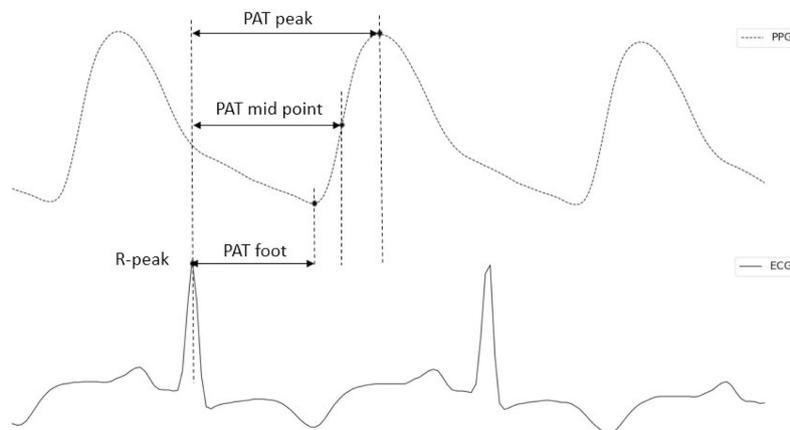


Figure 3.1: Example of PAT measurement [6]

PAT is the sum of the PTT and the Pre-Ejection Period (PEP), which is defined as the time interval between the depolarization of the left ventricle and the beginning of ventricular ejection of blood through the aorta.

PAT has been studied as a surrogate of blood pressure and many studies have demonstrated that a correlation exists between PAT and BP [43]. PAT consists of both PTT and PEP, meaning it is influenced not only by arterial stiffness and vascular properties but also by cardiac contractility and pre-ejection dynamics. As a result, its correlation with blood pressure (BP) is lower than that of PTT alone. Additionally, PEP is affected by factors such as stress, age, emotions, and physical exertion, introducing further variability to PAT. Another limitation of PAT-based BP estimation is the requirement to measure two distinct signals—ECG and PPG—which can be challenging to implement in wearable devices due to motion artifacts and the necessity for synchronized signal acquisition. Moreover, PAT can vary significantly between individuals, necessitating a calibration step in some studies [44].

3.2.2 Pulse Transit Time (PTT)

The *Pulse Transit Time* (PTT) is measured using two PPG sensors placed at a certain distance from each other: the sensor closer to the heart is positioned proximally and is also called the leading sensor, while the one further away is placed distally and is also termed lagging sensor. The PTT is defined as the time interval between the arrival of the pulsation wave at the proximal sensor and its arrival at the distal sensor [45].

PTT can be measured from different fiducial points of the two PPG signals measured at the proximal and distal location, as shown in Figure 3.2.

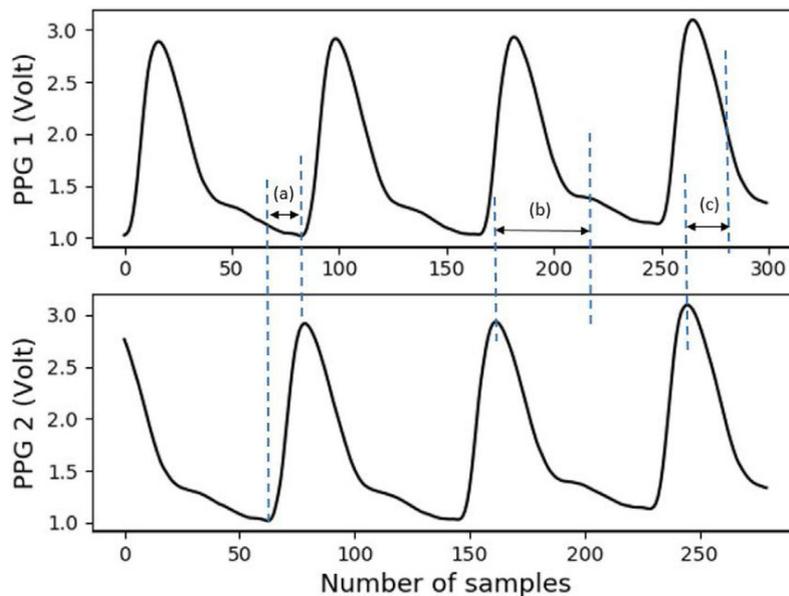


Figure 3.2: Example of PTT measurement [7]

PTT, like PAT, is often inversely related to BP and it depends on individual physiological factors. Since PTT does not depend on PEP, it is often considered a more direct approach to BP estimation than the use of PAT. The drawbacks of using PTT are the need to measure the PPG signal from two different body locations, so there is the need for two synchronised sensors, additionally this introduces problems due to motion artifacts, making the detection of PPG fiducial points more difficult.

3.2.3 Single PPG Signal

In recent years, the use of a single PPG signal for BP estimation has gained increasing attention, since it bypasses the requirement for ECG electrodes or multiple PPG sensors.

The main distinction of the methods that only use the PPG signal is between methods based on features extracted from the PPG signal or methods which directly take the PPG signal as input. The first approach is the classical one, which leverages traditional ML algorithms to model a relationship between the target output and the input features [46]. The second approach is the more modern one, which is possible thanks to the faster computing of modern hardware and advancements in DL techniques; this allows the DL model itself to learn the most relevant features from the raw input signal, eliminating the need for manual feature extraction [47].

3.3 State of the Art of PPG-based BP estimation

The estimation of blood pressure from PPG signals has been an active area of research for the last three decades. As explained in the previous section, there are many techniques of PPG-based BP estimation that have been presented in the last years [48].

The earliest approaches have been centered on PAT and PTT-based estimation, leveraging the inversely proportional relationship that can be found between those time intervals and the BP values. Several studies have explored this methodology, including [49, 50, 51].

With the rise of ML techniques, researchers have increasingly applied ML methods to the BP estimation problem. PPG-based BP estimation methods can be classified in three ways [52]:

- Features to label, where PPG is preprocessed and features are extracted. The prediction method takes the features as input and gives the discrete SBP and DBP as outputs.
- Signal to label, which means that the algorithm takes the temporal signal as input, since it extracts its features autonomously, and estimates the discrete SBP and DBP values.
- Signal to signal, where the input is given by the PPG signal samples and the label is the continuous ABP signal; the output is the reconstructed ABP waveform.

3.3.1 Features to label estimation

Kurylyak *et al.* computed several parameters from the PPG pulse waveform, namely the time periods and intervals in which the pulse reaches certain amplitudes with respect to its peak. The features used in this study are depicted in Figure 3.3 and were used in a multilayer feed-forward ANN for the BP regression task [8].

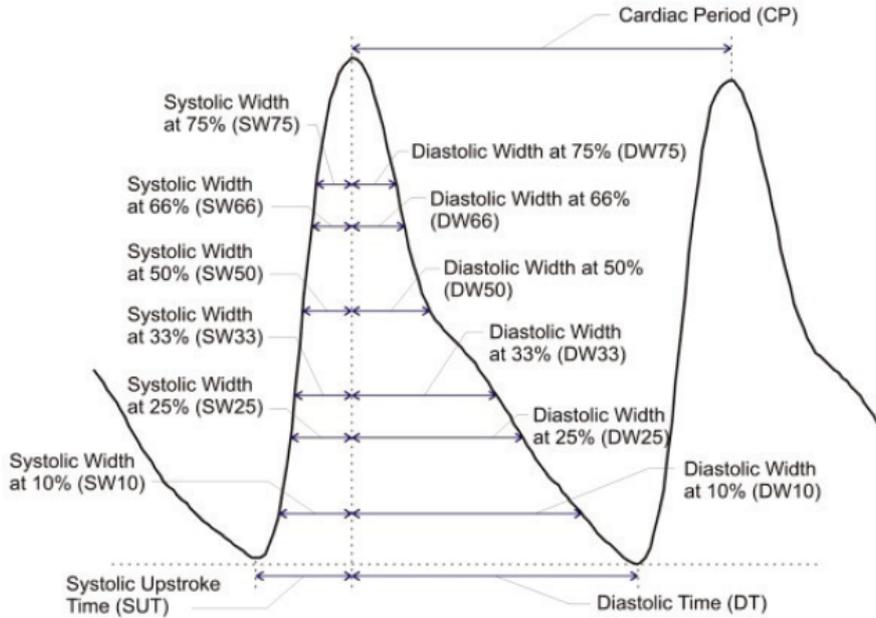


Figure 3.3: PPG features extracted from the pulse waveform [8]

Kachuee *et al.* computed features derived from the ratio of the peak and dicrotic notch amplitudes, the area under the PPG pulse, PAT and heart rate features and make use of different ML regression algorithms such as Linear Regression (LR), Decision Trees, SVR, AdaBoost and Random Forest [53].

Yang *et al.* used features computed from PAT, HR and PPG morphology features such as the ones shown in Figure 3.3 and features obtained from the second derivative of PPG, called Acceleration Plethysmogram (APG). The models used were LR, Random Forest, ANNs, and RNNs [54].

Maqsood *et al.* extract PPG time-domain features, such as those used by Kurylyak *et al.*, statistical features and frequency domain-based features. The extracted information is given as input to traditional ML models, such as LR, Random Forest, AdaBoost, and SVR, and neural models, consisting in LSTM, Bi-LSTM and Gated Recurrent Unit (GRU) [55].

3.3.2 Signal to label estimation

More recent BP estimation methods based on PPG consist in the use of DL models, capable of extracting the PPG features without the need of manual feature engineering. The signal to label approach consists in designing an algorithm which takes the temporal PPG signal as input, extracts its important features autonomously and uses that information to predict the SBP and DBP values for the corresponding time interval in which the PPG was measured.

Many studies make use of hybrid CNN architectures, for example Shaikh *et al.* use a CNN-LSTM architecture where there are two CNN branches for PPG and ECG signal respectively, which extract features that are concatenated and fed to a Bi-LSTM network followed by fully connected layers [56]. In the same study, Shaikh *et al.* perform an ablation study to verify how each module of the network contributes to the prediction accuracy. They observe that all the models perform better if also ECG data is given as input, moreover the CNN-LSTM with Bi-LSTM layers performs better than the simple CNN or LSTM networks.

A similar approach of employing CNN-LSTM networks has been followed by other researchers, like in [57, 58]. Another example of hybrid CNN model is described in [11], where signal features are extracted with a CNN and a Transformer branch.

3.3.3 Signal to signal estimation

Furthermore, the signal to signal approach aims to reconstruct the continuous ABP waveform directly from the PPG signal, providing a more comprehensive representation of BP variations. Paviglianiti *et al.* employ the signal to signal approach of ABP reconstruction, using WaveNet, ResNet and LSTM [59]. A similar approach was followed by Cheng *et al.* [60].

Other signal to signal models include UNet, which is fully convolutional and has an encoder-decoder architecture [52].

Table 3.1 summarises the results of various studies, including the type of model used, the dataset on which the model was trained and evaluated, and the best results obtained. The main difficulty in comparing different BP estimation methods in the literature lies in the fact that different authors have used different data, both in terms of the dataset source and number of subjects and the way the data was split for training and testing the model.

As observed in various studies, like [61], in many cases the researchers have not divided the training and testing set to guarantee that there was no data leakage, which means that the model performs better on the testing set due to the presence of similar or overlapping subject data used during training. This lack of proper data splitting can lead to overestimated performance metrics, making it challenging to assess the true effectiveness of the proposed models.

3 Non-Invasive Blood Pressure Estimation Methods

Approach	Signals	Dataset	Calibration Validation	Dataset size	SBP	DBP
Neural Network[8]	PPG	MIMIC	N/A	Subjects: N/A Segments: 15000 separate PPG heartbeats	MAE: 3.80 STD: 3.46	MAE: 2.21 STD: 2.09
Adaboost [53]	PPG, ECG	MIMIC-II	No 10-Fold	Subjects: 942 Segments: 3663	MAE: 11.17 STD: 10.09	MAE: 5.35 STD: 6.14
SVR [52]	PPG	Sensors	No 5-Fold	Subjects: 1195 Segments: 11102	MAE: 15.60 STD: 19.68	MAE: 7.50 STD: 9.81
AdaBoost [52]	PPG	BCG	No 5-Fold	Subjects: 40 Segments: 3063	MAE: 11.42 STD: 16.44	MAE: 8.06 STD: 10.73
LightGBM [52]	PPG	PPGBP	No 5-Fold	Subjects: 218 Segments: 619	MAE: 13.06 STD: 16.65	MAE: 8.16 STD: 10.30
ResNet [52]	PPG	UCI	No Hold-One-Out	Subjects: N/A Segments: 410596	MAE: 16.59 STD: 20.65	MAE: 8.30 STD: 10.84
CNN-LSTM [56]	PPG	PulseDB	Yes Train/Test split	Train Subjects: 3027 Segments: 1024321 Test Subjects: 2506 Segments: 100240	MAE: 10.74 STD: 14.45	MAE: 6.87 STD: 9.27
CNN-LSTM [56]	PPG, ECG	PulseDB	Yes Train/Test split	Train Subjects: 3027 Segments: 1024321 Test Subjects: 2506 Segments: 100240	MAE: 5.16 STD: 7.73	MAE: 3.24 STD: 5.14
PCTN [11]	PPG	MIMIC-III	No Train/Test split with subject holdout	Subjects: 808 Segments: N/A	MAE: 4.44 STD: 5.98	MAE: 2.36 STD: 3.22
XGBoost [61]	PPG	MIMIC-III	Yes Train/Test split	Subjects: 633 Segments: N/A	MAE: 2.85 STD: 4.31	MAE: 1.32 STD: 2.28
XGBoost [61]	PPG	MIMIC-III	No Train/Test split	Subjects: 633 Segments: N/A	MAE: 16.18 STD: 20.77	MAE: 8.45 STD: 11.60
ResNet+LSTM [59]	PPG	MIMIC	Yes Train/Test split	Subjects: 121 Segments: N/A	MAE: 7.12 RMSE: 11.21	MAE: 3.53 RMSE: 5.03
ResNet+LSTM [59]	PPG, ECG	MIMIC	Yes Train/Test split	Subjects: 121 Segments: N/A	MAE: 4.12 RMSE: 5.68	MAE: 2.23 RMSE: 2.99
ResNet+LSTM [59]	PPG	MIMIC	No Leave-One-Out	Subjects: 121 Segments: N/A	MAE: 23.59 RMSE: 27.64	MAE: 10.75 RMSE: 12.34
ResNet+LSTM [59]	PPG, ECG	MIMIC	No Leave-One-Out	Subjects: 121 Segments: N/A	MAE: 20.37 RMSE: 23.07	MAE: 9.55 RMSE: 10.85

Table 3.1: Summary of the SoA

3.3.4 Standards for wearable cuffless blood pressure measuring devices

Given the emerging and promising alternatives to cuff-based BP estimation, different associations established standards of evaluation to ensure the accuracy, reliability, and clinical acceptance of these proposed methods.

Key standards include those set by the IEEE (Institute of Electrical and Electronics Engineers), BHS (British Hypertension Society), and AAMI (Association for the Advancement of Medical Instrumentation). BHS and AAMI standards in particular have been frequently used by researchers to demonstrate the effectiveness and accuracy of their models in estimating BP.

IEEE Standard (IEEE Std 1708)

IEEE Std 1708 provides a comprehensive framework for the objective performance evaluation of wearable cuffless BP measuring devices. The standard is applicable to all types of wearable BP devices, including epidermal and unobtrusive devices. It specifies procedures for testing accuracy, repeatability and robustness of cuffless devices and includes guidelines for acceptable error margins in SBP and DBP readings [62].

British Hypertension Society (BHS) Standard

The BHS protocol provides a grading system based on the accuracy of the devices in measuring SBP and DBP. Devices are graded from A (highest accuracy) to D (lowest accuracy) based on the percentage of readings within specified error ranges as defined in Table 3.2 [63, 64].

Grade	Error ≤ 5 mmHg	Error ≤ 10 mmHg	Error ≤ 15 mmHg
A	60%	85%	95%
B	50%	75%	90%
C	40%	65%	85%

Table 3.2: BHS standard for BP measuring devices

Association for the Advancement of Medical Instrumentation (AAMI) Standard

The AAMI standard is widely recognised for the validation of BP measuring devices, including cuffless wearable devices. According to the AAMI protocol, a device is considered accurate if the mean difference between the device measurements and a reference standard is within ± 5 mmHg, with a standard deviation of less than 8 mmHg for both SBP and DBP. The evaluation process involves testing the device on a minimum of 85 subjects, ensuring a diverse range of age, gender, and blood pressure levels to reflect a general population [65].

Materials and Methods

This chapter describes the material and methods used to carry out this thesis' work.

Firstly, the publicly available datasets containing the required signals are described. Then, an overview of the selected benchmark models is presented. Finally, the training and testing approach that was followed for evaluating the models' performances is described, along with the metrics that were utilised to compare the performances.

4.1 Public Physiological Signals Datasets

A search was conducted to identify publicly available datasets containing PPG signals and BP measurements, to develop a model to estimate the BP values from the associated PPG signals.

Dataset Name	Signals	Subjects	Duration	Sampling Frequency
MIMIC-III	PPG, ECG, BP, others	> 30000	Variable	125 Hz
VitalDB	PPG, ECG, BP, others	6388	Variable	500 Hz
PulseDB (MIMIC-III + VitalDB)	PPG, ECG, BP, others	5361	10 seconds	125 Hz
Sensors (MIMIC-III)	PPG, BP	1196	15 seconds	125 Hz
UCI (MIMIC-II)	PPG, ECG, BP	N/A	Variable	125 Hz
Bed-Based BCG Dataset	BCG, ECG, PPG, BP	40	Variable	1 kHz
PPGBP	PPG, BP	219	2.1 seconds	125 Hz
Benchmark datasets (Sensors, UCI, BCG, PPGBP)	PPG, BP	Variable	5 or 2.1 seconds	125 Hz

Table 4.1: Characteristics of public datasets containing PPG and blood pressure signals

4 Materials and Methods

The selected datasets needed to have sufficiently long recordings of PPG signals and corresponding BP values measured at the same time instants, either in the form of a continuous ABP signal or as discrete values of systolic and diastolic pressure.

Table 4.1 summarises the main characteristics of the datasets that were found during the search. The table provides information on the measured signals, the number of subjects involved in the recordings, the duration of the acquired signals, and the sampling frequency.

Given the presented datasets, the following observations can be made:

- The MIMIC-III dataset (*Multi-parameter Intelligent Monitoring for Intensive Care*) and its previous versions contain signals acquired from Intensive Care Unit (ICU) patients at the Beth Israel Deaconess Medical Center in Boston. Since not all physiological signals were acquired for every patient, filtering is necessary to select subjects for whom both BP and PPG signals are available. Moreover, as stated in [66, 67, 68], the measured signals are not synchronised, since there could be delays for example due to signal acquisition and filtering. This means that MIMIC-III is not a suitable dataset for PAT-based BP estimation.
- The VitalDB database collects signals recorded at Seoul National University Hospital from patients undergoing routine or emergency non-cardiac surgeries. Similarly to the previous case, record filtering is required since not all signals are available for every patient [69].
- The PulseDB dataset compiles signals selected from the MIMIC-III and VitalDB databases by keeping the cleaned records containing both PPG, ECG and BP values, in order to establish a benchmark for BP estimation based on physiological signals. The obtained PulseDB dataset contains records of 2423 subjects from the MIMIC dataset and 2938 subjects from the VitalDB dataset. Specifically, the Authors provide a training subset of signals and BP labels along with different testing subsets to allow reproducible testing of DL models applied to BP estimation [70]. Moreover, a subset with only the VitalDB signals is provided by the Authors, since the MIMIC-III does not guarantee inter-waveform alignment of PPG and ECG signals, for testing models that use both signals.
- The Sensors and UCI datasets are respectively subsets of MIMIC-III and MIMIC-II and they are described in [71] and [72] and made available by the Authors.
- The Bed-Based Ballistocardiography Dataset collects ballistocardiograms (BCG) and time-aligned ECG, PPG and BP waveforms measured from 40 different subjects at Kansas State University [73].

- The PPGBP dataset provides three measurements of 2.1 seconds of PPG signal and discrete BP readings obtained from 219 subjects with different cardiovascular diseases at Guilin People’s Hospital [74].
- The benchmark datasets Sensors, UCI, BCG and PPGBP are presented in [52]. They are taken from datasets available online and described above, which are cleaned and processed in order to create a benchmark for BP estimation methods evaluation.

In particular, the MIMIC-III database and its previous version, MIMIC-II, are the most frequently used datasets in the literature, primarily due to the large number of available records.

For this study, the VitalDB subset from the PulseDB dataset along with the benchmark datasets (Sensors, UCI, BCG, and PPGBP) were selected to train and test different models for BP estimation from PPG. The selection was based on the availability of synchronised PPG and BP values, the quality of the preprocessed data, and the standardised validation strategies provided in these datasets.

4.1.1 VitalDB Subset of PulseDB

The Authors of the PulseDB dataset provide the signal segments, scripts and information needed to reproduce the subject-balanced training set and testing sets. However after trying to reproduce the dataset using Python, it was observed that in the files provided not all the needed segments are available, therefore the final dataset that was reproduced has a smaller number of records than the one described in the article and a different BP distribution. Nevertheless, the Authors provide the VitalDB subsets ready to download, so it was chosen to proceed with the VitalDB subset of PulseDB, in order to have a balanced training set and testing sets.

The Authors provide the training subset, named VitalDB Train, which contains 465480 segments recorded from 1293 subjects. Additionally, three testing subsets are made available.

The first one is a calibration based subset (VitalDB CalBased), since the subjects whose signals are included are the same subjects of the training subset, which means that during training the model can leverage information from these subjects to improve performance. This subset is useful for evaluating how well the model can generalise to seen subjects.

The second subset (VitalDB CalFree) is the calibration free subset, consisting in 57600 segments from unseen subjects in the training.

Finally, the third testing subset, Vital AAMI Test, consists in a smaller number of records of unseen subjects during training, whose BP labels distribution respects the AAMI standard described in Subsection 3.3.4.

The available training and testing sets obtained from VitalDB are described with their statistical characteristics in Table 4.2.

4 Materials and Methods

Each record contains a segment of PPG, ECG and continuous ABP signals of 10 seconds duration sampled at 125 Hz, so 1250 samples per signal, and the discrete BP values for the 10 seconds interval.

Subset Name	Dataset size	Demography (% Male & Age)	SBP (mmHg)	DBP (mmHg)
VitalDB Train	Subjects: 1293 Segments: 465480	57.70% 58.98 ± 15.03	Min: 38.51 Max: 286.58 Mean: 115.48 SD: 18.93	Min: 16.23 Max: 276.93 Mean: 62.92 SD: 12.08
VitalDB CalBased	Subjects: 1293 Segments: 51720	57.70% 58.98 ± 15.03	Min: 46.55 Max: 248.48 Mean: 115.50 SD: 18.85	Min: 22.46 Max: 241.00 Mean: 62.94 SD: 12.07
VitalDB CalFree	Subjects: 144 Segments: 57600	59.72% 58.17 ± 15.43	Min: 46.48 Max: 211.96 Mean: 115.43 SD: 18.72	Min: 22.98 Max: 193.57 Mean: 63.02 SD: 11.87
VitalDB AAMI Test	Subjects: 116 Segments: 666	59.16% 57.27 ± 13.50	Min: 72.82 Max: 214.23 Mean: 134.67 SD: 30.90	Min: 37.72 Max: 131.79 Mean: 75.39 SD: 18.22

Table 4.2: Characteristics of VitalDB subsets of the PulseDB database

4.1.2 Benchmark Datasets

The benchmark datasets denominated Sensors, UCI, BCG and PPG are compiled by González *et al.* in order to provide a baseline benchmark for PPG-based BP estimation using ML and DL methods [52]. The Authors observe that in the studies presented in literature, many researchers use arbitrary subsets taken from available public datasets of physiological signals, often without providing clear instructions to reproduce their result. Moreover many studies use the records from subjects present both in the training set and the testing set, which causes information leakage and overestimates the model’s real performance on unseen data.

Therefore, they endeavoured to create a benchmark with four preprocessed datasets with validation strategy, obtained from open datasets, in order to allow an easier model comparison.

The four datasets are the preprocessed version of the Sensors, UCI, Bed-Based Ballistocardiography and PPGBP datasets described in Table 4.1. The modified datasets contain records which have been selected according to signal quality and preprocessed with the same procedure for all the datasets.

Sensors, BCG and PPGBP datasets are divided by the Authors in 5 folds in order to carry out a 5-Fold Cross-Validation, while the UCI dataset, which is the biggest datasets among the selected datasets, is split in 3 sets for a Hold-One-Out strategy.

The statistical characteristics of each of the four datasets are presented in Table 4.3.

Dataset Name	Dataset size	Demography (% Male & Age)	Segment length (s)	SBP (mmHg)	DBP (mmHg)
Sensors	Subjects: 1195 Segments: 11102	59.8% 57.1 ± 14.2	5	Min: 81.84 Max: 198.66 Mean: 134.36 SD: 21.78	Min: 50.07 Max: 116.64 Mean: 65.37 SD: 10.51
UCI	Subjects: N/A Segments: 410596	N/A	5	Min: 64.45 Max: 199.66 Mean: 131.57 SD: 11.16	Min: 50.00 Max: 102.18 Mean: 66.79 SD: 10.48
BCG	Subjects: 40 Segments: 3063	44.5% 34.2 ± 14.5	5	Min: 71.75 Max: 191.07 Mean: 120.99 SD: 15.29	Min: 44.47 Max: 100.67 Mean: 67.23 SD: 9.30
PPGBP	Subjects: 218 Segments: 619	46.9% 56.9 ± 15.8	2.1	Min: 80.00 Max: 182.00 Mean: 128.02 SD: 20.50	Min: 42.00 Max: 107.00 Mean: 71.91 SD: 11.20

Table 4.3: Characteristics of the benchmark datasets

4.2 Blood Pressure Estimation Algorithms

As previously explained, different methods of BP estimation from the single PPG signal have been developed in the past decades. The aim of this work is to provide a comprehensive analysis of these methods, evaluating their effectiveness and potential limitations.

Following the literature review, different methods were tested on the previously described datasets. The first approach leverages ML algorithms which take features extracted from the PPG signal as input, then, various DL models were implemented, in order to compare the performances of traditional ML algorithms and DL models for BP estimation.

Each model that was implemented is described in the following paragraphs.

4.2.1 Feature Extraction

Building on the PPG features discussed in Section 3.3, various characteristics of the PPG signal have been extracted and utilised for regression with different ML algorithms. These features, summarised in Table 4.4, comprise of PPG waveform features and frequency-domain features. They were derived from the PPG signal through custom Python functions developed for feature extraction.

Time-domain features

The Python code that was developed extracts several time-domain features that describe the morphology and time intervals of the PPG waveform. These features include cardiac periods (determined from PPG systolic peak-to-peak intervals), peak amplitudes, systolic upstroke times, and diastolic times.

Additionally, the implemented function identifies and analyses fiducial points such as the Dicrotic Notch (DN), a and b peaks from the second derivative of the PPG signal, and computes ratios like the b/a ratio. Augmentation Index (AI) and the Large Artery Stiffness Index (LASI) are also derived from the relationship between peaks and the DN.

Figure 4.1 displays a PPG signal taken from the BCG dataset, with identified onsets, peaks, and the second derivative of the signal, highlighting key fiducial points such as the a and b peaks.

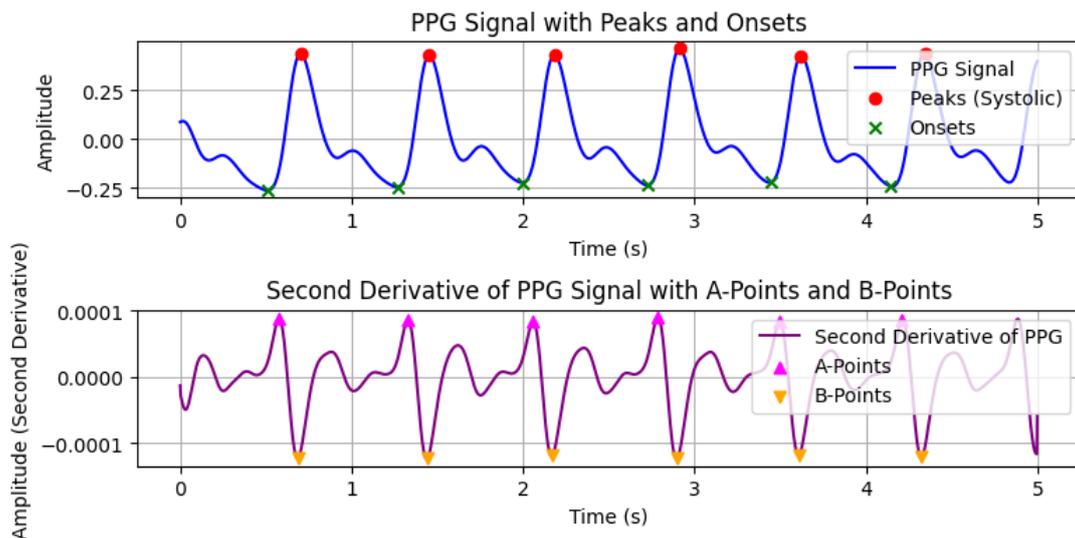


Figure 4.1: PPG waveform with detected peaks, onsets, and its second derivative showing fiducial points a and b

Moreover, the areas under specific parts of the PPG waveform, such as between the peak and DN, and from the DN to the next onset, are computed to capture information about the signal. Figure 4.2 shows the location of the DN within the PPG waveform. The position of this notch, along with its relationship to surrounding peaks, is used to compute the Augmentation Index and Large Artery Stiffness Index (LASI) and the areas under defined portions of the PPG waveform.

Other features of the PPG waveform are analysed by dividing the pulses into intervals before and after peaks at various percentages of the maximum amplitude (25%, 33%, 50%, 66%, and 75%). For each of these intervals, features like Systolic Width (SW), Diastolic Width (DW), total width, and the ratio between diastolic and systolic width are calculated.

Figure 4.3 illustrates the division of the PPG waveform into different intervals at 25%, 33%, 50%, 66%, and 75% of the maximum amplitude.

4 Materials and Methods

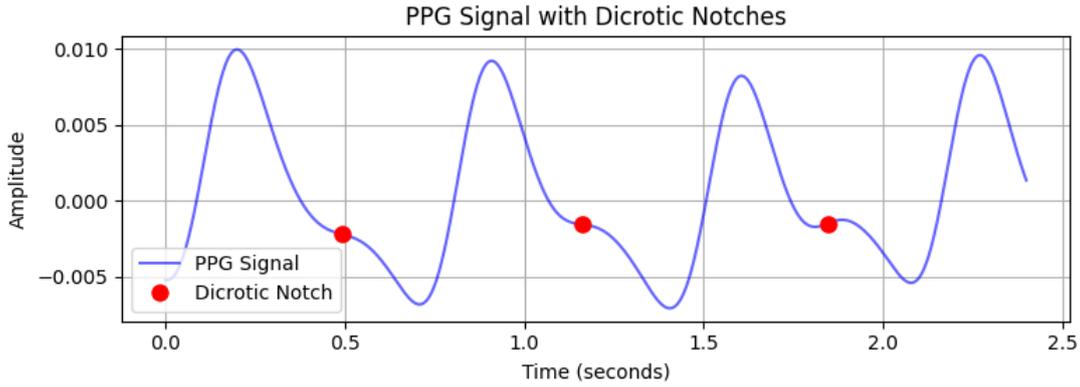


Figure 4.2: PPG waveform with the identification of the dicrotic notch

These metrics provide insights into the shape of the PPG waveform and the dynamics of blood volume changes, as described in Subsection 3.3.1.

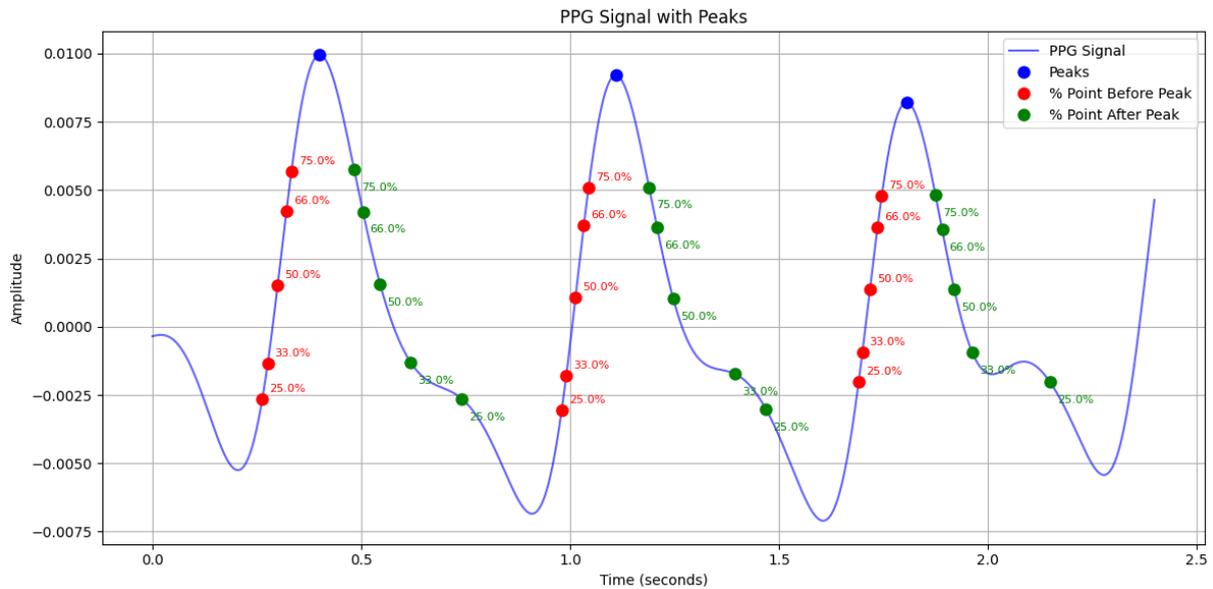


Figure 4.3: PPG waveform with intervals marked at 25%, 33%, 50%, 66%, and 75% of the maximum amplitude

Frequency-domain features

In addition to time-domain characteristics, the code extracts frequency-domain features using power spectral analysis. By applying the Welch method, the Power Spectral Density (PSD) of individual PPG beats is computed.

Features derived from the PSD include total power, power in very low frequency (VLF), low frequency (LF), and high frequency (HF) bands, as well as the LF/HF ratio.

Dominant frequency, spectral entropy, and spectral centroid are also extracted to characterise the frequency distribution and complexity of the PPG signal.

4 Materials and Methods

Furthermore, wavelet analysis is performed using discrete wavelet transforms with Daubechies wavelets (db4) to extract energy levels at different decomposition levels. These wavelet energy features capture localised changes and transients in the PPG signal.

Feature name	Feature description
Cardiac period	Time interval between two subsequent systolic peaks
Peak amplitude	Amplitude of the systolic peaks
Systolic time	Time interval between the PPG onset and systolic peak
Diastolic time	Time interval between the systolic peak and the onset
b/a ratio	Ratio between peak and minimum and maximum of the second derivative of PPG
Systolic Width	Interval between samples at 25%, 33%, 50%, 66% and 75% of the peak and the peak
Diastolic Width	Interval between the peak and samples at 25%, 33%, 50%, 66% and 75% of the peak
Total Width	Interval between samples at 25%, 33%, 50%, 66% and 75% before and after the systolic peak
Width ratio	Ratio of the diastolic width and corresponding systolic width at 25%, 33%, 50%, 66% and 75%
Augmentation index	Ratio between the systolic peak and the first DN after it
LASI (Large Artery Stiffness Index)	Interval between the systolic peak and the first DN after it
Peak-DN Area	Area under the curve between the peak and the DN
DN-Onset Area	Area under the curve between the DN and the onset
Mean	Average amplitude of the PPG signal
STD	Measure of the variability in the PPG signal within a beat
Skewness	Asymmetry of the PPG waveform distribution
Kurtosis	Degree of peakedness of the PPG waveform
Dominant frequency	Frequency component with the highest power
Power in frequency bands	Total power and power in the bands: 0.003 to 0.04 Hz, 0.04 to 0.15 Hz and 0.15 to 0.4 Hz
Spectral entropy	Measure of the spectral complexity of the PPG beat
Wavelet Energy	Energy of the wavelet coefficients

Table 4.4: PPG features summary

BP regression from PPG features

Four different ML models have been tested on PPG features for BP estimation, consisting in:

- MLP, implemented with 3 Linear layers using PyTorch. The architecture includes an input layer with input size matching the number of features, two hidden layers and an output layer that predicts the SBP and DBP values. The activation function used between layers is ReLU.

- Decision Trees, implemented with the *DecisionTreeRegressor* class of the scikit-learn Python library.
- Random Forest, implemented with the *RandomForestRegressor* class available in scikit-learn.
- SVR, implemented like the previous models, with scikit-learn. Two separate SVR models are used: one for SBP prediction and one for DBP prediction.

4.2.2 Neural Networks with Temporal Signal Input

Starting from the models discussed in the literature review presented in Section 3.3, various DL models which take the temporal PPG signal as input have been implemented, using Python and the PyTorch framework. The performance of these models has then been evaluated and compared as described in Chapter 5.

Convolutional Neural Network

The first model that was implemented consists in a simple 1D CNN, with three convolutional layers followed by a max pooling layer. The convolutional layers use a kernel size of 5 and progressively increase the number of output channels from 64 to 256, with padding to maintain the temporal dimension of the input signal. The pooling layers reduce the size of the feature maps by half after each convolution. After the convolutional and pooling layers, the output is flattened and given as input to three fully connected layers, which reduce the feature vector to predict the systolic and diastolic BP values. ReLU (Rectified Linear Unit) activation functions are applied after each convolutional layer and fully connected layer, to introduce non-linearity in the model.

Temporal Convolutional Network: TEMPONet

Another model that was applied to the BP estimation problem is TEMPONet, which is a TCN that was designed for biosignal based classification and is presented in [75]. For the scope of this work, TEMPONet was adapted to implement the regression of the two SBP and DBP values.

TEMPONet consists in three blocks, each composed of 2 temporal convolutional layers with kernel size 3 and a convolutional layer of kernel size 5 followed by an average pooling with kernel size 2. After each layer, the ReLU activation function and Batch-Normalisation are applied. In each block, the dilation rate progressively increases to 2, 4 and 8, while the stride follows the same pattern, set to 1, 2, and 4, respectively. Similar to a CNN, the output is then flattened and passed through fully connected layers to predict SBP and DBP.

CNN-LSTM Model

Another model that was implemented for this work is a hybrid CNN-LSTM model. Such model was built following the model described in [9], which is shown in Figure 4.4. In this work, since some datasets contain only the PPG signal without the synchronised ECG, also a model with only the CNN branch for PPG was implemented.

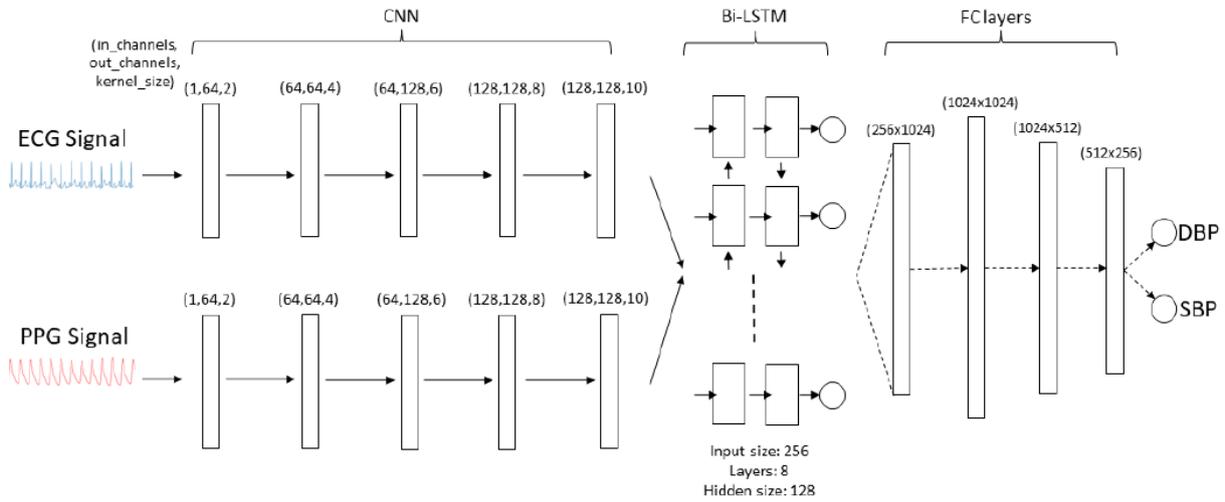


Figure 4.4: Hybrid CNN-LSTM network [9]

The implemented CNN-LSTM model consists of a convolutional block followed by an LSTM block and a fully connected layers block. The convolutional block consists of five convolutional layers with increasing channel sizes (from 64 to 256) and kernel sizes (respectively 3, 5, 7, 9, 11); the last three layers use a stride of 2 to reduce dimensionality. After each layer, Batch-Normalisation and ReLU are applied.

The LSTM block consists of a Bi-LSTM layer, which processes the output of the convolutional block and is followed by four linear layers that process the output BP predictions.

ResNet

ResNet is a deep CNN, which was initially developed for image recognition [76]. It incorporates residual connections, consisting in the addition of the input tensor to the output of some layers, making it easier for gradients to flow backward during training.

The ResNet-18 model that was used has the following architecture:

- Initial convolution layer, followed by batch normalisation and ReLU activation function;
- Residual blocks, lumped in four blocks, with an increasing number of output channels;
- Average pooling, that reduces the feature maps;
- Fully connected layer, which computed the output.

The residual building block of ResNet is shown in Figure 4.5.

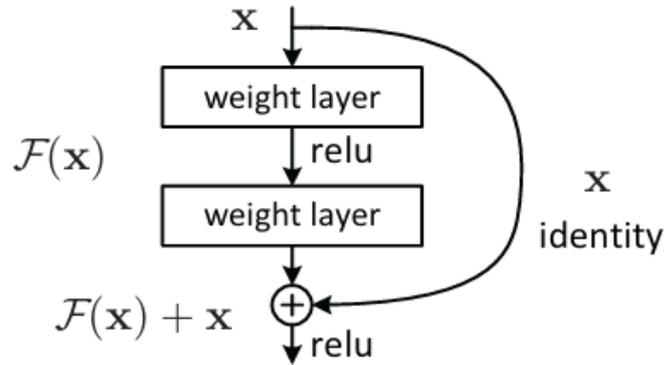


Figure 4.5: Residual block of the ResNet NN

The implementation of this NN was done by using the PyTorch torchvision library and adapting the ResNet model to accept 1D input tensors instead of images. By default, ResNet-18 expects 3-channel (RGB) images as input, however the 2D convolutional layers of the original ResNet were replaced with convolutional layers that accept 1D data, to capture temporal patterns in the sequential PPG data effectively.

UNet

UNet is a CNN that was designed for biomedical image segmentation [10]. It is a fully convolutional network, as it does not make use of fully connected layers, and it has an encoder-decoder structure, as depicted in Figure 4.6.

Similarly to the ResNet, also UNet can be adapted to 1D data; the 1D UNet is a signal-to-signal NN, which means that it can reconstruct a continuous output signal.

The architecture begins with an initial convolutional layer that extracts low-level features. The encoder follows, progressively reducing the temporal resolution while increasing the number of channels through stacked convolutional layers and downsampling operations. Each stage in the encoder consists of multiple convolutions followed by a downsampling step.

At the network's deepest point, the bottleneck captures high-level feature representations with additional convolutional layers. The decoder then reverses this process, using upsampling operations to gradually restore the original sequence resolution. Skip connections from corresponding encoder stages ensure fine-grained details are preserved.

The final output layer maps the reconstructed feature maps to the desired output shape.

For this work, the UNet was implemented using the 1D UNet presented in the Modern Deep Network Toolkits for PyTorch (MDNC) framework [77], modifying it by adding a MLP regressor at the output, which extrapolates the SBP and DBP predicted values.

4 Materials and Methods

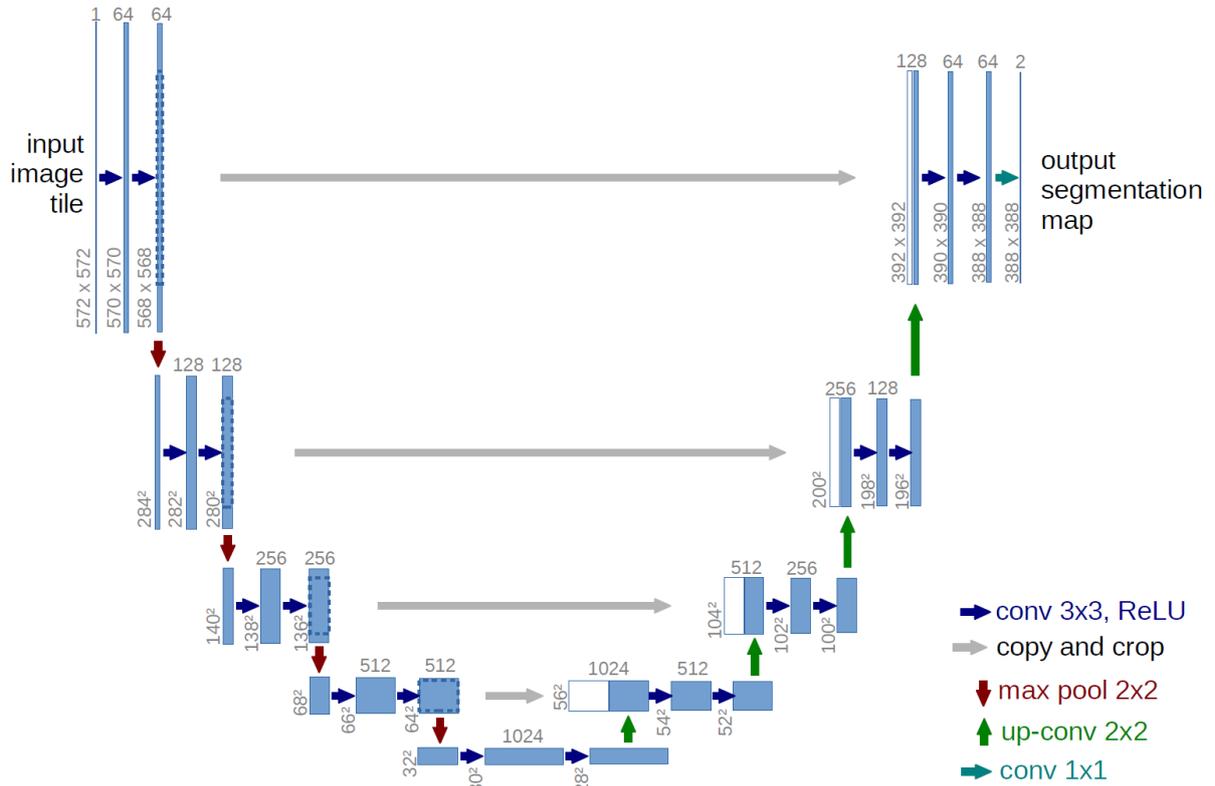


Figure 4.6: Original UNet architecture [10]

Paralleled CNN and Transformer network

Another model that was implemented is a Paralleled CNN and Transformer network (PCTN), which was presented in [11].

The PCTN is a hybrid DL architecture designed for BP estimation from PPG signals. This model integrates CNNs for local feature extraction and a transformer branch for capturing global dependencies, followed by a feature fusion module and a regression head to predict SBP and DPB values.

As depicted in Figure 4.7, the network begins with a Stem Module, which processes the input PPG segments using a 1D convolutional layer followed by batch normalisation and ReLU activation. This module extracts shallow features and reduces the input's temporal dimension. The extracted features are then passed through two parallel branches: a CNN Branch and a Transformer Branch. The CNN branch consists of a convolutional block that applies multiple convolutional layers, which use a combination of under-projection, spatial, and up-projection convolutions with residual connections to refine local features. Meanwhile, the Transformer Branch processes the extracted features by embedding the input into smaller patches, adding positional embeddings, and passing them through multiple Transformer Blocks, each containing multi-head self-attention mechanisms and feed-forward layers to capture long-range dependencies in the signal.

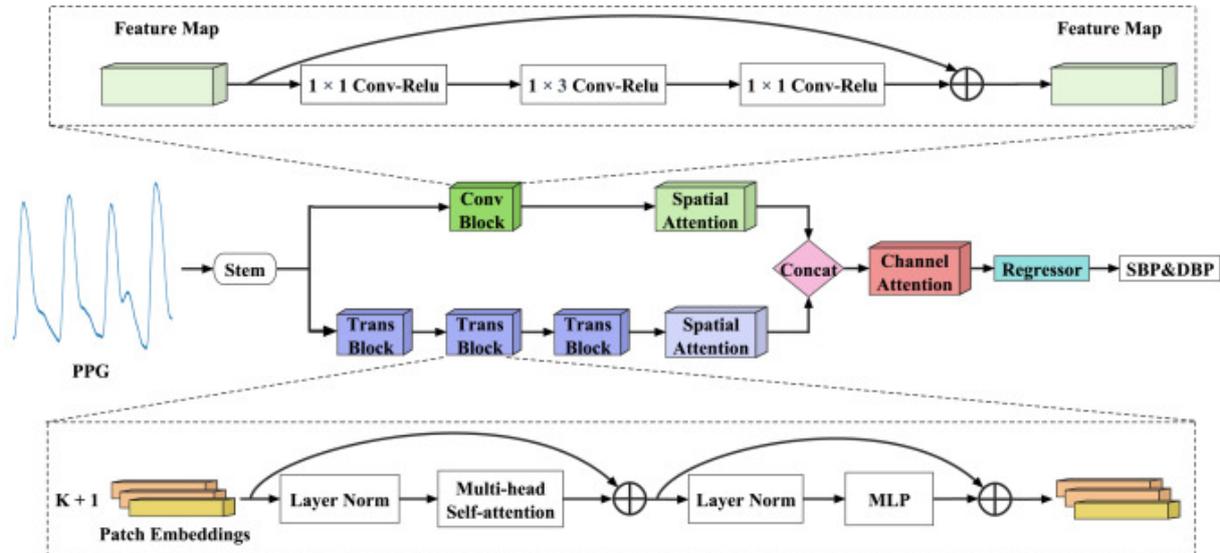


Figure 4.7: Paralleled CNN and Transformer network [11]

After feature extraction, the outputs from both branches are fused using CBAM (Convolutional Block Attention Module), which applies both spatial and channel attention mechanisms. Spatial attention enhances relevant regions of the signal, while channel attention emphasises important feature dimensions, ensuring a robust fusion of local and global representations. The fused features are then passed through a Regressor Module, which applies global average pooling followed by two fully connected layers with ReLU activations. The final output consists of the predicted systolic and diastolic BP values.

Transformer network

The next model that was implemented was a Transformer network, which was inspired by the PCTN but without the parallel CNN path. It consists in a Stem Module, a Transformer Branch and a Regressor module.

The Stem Module is a shallow feature extractor that applies a 1D convolution, batch normalisation, and ReLU activation to extract meaningful local features from the raw PPG input. This module reduces the dimensionality and prepares the data for further processing.

Next, the Transformer Branch models long-range dependencies within the extracted features. It utilises patch embeddings and a series of transformer blocks. Each transformer block contains multi-head self-attention, layer normalisation, and an MLP with residual connections.

After passing through the transformer, the extracted features are fed into the Regressor, a fully connected module that maps the high-dimensional features to the final BP predictions.

ResUNet + Attention

A more complex model that was implemented follows the architecture described in [12] and shown in Figure 4.8. It consists in the integration of a Residual UNet and self-attention mechanism for feature extraction.

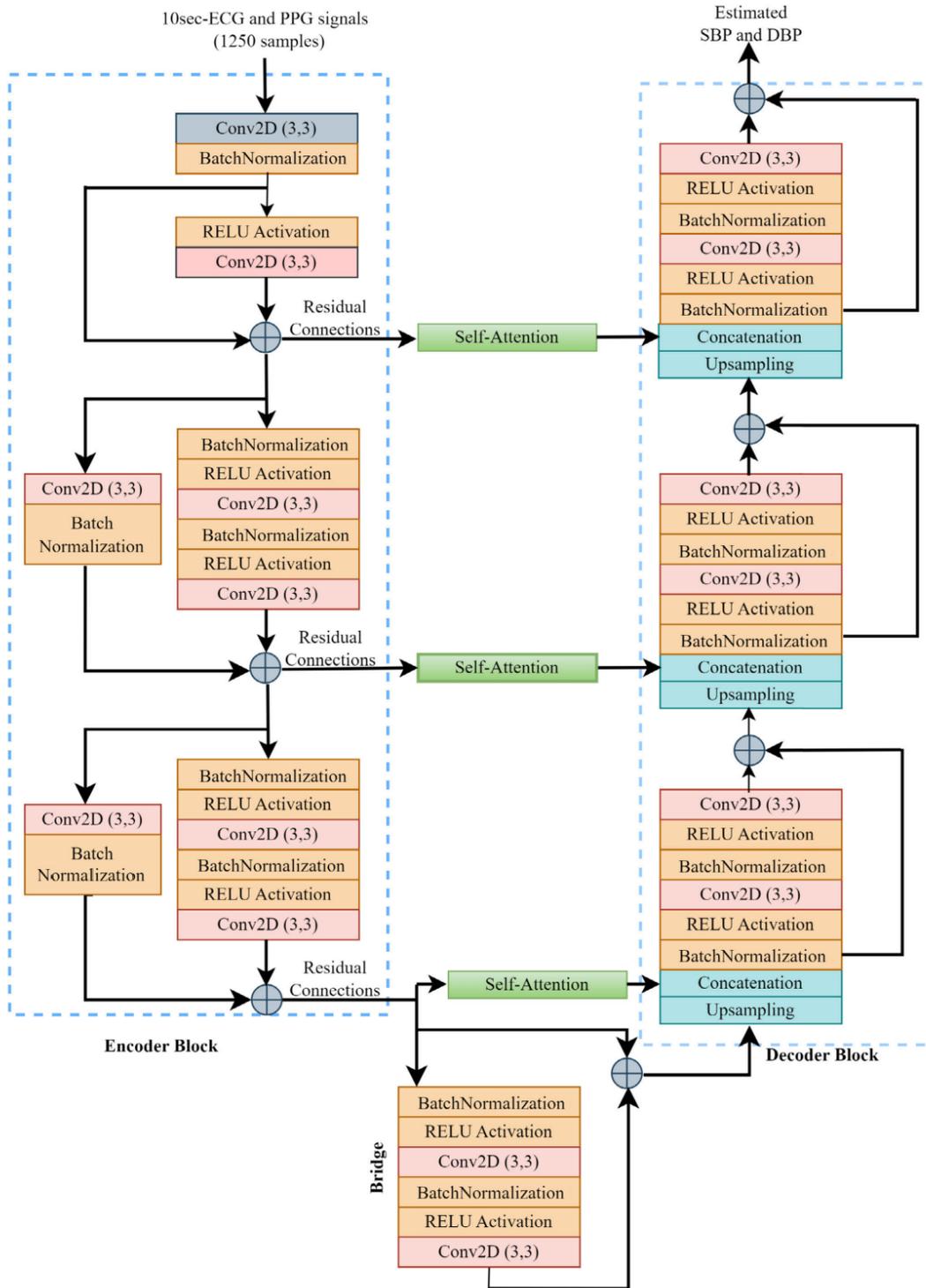


Figure 4.8: ResUNet + Attention network [12]

The model leverages a residual UNet, which combines the strengths of UNet for preserving high-level semantic features with residual connections that mitigate the vanishing gradient problem. The architecture is composed of a three-layer encoder, a bridge layer, and a three-layer decoder. Additionally, a self-attention mechanism is integrated in the skip connections of each encoder block before concatenation with the corresponding decoder features. This allows the network to focus on the most relevant spatial and temporal features by computing attention scores from query, key, and value matrices, following the formulation of the transformer architecture.

The model was implemented with PyTorch Lightning and processes 1D physiological signals, expands them into 2D with an initial convolutional layer of 64 channels, and applies a series of convolutional layers with 128, 256, and 512 channels, each followed by batch normalization and ReLU activation for hierarchical feature extraction.

The encoder gradually reduces spatial dimensions with convolutional blocks, each with kernel size 3 and stride 2, employing residual connections to retain information. A bridge block with 512 channels refines representations before passing them to the decoder.

The decoder progressively reconstructs the input structure using transposed convolutions with 512, 256, 128, and 64 channels, integrating skip connections from the corresponding encoder layers.

Self-attention layers with 2, 4 and 8 heads are incorporated after each encoding block to capture long-range dependencies. The final output is generated through a convolutional layer that aggregates the global features, producing two values representing SBP and DBP.

Foundation Model

The final model explored in this study was a transformer Foundation Model designed for time-series analysis, known as Moment, as described in [78]. FMs have gained significant attention in various domains due to their ability to learn generalisable representations from large datasets, which can then be fine-tuned for specific tasks. Given their success in natural language processing and computer vision, this research aimed to investigate whether such models could be effectively applied to BP estimation using PPG signals.

In this study, the pretrained Moment model was employed and adapted for regression by incorporating a Regression Head into its architecture. This modification allowed the model to predict SBP and DBP values directly from PPG-derived features.

The rationale behind using a FM was to leverage its ability to capture complex temporal dependencies and feature representations in time-series data, potentially improving prediction accuracy compared to conventional ML models.

The evaluation of this approach involved fine-tuning the pretrained model on a dataset of PPG signals paired with reference BP measurements.

4.3 Training and Testing

This section presents the approach that was followed to train and test the various prediction models, including the data normalisation approach, the implementation of the models and the computational resources utilised.

4.3.1 Data normalisation

The PPG signals obtained from the datasets were normalised with a min-max approach, as shown in Equation 4.1, so that the all the signals are scaled between 0 and 1. The minimum and maximum of the PPG signal are computed across the training sets and the same values are used on the testing sets, to ensure consistent scaling and prevent data leakage between training and testing phases. This guarantees that the normalisation process does not incorporate information from the test set into the training phase, maintaining the integrity and generalisability of the model's performance evaluation.

The min-max normalisation is defined as:

$$PPG_{norm} = \frac{PPG - PPG_{min}}{PPG_{max} - PPG_{min}} \quad (4.1)$$

Where:

- PPG_{norm} is the normalised signal value;
- PPG is the original signal value;
- PPG_{min} and PPG_{max} are the minimum and maximum values of the signal, respectively.

The same min-max normalization approach was applied to the extracted features for feature-based ML models, ensuring consistent data scaling across different inputs.

The normalisation for features was defined as:

$$Feature_{norm} = \frac{Feature - Feature_{min}}{Feature_{max} - Feature_{min}} \quad (4.2)$$

Where:

- $Feature_{norm}$ represents the normalised feature value.
- $Feature$ is the original extracted feature value.
- $Feature_{min}$ and $Feature_{max}$ are the minimum and maximum values of the feature.

4.3.2 Model implementation and training

The DL models were implemented in Python, with the PyTorch library and PyTorch Lightning framework to allow for faster training of the models using multi-GPU training.

Each DL model has been trained and tested with the same approach, using:

- Adam optimiser with a starting learning rate of 0.0001;
- Tensorboard logger to monitor the training metrics;
- Maximum 150 epochs with an early stopping strategy and 10 epochs patience, to prevent overfitting to the training data.

Given the substantial size of some of the datasets that were used, particularly the VitalDB subsets, it was necessary to leverage high-performance computing resources. Therefore, the Leonardo supercomputer was utilised to efficiently train the models. In fact, Leonardo is a pre-exascale supercomputer hosted by CINECA and built in the Bologna Technopole, which is one of the most powerful computing infrastructures in Europe [79]. As part of the EuroHPC initiative, Leonardo provides cutting-edge computational capabilities, enabling researchers to tackle complex scientific and industrial challenges.

In particular, Leonardo’s system consists of two main compute blade architectures [80]:

- X2135 GPU Blade, equipped with NVIDIA Ampere A100-64 accelerators, available through the Leonardo Booster partition.
- X2140 CPU-only Blade, based on Intel Sapphire Rapids processors, available through the Leonardo Data Centric General Purpose (DCGP) partition, suitable for CPU-intensive computations.

In this work, Leonardo’s capabilities were exploited, thanks to its high-speed storage system that facilitated the handling of large datasets, ensuring efficient data loading and processing, and the computational power to handle large-scale data processing and model training.

Specifically, the GPU Booster partition was employed, since it provided the necessary computational resources to efficiently train DL models, as GPUs are particularly well-suited for this task due to their parallel processing capabilities, which significantly accelerate matrix operations and backpropagation, making them ideal for handling the intensive computations required.

The Python-based training pipeline was executed on Leonardo’s GPU nodes, enabling accelerated computations and reducing training times significantly. This approach allowed for the training of DL models with optimised parallel processing, leveraging the PyTorch Lightning framework for multi-GPU training.

The use of supercomputing resources was particularly crucial for training Transformer-based architectures, which require extensive memory and computational power due to their self-attention mechanisms and number of parameters.

To efficiently manage computational workloads, the SLURM scheduler is used to submit and queue jobs on the Leonardo system, which runs on Red Hat Enterprise Linux 8.6.

Due to the shared nature of the system, jobs were scheduled based on resource availability, ensuring fair access to the compute nodes. The jobs were executed in Batch Mode, which required preparing shell scripts specifying the operations to be performed once resources were allocated.

4.4 Evaluation Metrics

Different metrics can be used for the evaluation of the model performance for regression tasks.

For this work, Mean Squared Error (MSE) was used for the training loss function. MSE is defined by Equation 4.3, where N is the number of predicted values, y_i is the actual value and \hat{y}_i is the predicted value.

MSE measures the mean of the squared differences between the predicted values and the labels, it is more sensitive to large errors, therefore it is more sensitive to outliers.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.3)$$

To compare the model's performance to the ones presented in the literature review, the Mean Absolute Error (MAE) metric was computed during the testing, since it is one of the most popular metrics for this problem. The reason why it is used frequently is because the error units are the same of the target label units, in this case the blood pressure unit mmHg so it is more easily interpreted. Additionally, since it is computed as the average of the absolute error between real and predicted values as shown in Equation 4.4, the MAE values increase linearly with the errors.

Since the two distinct value of systolic and diastolic blood pressure are computed from the models, the MAE equation is specialised for the two separate values as indicated in Equations 4.5 and 4.6 , considering that it is interesting to verify which quantity is better predicted.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (4.4)$$

$$MAE_{SBP} = \frac{1}{N} \sum_{i=1}^N |y_{SBP,i} - \hat{y}_{SBP,i}| \quad (4.5)$$

4 Materials and Methods

$$MAE_{DBP} = \frac{1}{N} \sum_{i=1}^N |y_{DBP,i} - \hat{y}_{DBP,i}| \quad (4.6)$$

Another interesting metric to compute is the Standard Deviation of Errors (STD), which gives an indication on the variability of the errors between the real and predicted values.

The general STD equation is shown in Equation 4.7, where e_i is the difference between each label and predicted value and \bar{e} is the Mean Error (ME), computed like in Equation 4.8.

Moreover, Equations 4.9 and 4.10 show how the STD is computed separately for SBP and DBP values.

$$STD = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_i - \bar{e})^2} \quad (4.7)$$

$$ME = \bar{e} = \frac{1}{N} \sum_{i=1}^N e_i \quad (4.8)$$

$$STD_{SBP} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_{SBP,i} - \bar{e}_{SBP})^2} \quad (4.9)$$

$$STD_{DBP} = \sqrt{\frac{1}{N} \sum_{i=1}^N (e_{DBP,i} - \bar{e}_{DBP})^2} \quad (4.10)$$

Results

This chapter presents the BP estimation results on the test datasets, obtained from the models described in Section 4.2. The performance of each model is evaluated on a range of metrics presented in Section 4.4.

5.1 Comparison Between Models

Each model was trained and tested on VitalDB, UCI, Sensors, BCG and PPGBP datasets.

In the case of VitalDB, the subsets derived from PulseDB were used. Each model was trained on the VitalDB Train Subset and tested on each of the three testing subsets, which are described in Subsection 4.1.1.

Considering UCI, Sensors, BCG and PPGBP datasets, which are available from the benchmark described in [52], a different approach was used. As these datasets are provided already divided in 5 folds in the case of Sensors, BCG and PPGBP, and 3 sets in the case of UCI, each model was trained leaving one set out and testing on the remaining fold. Then, the metrics obtained from the different tests have been averaged.

For each testing set, MAE, ME and STD were computed for SBP and DBP separately, in order to assess the model's performances in estimating the two BP values.

Furthermore, to verify the accuracy of the predictions and verify how closely the model is able to correctly predict the BP, different plots were generated from the test data.

The first plot consists in a scatter plot which maps the real and predicted values of SBP and DBP, it includes also the ideal prediction line to allow a visual assessment of the accuracy and variance in the predictions.

The second type of plot consists in a histogram, which overlays the real distribution of BP and the predicted one, in order to verify the presence of possible bias in the prediction or imbalance in the distribution.

5 Results

The third plot is another histogram, which represents the error distribution. This helps identify if the errors are centered around zero or if the model is biased, additionally it can help identify if the model consistently overpredicts or underpredicts the BP values. Moreover, it can also help visualise the variance in the errors.

5.1.1 VitalDB results

Table 5.1 summarises the performances of the different methods on the VitalDB subsets.

Model	VitalDB_CalBased			VitalDB_CalFree			VitalDB_AAMI		
	MAE	ME	STD	MAE	ME	STD	MAE	ME	STD
Feat. with MLP	Sys: 12.78	-0.55	16.24	13.09	-0.36	16.53	20.87	-10.30	24.68
	Dias: 8.35	0.10	10.62	8.32	0.12	10.50	13.21	-8.32	14.89
Feat. with Decision Tree	Sys: 14.38	0.02	20.08	18.02	-1.20	23.04	25.03	-11.15	30.28
	Dias: 9.32	-0.10	13.04	11.63	-0.57	14.75	15.86	-9.02	18.61
Feat. with Random Forest	Sys: 10.14	0.56	13.56	13.01	-1.34	12.46	20.71	-11.03	16.48
	Dias: 6.60	0.20	8.85	8.30	0.31	7.00	13.22	-7.99	10.48
Feat. with SVR	Sys: 18.52	-3.78	22.77	19.49	-7.89	23.74	29.01	-12.87	30.00
	Dias: 16.96	-2.34	20.77	17.41	-3.04	21.39	24.01	-9.12	26.47
CNN	Sys: 12.21	-0.1	13.77	13.45	-0.42	12.46	19.50	-10.80	22.51
	Dias: 7.98	0.01	8.46	8.44	-0.29	7.00	12.72	-7.55	13.69
CNN with ECG	Sys: 9.84	-0.60	12.13	13.94	-1.07	13.03	20.14	-12.6	22.8
	Dias: 6.51	-0.25	7.70	9.08	-0.68	7.91	13.46	-8.63	14.63
CNN-LSTM	Sys: 8.88	0.67	11.06	13.77	0.42	12.61	18.78	-8.06	22.8
	Dias: 5.89	0.73	6.98	8.63	0.53	7.27	11.55	-5.67	13.01
CNN-LSTM with ECG	Sys: 5.33	-0.20	7.29	13.27	-1.44	11.67	17.43	-8.02	20.10
	Dias: 3.42	-0.08	4.69	8.76	-0.24	7.25	11.48	-5.53	12.95
TCN	Sys: 10.21	-0.31	12.02	13.33	-1.54	12.05	18.76	-9.06	21.88
	Dias: 6.83	-0.22	7.64	8.43	-0.42	7.08	12.08	-6.59	13.19
TCN with ECG	Sys: 7.92	-1.25	9.97	13.21	-2.60	11.86	18.29	-10.54	20.82
	Dias: 5.49	-0.56	6.63	8.88	1.22	7.42	12.25	-7.57	12.80
Transformer	Sys: 11.84	0.71	12.75	12.61	0.38	10.93	18.34	-9.7	21.59
	Dias: 7.78	0.10	7.89	8.04	-0.27	6.19	12.04	-7.40	12.79
Transformer with ECG	Sys: 8.88	-0.40	10.94	13.52	-1.58	12.13	18.97	-11.55	21.59
	Dias: 5.98	0.19	7.09	8.83	-0.40	7.42	12.79	-7.42	14.01
PCTN	Sys: 9.66	0.06	11.84	12.65	-2.11	10.75	20.46	-14.06	22.50
	Dias: 6.56	-0.01	7.58	8.15	-1.85	6.05	13.41	-9.79	13.28
ResNet	Sys: 9.67	-2.34	11.47	13.71	-2.41	12.19	19.06	-11.23	21.64
	Dias: 6.30	-0.78	7.31	8.55	-0.85	7.15	12.16	-7.24	12.93
UNet	Sys: 12.73	0.64	13.44	12.82	0.21	11.18	21.31	-13.96	23.39
	Dias: 8.42	-0.09	8.34	8.31	-0.42	6.23	13.77	-9.98	14.45
ResUNet + Attention	Sys: 8.29	-0.05	9.86	13.72	-0.35	12.31	18.42	-8.65	21.02
	Dias: 5.65	-0.18	6.22	8.63	-0.47	7.12	11.89	-6.28	12.77
ResUNet + Attention with ECG	Sys: 5.60	0.28	7.36	13.69	-1.05	11.34	17.12	-7.90	20.54
	Dias: 3.80	0.08	4.88	8.91	-0.67	7.00	11.78	-6.04	13.15
Moment	Sys: 10.47	0.03	12.90	14.20	0.07	13.59	19.81	-10.17	23.29
	Dias: 6.99	0.03	7.99	8.70	-0.10	7.65	12.64	-7.09	13.81

Table 5.1: Model performances on the VitalDB subsets

5 Results

Considering the results across the different test sets, for all the various implemented models, it can be observed that generally the resulting prediction error is significantly lower in the calibration based testing, compared to the two calibration free testing cases. This suggests that calibration on subject data significantly enhances BP estimation accuracy.

Model performance varies significantly across the three subsets of the VitalDB dataset, in particular no single model consistently outperforms the others across all testing sets.

Considering the Calibration Based subset, the best overall performance for both SBP and DBP was achieved with the ResUNet + Attention model, with a MAE of respectively 8.29 mmHg and 5.65 mmHg without the ECG data and 5.33 mmHg and 3.42 mmHg with the ECG data, using the CNN-LSTM model.

Among feature-based methods, Random Forest demonstrated the lowest MAE (10.14 for SBP and 6.60 for DBP). However, SVR exhibited the worst performance, with a MAE of 18.52 (SBP) and 16.96 (DBP). This trend remains consistent across all three subsets of VitalDB, demonstrating limited generalizability of SVR for BP estimation over large datasets.

The best model for the Calibration Free subset was the Transformer model, with a resulting MAE of 12.61 mmHg for SBP and 8.04 mmHg for DBP.

Finally, in the case of the AAMI subset, the Transformer model achieved the lowest MAE for SBP for data without the ECG, with 18.24 mmHg, while the ResUNet + Attention architecture achieved 17.12 mmHg when ECG was included. In the case of DBP, the best performing model was CNN-LSTM both with and without ECG, with MAE of respectively 11.48 and 11.55 mmHg.

As a whole, models utilising additional ECG data (e.g., CNN-LSTM with ECG, ResUNet + Attention with ECG) tend to outperform their counterparts that rely solely on PPG signals, especially considering the calibration based approach. This improvement is particularly noticeable in SBP estimation, which is more difficult to estimate compared to DBP, since it can be observed that generally the prediction results in much higher errors for SBP compared to DBP.

The CNN-LSTM and ResUNet + Attention architectures consistently achieve lower errors compared to simpler CNN and UNet models, highlighting the benefits of combining convolutional and sequential processing capabilities. Transformer-based models also demonstrated strong performance, particularly in the calibration-free setting.

Regarding the performances of the FM Moment model, it can be observed that the increase in model complexity and number of parameters did not correspond to a reduction of the prediction errors, as the model does not perform significantly better than other DL models.

5 Results

Despite being fine-tuned, leveraging the pretrained knowledge alone was not sufficient to achieve superior performance in BP estimation. This suggests that while pretrained FMs may offer generalisable features, their adaptation to specific physiological signal tasks like BP estimation requires further optimisation beyond standard fine-tuning.

In Figure 5.1 the plots of the predictions from the Transformer model on Vital AAMI Test are shown. They represent the best results on that test subset with the use of the single PPG. Whereas in Figure 5.2 are shown the results on the same subset obtained from the ResUNet + Attention model, which includes the ECG and gives the best SBP estimation among the models that jointly use PPG and ECG.

It can be observed that in the case of the model that includes the ECG, the distribution of the predicted SBP and DBP values matches better with the real distribution. One reason why the distributions do not match perfectly, particularly in the range of 115-130 mmHg for SBP, could be due to the fact that the Vital AAMI Test subset follows the AAMI Standard, which requires a certain proportion of hypertensive and hypotensive BP measurements; the resulting distribution does not necessarily match the training set distribution, so the model has difficulty in correctly predicting some BP values. Specifically, the model is less accurate in the prediction of highest and lowest SBP and DBP values, which can be seen since some of the points representing the predictions in Figure 5.2 stay below or above the ideal prediction line and the distributions of the predicted SBP and DBP values has a smaller width compared to the real distribution of values in the test set.

Moreover, in Figure 5.3 are shown the prediction results for the VitalDB Calibration Free subset using the Transformer model, which gave the lowest prediction metrics on this subset.

It can be observed that the same problem of predicting the extreme SBP and DBP values in the distribution persists, even with this different test set. This could be due in part to the fact that these labels are not sufficiently represented in the training set, so the models don't learn to generalise well for extreme BP values.

Additionally, Figure 5.4 shows the prediction results for the VitalDB Calibration Based subset obtained with the CNN-LSTM model without ECG. In this case, we can see that the predictions follow more closely the ideal distribution compared to the Calibration Free case, so using calibration and therefore having the same subjects in both the training and testing data can significantly lower the prediction errors.

Finally, Figure 5.5 displays the results obtained using the CNN-LSTM model with the ECG signal on the same test subset. It can be noted that adding the ECG signal remarkably improves the prediction, both in terms of the metrics reported in Table 5.1 both in the distribution of the predicted values.

5 Results

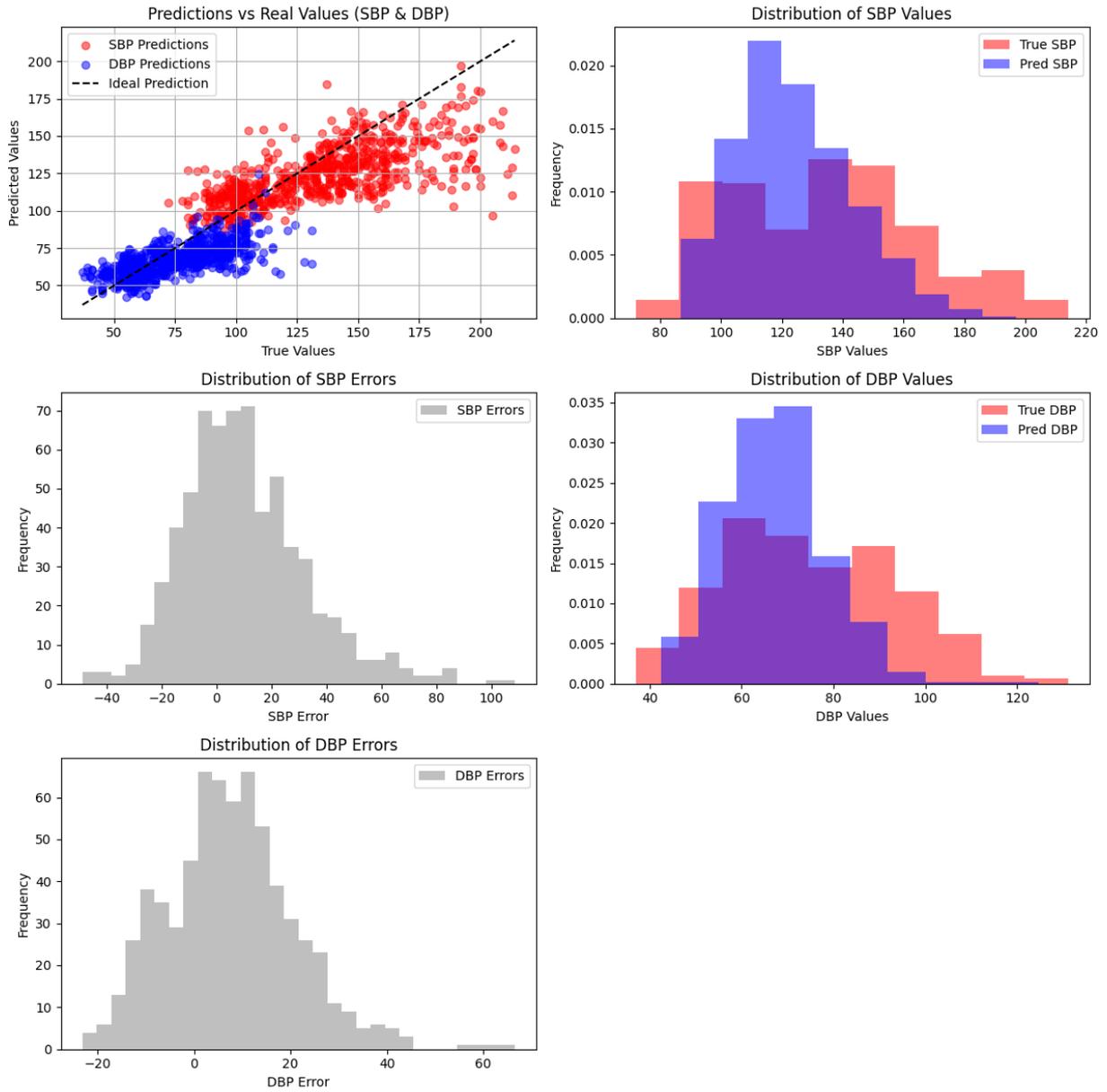


Figure 5.1: Prediction results on Vital AAMI Test subset using the Transformer-based model

5 Results

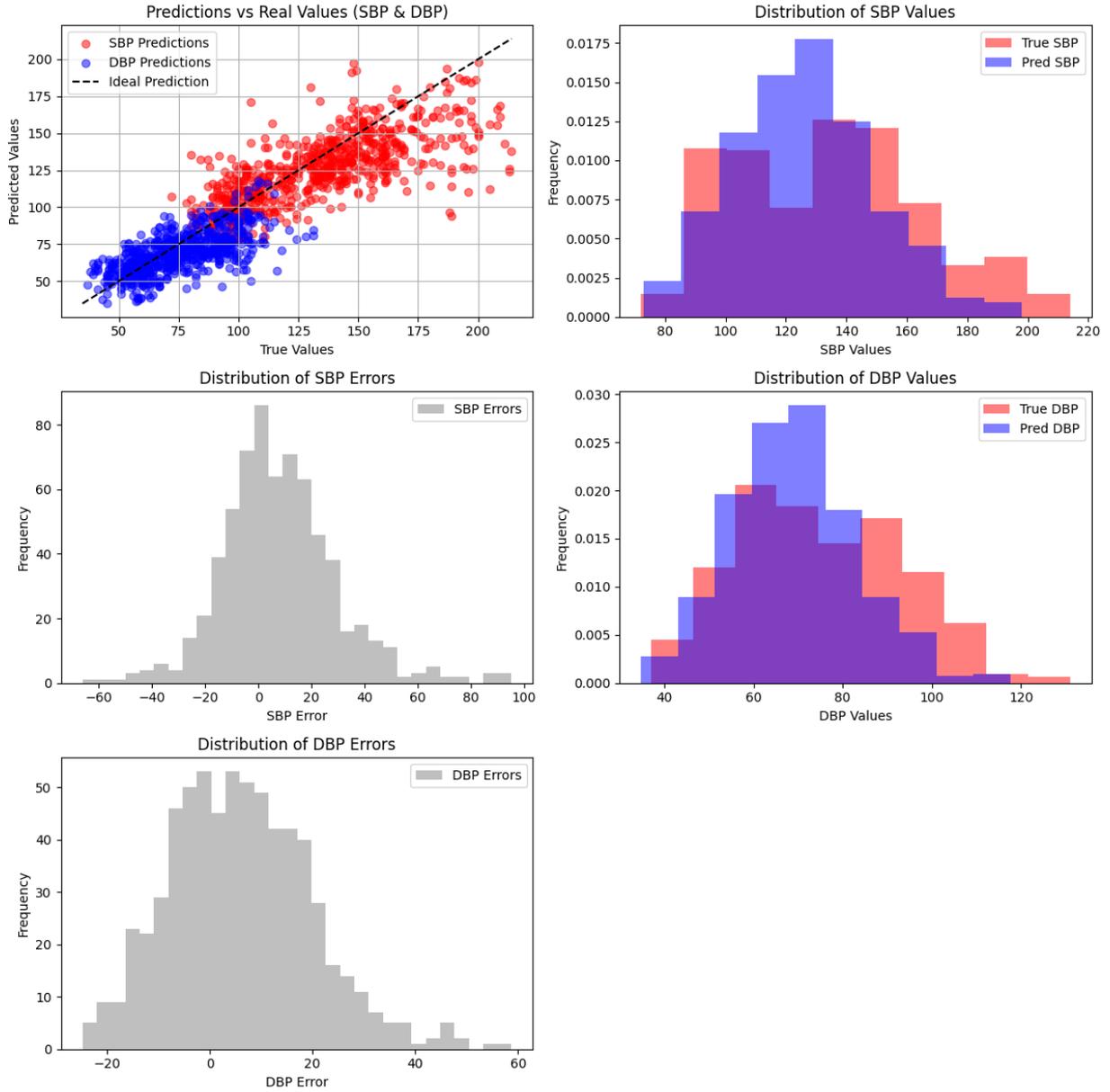


Figure 5.2: Results on Vital AAMI Test subset using the ResUNet + Attention with ECG

5 Results

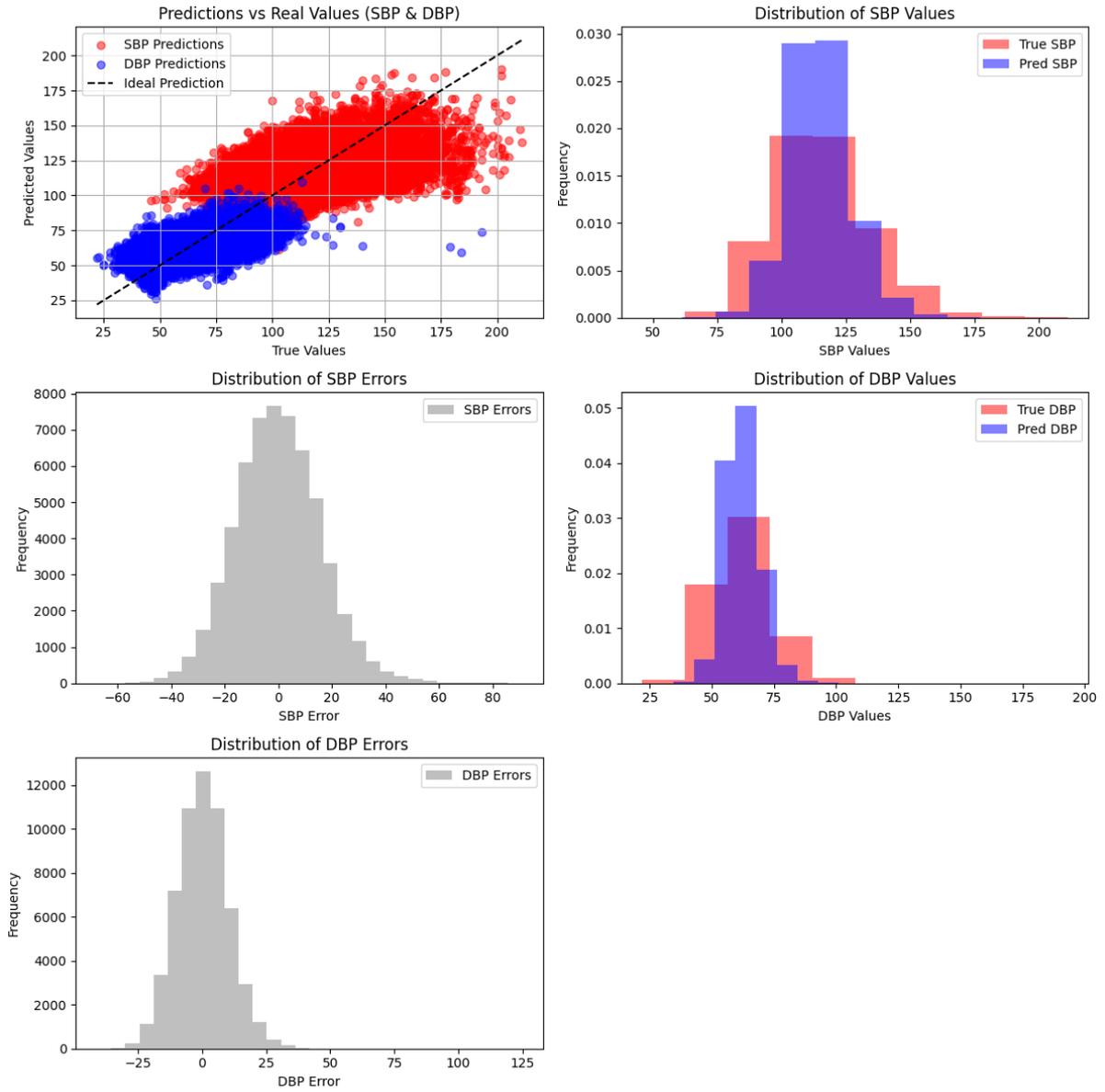


Figure 5.3: Results on Vital CalFree subset with the Transformer model

5 Results

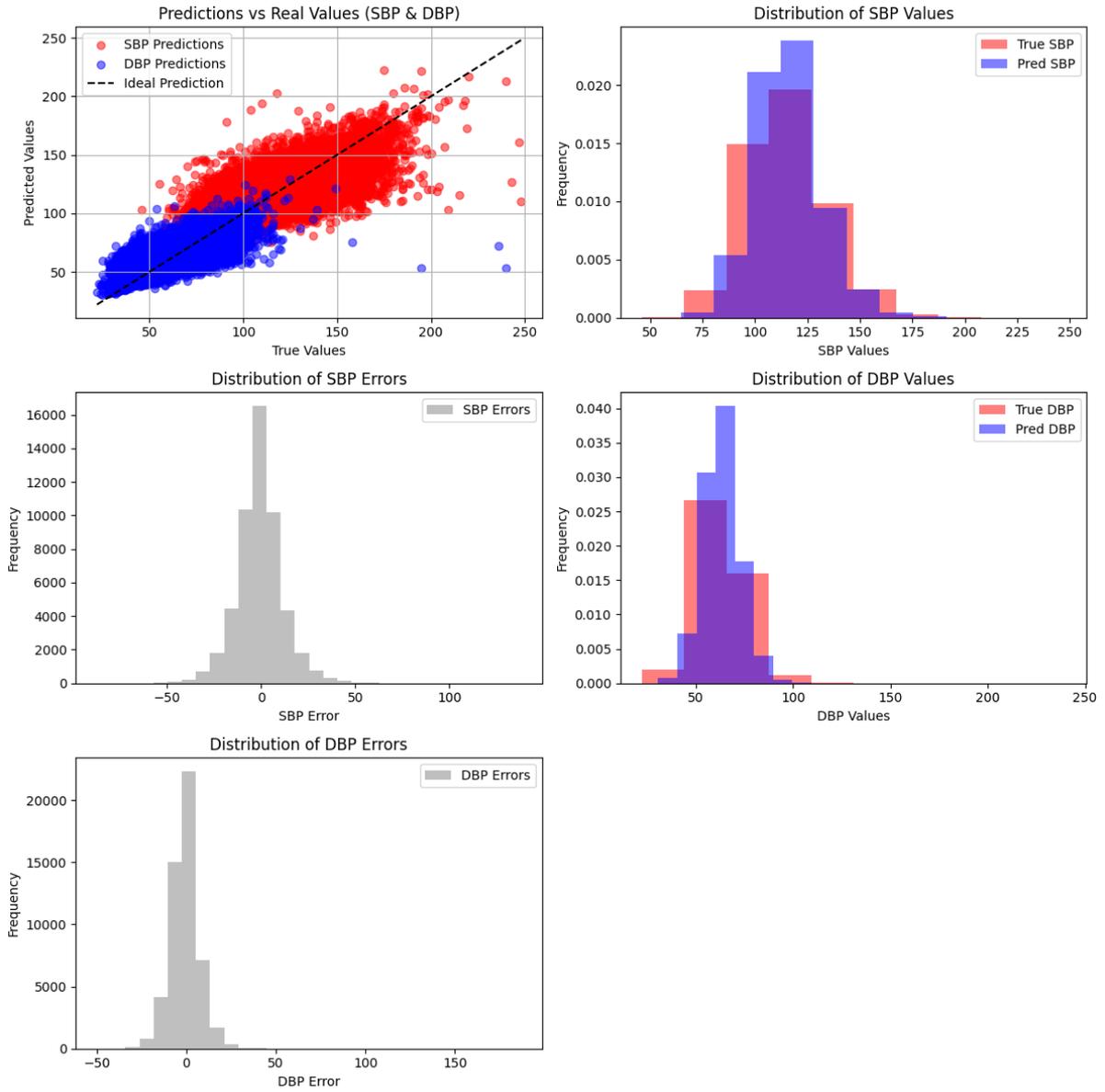


Figure 5.4: Results on Vital CalBased subset with the CNN-LSTM model with only the PPG

5 Results

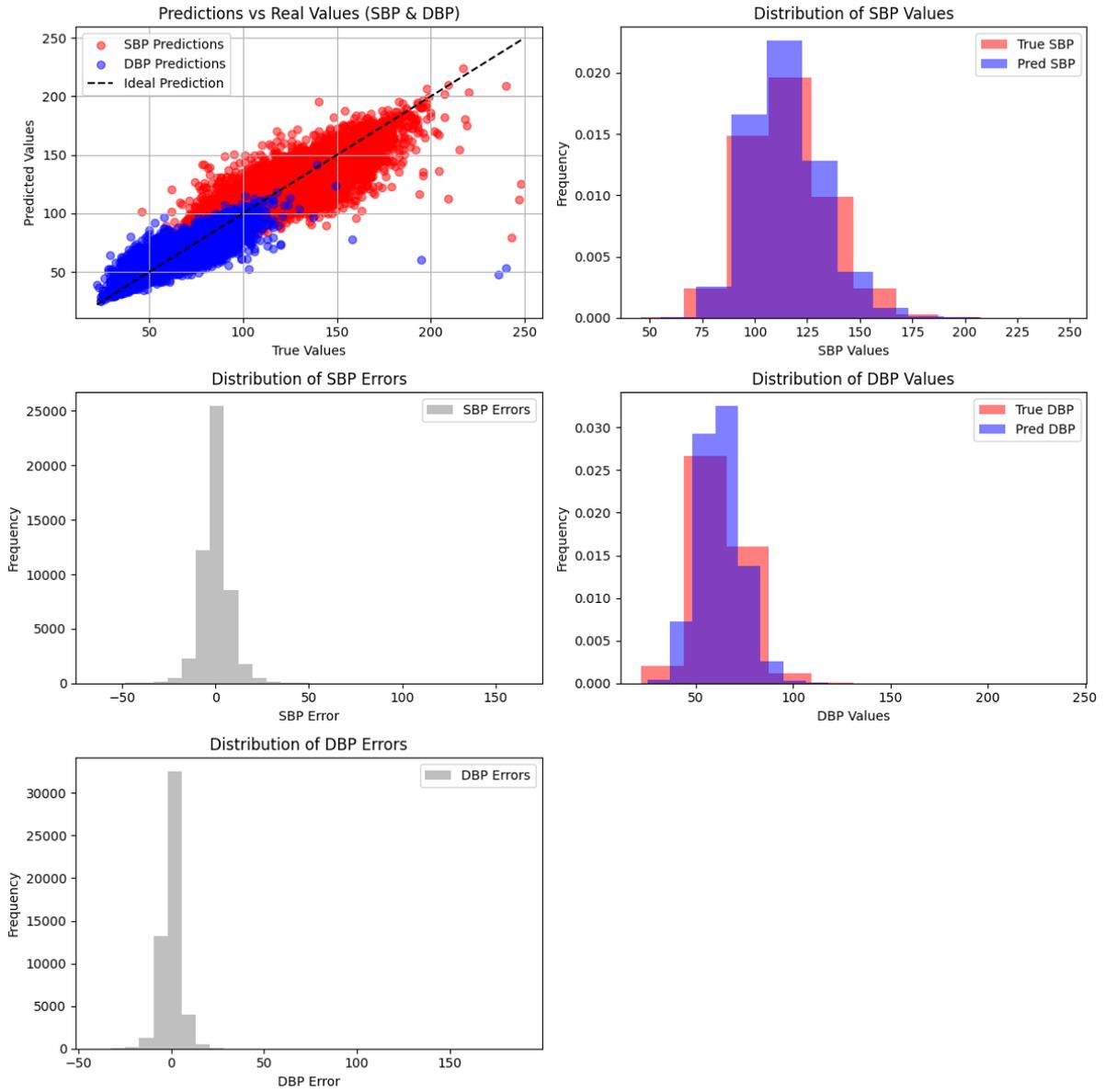


Figure 5.5: Results on Vital CalBased subset with the CNN-LSTM model with ECG

Model size

The size of each DL model, in terms of both the number of parameters and memory footprint, is a crucial factor in evaluating computational efficiency and feasibility for real-world deployment. Table 5.2 summarises these aspects for the different models analysed in this work.

Model	Number of Parameters	Size in Memory (MB)
CNN	5.3 M	21.304
CNN-LSTM	3.7 M	14.700
TCN	406 K	1.626
Transformer	1.5 M	5.934
PCTN	1.9 M	7.521
ResNet	11.2 M	44.685
UNet	80.6 K	0.322
ResUNet + Attention	6.7 M	26.608
Moment	35.5 M	141.877

Table 5.2: Number of parameters and memory size for the different DL models

Among the tested architectures, ResNet and Moment exhibit the highest number of parameters, with 11.2 M and 35.5 M, respectively, leading to a considerable memory footprint.

Notably, some of the best-performing models (CNN-LSTM, Transformer, and ResUNet + Attention) achieve a balanced trade-off between model size and effectiveness.

CNN-LSTM, with 3.7 M parameters and a memory footprint of 14.7 MB, demonstrates a strong capacity for sequence modeling while maintaining moderate computational requirements.

The Transformer architecture, known for its efficiency in handling long-range dependencies, has a relatively compact size of 1.5 M parameters and requires only 5.9 MB in memory.

ResUNet + Attention, despite its more complex structure incorporating attention mechanisms, remains computationally manageable with 6.7 M parameters and 26.6 MB in memory, benefiting from its ability to focus on relevant features while maintaining efficiency.

5.1.2 Benchmark datasets results

In the case of the smaller benchmark datasets (UCI, Sensors, BCG and PPGBP), it was observed that the deep models used are too complex to be adequately trained due to the limited size of these datasets. The insufficient amount of training data leads to overfitting and prevents the models from effectively learning the underlying patterns necessary for accurate BP estimation. To address this issue, a transfer learning approach was adopted: the models were first pre-trained on the larger VitalDB Training set and subsequently these pre-trained models were fine-tuned

5 Results

on the smaller datasets. This allows the models to exploit the information acquired from the larger dataset and employ it with the smaller datasets.

Table 5.3 summarises the performances of the different methods on the benchmark datasets.

Model	UCI			Sensors			BCG			PPGBP		
	MAE	ME	STD	MAE	ME	STD	MAE	ME	STD	MAE	ME	STD
Feat. with MLP	Sys: 17.81	-0.03	21.80	16.62	2.09	20.57	14.22	0.15	16.10	19.88	-1.47	25.47
	Dias: 8.28	0.11	10.39	7.98	0.83	10.30	9.04	0.01	10.41	10.82	-0.68	13.65
Feat. with Decision Tree	Sys: 23.87	2.49	29.84	22.98	0.40	28.82	14.62	-1.04	16.96	19.44	0.53	24.80
	Dias: 11.45	1.42	15.37	11.01	0.34	14.37	10.29	1.39	11.47	11.83	-0.25	15.02
Feat. with Random Forest	Sys: 18.18	2.46	22.25	16.28	0.35	20.30	13.16	-0.93	15.83	14.20	0.23	18.39
	Dias: 8.42	1.32	10.70	7.99	0.19	10.14	8.35	0.42	9.28	8.81	-0.06	11.06
Feat. with SVR	Sys: 17.21	-0.32	21.22	16.00	-0.45	20.05	14.20	-0.20	16.72	13.84	-0.75	18.28
	Dias: 7.58	-1.56	10.12	7.82	-1.59	10.16	8.36	0.47	8.71	8.95	-1.16	11.17
CNN	Sys: 17.19	0.73	7.27	16.33	0.09	18.72	12.18	0.81	9.49	16.17	0.66	19.31
	Dias: 7.97	0.89	3.72	8.00	0.05	9.16	8.29	0.39	5.90	8.85	-0.02	10.57
CNN-LSTM	Sys: 18.69	2.64	9.35	18.42	0.42	21.49	14.63	-0.37	9.57	14.56	-0.44	18.20
	Dias: 7.99	-0.03	4.28	8.22	-0.21	8.22	9.37	0.59	5.48	8.45	-0.29	10.30
TCN	Sys: 17.28	0.79	8.41	17.39	-1.09	20.44	13.94	-3.85	10.61	23.59	-17.13	23.39
	Dias: 8.01	0.59	4.34	8.31	-0.58	9.77	10.04	-2.77	6.61	14.35	-11.65	13.19
Transformer	Sys: 17.65	1.55	8.15	16.78	1.45	19.18	13.31	0.14	9.62	15.15	1.86	18.26
	Dias: 8.06	1.03	4.11	8.16	0.40	9.35	8.93	0.42	5.61	8.83	0.31	10.67
PCTN	Sys: 18.78	0.85	10.71	16.87	-0.89	19.19	14.28	2.41	9.72	15.68	-1.53	19.29
	Dias: 8.33	-0.02	4.95	8.19	0.09	9.34	8.57	1.64	5.59	9.11	-1.43	11.14
ResNet	Sys: 19.82	3.76	10.93	18.01	-0.43	21.00	15.05	-0.56	9.91	19.88	-12.57	21.38
	Dias: 9.07	2.12	5.43	8.44	0.06	10.01	9.65	0.04	6.02	12.00	-8.69	11.89
UNet	Sys: 17.48	2.05	7.91	16.17	-0.57	18.39	12.49	0.36	8.92	15.79	0.25	19.29
	Dias: 7.89	0.72	4.08	7.93	-0.31	9.09	8.60	0.39	5.08	8.89	-0.09	10.75
ResUNet + Attention	Sys: 18.86	3.25	10.08	16.44	-0.87	18.95	14.09	0.45	8.92	15.28	-0.96	19.08
	Dias: 9.01	1.41	5.12	8.08	0.02	9.53	8.57	0.65	5.67	9.03	-1.12	11.16
Moment	Sys: 19.03	2.93	11.42	17.23	-0.19	20.18	13.19	-0.05	9.48	14.42	0.21	17.65
	Dias: 8.92	1.18	5.73	8.26	-0.13	9.76	7.77	0.16	5.48	8.84	-0.59	10.80

Table 5.3: Model performances on the UCI, Sensors, BCG, and PPGBP datasets

From the results, it can be observed that the best performing model depends on the dataset and that there is not a model that consistently outperforms the others across all datasets.

For the UCI dataset, the best model was CNN for SBP prediction, with the lowest MAE of 17.19 mmHg, and SVR for DBP prediction, with a MAE of 7.58 mmHg.

For the Sensors dataset, the best performing model was SVR, with the lowest MAE of 16.00 and 7.82 mmHg respectively per SBP and DBP. This suggests that traditional machine learning methods may still have an advantage, especially when data volume is limited.

For the BCG dataset, CNN provided the best performance for SBP, with a MAE of 12.18 mmHg, while the Transformer model performed best for DBP, with a MAE of 8.29 mmHg.

In the PPGBP dataset, SVR achieved the lowest MAE for SBP, with MAE 13.84 mmHg, while CNN-LSTM performed best for DBP, with MAE 8.45 mmHg.

The significantly lower errors obtained from BCG and PPGBP datasets could be due to the characteristics of the dataset such as the BP distribution; moreover, the quality of the signal

could be higher, since the acquisitions of the original datasets were not carried out in an ICU environment, contrary to the Sensors and UCI datasets.

Overall, it can be observed that no single model surpasses all the others, so data characteristics have an impact on model performances.

Moreover, traditional ML models, like SVR, remain competitive, particularly in the case of small datasets, where DL models risk overfitting and low generalisability.

Additionally, in some cases SBP and DBP values are better predicted from different models, this could mean that the underlying patterns in the signal which are influenced by SBP and DBP are not entirely the same, therefore the two values require different modelling approaches to obtain a better performance.

Similarly to what was observed with VitalDB, also for the benchmark datasets the FM Moment did not provide significant accuracy in terms of predictions, therefore simpler approaches like traditional ML methods yield better results, particularly in smaller datasets.

5.2 Comparison with State-of-the-Art

As mentioned in Section 3.3, comparing the results from different authors regarding BP estimation from PPG signals can be difficult.

Firstly, different studies use different starting datasets; this issue was addressed in this work by using benchmark datasets, which made the comparison easier.

Furthermore, several authors use different metrics, so in this case, the most commonly used metrics in BP regression problems, namely MAE, ME, and STD, were computed in order to compare the different methods.

Comparing the results presented in this chapter to those presented in Section 3.3, it can be noted that some models achieve the SoA results, when considering the calibration free approach.

In particular, considering the VitalDB dataset, the results obtained with the best performing model (CNN-LSTM) are comparable to the results presented by the study that inspired the use of this network, [56]. Shaikh *et al.* use a calibration based approach, by using the PulseDB Calibration Based testing subset as the testing set, while the other subsets (PulseDB Training, Calibration Free and AAMI Testing sets) were merged and used as the training set.

In this work it was decided to use the VitalDB sets as provided by their Authors, therefore the comparison can be made considering the performances on VitalDB CalBased testing set with the VitalDB Training set as a training set, to maintain the calibration based approach. In this thesis, it was achieved 5.33 and 3.42 mmHg MAE for SBP and DBP respectively, while Shaikh *et al.* obtained 5.16 and 3.24 mmHg MAE, in the case of the model that includes the ECG.

Moreover, the model implemented in this work outperforms its original counterpart when using only the PPG signal, in fact 8.88 and 5.89 mmHg for SBP and DBP were obtained compared to

5 Results

10.74 and 6.87 mmHg. The differences in the results could be due to the different testing and training sets that were used, even if PulseDB contains the VitalDB data that was used in this work.

Considering the benchmark datasets, the achieved results were close to the SoA set by the study that provided them [52], achieving slightly higher errors.

Considering the UCI dataset, for SBP prediction, the CNN model achieved an MAE of 17.19 mmHg, which is slightly higher than the 16.59 mmHg reported for ResNet in González *et al.* For DBP prediction, SVR performed better than ResNet, achieving a lower MAE (7.58 mmHg vs. 8.30 mmHg).

Regarding the Sensors dataset, SVR was the best-performing model, achieving an MAE of 16.00 mmHg for SBP and 7.82 mmHg for DBP. These values are close to the 15.60 mmHg and 7.50 mmHg reported by González *et al.* with the same method, but slightly worse. This suggests that traditional ML models still hold an advantage in this dataset and González *et al.* probably achieved a better result due to the different signal features that they used and possibly the fine-tuning of SVR parameters.

For the BCG dataset, for SBP prediction, CNN achieved an MAE of 12.18 mmHg, which is slightly worse than the 11.42 mmHg achieved by AdaBoost in González *et al.* However, CNN had a much lower standard deviation, indicating more consistent predictions.

For DBP prediction, the Transformer model performed similarly to AdaBoost (8.29 mmHg vs. 8.06 mmHg), but again with a much lower standard deviation.

Finally, for PPGBP dataset, in the case of SBP prediction SVR achieved an MAE of 13.84 mmHg, which is slightly worse than the 13.06 mmHg obtained with LightGBM in González *et al.*; similarly, for DBP prediction, CNN-LSTM had a slightly higher MAE than LightGBM (8.45 mmHg vs. 8.16 mmHg).

Despite these results, which reach other studies' levels of accuracy in predicting SBP and DBP, if compared to other SoA performances, which present much lower estimation errors, the models described in this work appear to perform worse. However, this discrepancy is largely due to differences in the datasets used and on the validation approach that was employed.

As a matter of fact, many studies either use proprietary datasets or a selection of data extracted from publicly available datasets, though they do not disclose the data selection procedure that was followed, limiting the reproducibility of their results.

Moreover, many studies employ a calibration-based approach, where data acquired from the same subjects is included in both the training and testing sets. This is typically done by randomly splitting the initial dataset in training, validation and testing subsets without ensuring that subjects do not overlap across these partitions. As a result, these studies often do not validate their models on entirely unseen subjects, making it difficult to assess the true generalisation

5 Results

performance of different models.

To address this limitation, the model evaluation presented above is based on reproducible benchmark datasets, ensuring a fair and standardised comparison of various BP estimation techniques.

Considering the evaluation standards described in Subsection 3.3.4, it is clear that solely the proposed models are still not adequate to be used in medical BP measuring devices, since they do not achieve sufficiently accurate predictions for clinical and diagnostic applications.

However, this could be solved in the future by improving the model architectures, incorporating more diverse and representative datasets, and refining feature extraction techniques. Future work should also focus on exploring personalised calibration techniques to reduce estimation errors and improve the clinical applicability of the proposed methods.

Additionally, a key direction for future research is to investigate alternative multimodal approaches, which could enhance both the reliability and accuracy of BP estimation, making it more suitable for real-world healthcare settings.

Conclusion and Future Work

This thesis explored various ML and DL approaches for non-invasive BP estimation using physiological signals, particularly PPG. The research focused on evaluating different models, including traditional ML algorithms such as Decision Trees, Random Forest and SVR, and advanced DL models, such as CNN based architectures, LSTM and Transformer based models. The implemented models were tested on publicly available benchmark datasets to ensure a fair comparison and to validate the effectiveness of the proposed methodologies.

The results indicate that no single model consistently outperforms all others across different datasets. Traditional ML models such as SVR demonstrated strong performance, particularly for smaller datasets, where deep learning models tend to overfit.

The comparison with SoA methods revealed that the proposed approaches achieved competitive results, particularly when considering calibration free BP estimation. While some models reached SoA accuracy, variations in dataset characteristics and BP distribution affected the overall performance. This highlights the complexity of BP estimation and the need for model selection based on specific use cases and data properties.

While the results obtained are promising, a purely algorithmic solution based solely on PPG or on the simple combination of PPG and ECG does not appear to achieve a level of reliability sufficient for clinical or diagnostic use. This suggests that a key direction for future research is the exploration of alternative multimodal configurations that can enhance both the accuracy and robustness of BP estimation.

Building from the presented results, several areas remain open for improvement and future research:

- Implementation of a wearable system that integrates DL models with PAT and PTT approaches, leveraging the strengths of both techniques for BP estimation;

6 Conclusion and Future Work

- Development of subject-specific calibration approaches to improve the prediction based on individual physiological characteristics;
- Expansion of datasets by incorporating real-world wearable device data to test the robustness and reliability in practical applications;
- Deployment of models for real-time BP monitoring on edge devices, ensuring efficient and low-power operation for continuous health tracking.

By addressing these challenges, future research can further improve the accuracy, reliability, and real-world applicability of ML and DL models for non-invasive BP estimation, contributing to better cardiovascular health monitoring and management.

Bibliography

- [1] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida, “Wearable Photoplethysmographic Sensors—Past and Present,” *Electronics*, vol. 3, no. 2, pp. 282–302, 2014. [Online]. Available: <https://www.mdpi.com/2079-9292/3/2/282>
- [2] P. A. Kyriacou and S. Chatterjee, “2 - The origin of photoplethysmography,” in *Photoplethysmography*, J. Allen and P. Kyriacou, Eds. Academic Press, 2022, pp. 17–43. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128233740000049>
- [3] E. Mejía-Mejía, J. Allen, K. Budidha, C. El-Hajj, P. A. Kyriacou, and P. H. Charlton, “4 - Photoplethysmography signal processing and synthesis,” in *Photoplethysmography*, J. Allen and P. Kyriacou, Eds. Academic Press, 2022, pp. 69–146. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128233740000153>
- [4] R. Pal, A. Rudas, S. Kim, J. N. Chiang, A. Barney, and M. Cannesson, “An algorithm to detect dicrotic notch in arterial blood pressure and photoplethysmography waveforms using the iterative envelope mean method,” *Computer Methods and Programs in Biomedicine*, vol. 254, p. 108283, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260724002785>
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [6] C. El-Hajj and P. A. Kyriacou, “A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure,” *Biomedical Signal Processing and Control*, vol. 58, p. 101870, Apr 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809420300264>

Bibliography

- [7] R. Mukkamala, J.-O. Hahn, O. T. Inan, L. K. Mestha, C.-S. Kim, H. Toreyin, and S. Kyal, "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice," *IEEE Transactions on Biomedical Engineering*, vol. 62, pp. 1879–1901, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:31914690>
- [8] Y. Kurylyak, F. Lamonaca, and D. Grimaldi, "A Neural Network-based method for continuous blood pressure estimation from a PPG signal," in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013, pp. 280–283.
- [9] M. Shaikh and M. Forouzanfar, "Dual-Stream CNN-LSTM Architecture for Cuffless Blood Pressure Estimation from PPG and ECG Signals: A PulseDB Study," *IEEE Sensors Journal*, vol. PP, pp. 1–1, 01 2024.
- [10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
- [11] Z. Tian, A. Liu, G. Zhu, and X. Chen, "A paralleled CNN and Transformer network for PPG-based cuff-less blood pressure estimation," *Biomedical Signal Processing and Control*, vol. 99, p. 106741, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809424007997>
- [12] Z. Jamil, L. Ting Lui, and R. H. M. Chan, "Blood Pressure Estimation Using Self-Attention Mechanism Built-In ResUNet on PulseDB: Demographic Fairness and Generalization," *IEEE Sensors Journal*, vol. 25, no. 1, pp. 1694–1705, 2025.
- [13] National Heart Lung and Blood Institute, "What is high blood pressure?" <https://www.nhlbi.nih.gov/health/high-blood-pressure#:~:text=A%20healthy%20systolic%20blood%20pressure,80%20mm%20Hg%20or%20higher.>, [Accessed 7-03-2025].
- [14] "Hypertension," <https://www.who.int/news-room/fact-sheets/detail/hypertension>, accessed: 2025-2-20.
- [15] J. E. Sharman, I. Tan, G. S. Stergiou, C. Lombardi, F. Saladini, M. Butlin, R. Padwal, K. Asayama, A. Avolio, T. M. Brady, A. Murray, and G. Parati, "Automated 'oscillometric' blood pressure measuring devices: how they work and what they measure," *Journal of Human Hypertension*, vol. 37, no. 2, pp. 93–100, Feb 2023. [Online]. Available: <https://doi.org/10.1038/s41371-022-00693-x>
- [16] C. Landry and R. Mukkamala, "Current evidence suggests that estimating blood pressure from convenient ECG waveforms alone is not viable," *Journal of Electrocardiology*,

Bibliography

- vol. 81, pp. 153–155, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002207362300211X>
- [17] S. A. Alowais, S. S. Alghamdi, N. Alsuhebany, T. Alqahtani, A. I. Alshaya, S. N. Almohareb, A. Aldairem, M. Alrashed, K. Bin Saleh, H. A. Badreldin, M. S. Al Yami, S. Al Harbi, and A. M. Albekairy, “Revolutionizing healthcare: the role of artificial intelligence in clinical practice,” *BMC Medical Education*, vol. 23, no. 1, p. 689, Sep 2023. [Online]. Available: <https://doi.org/10.1186/s12909-023-04698-z>
- [18] V. Nasteski, “An overview of the supervised machine learning methods,” *Horizons. b*, vol. 4, no. 51-62, p. 56, 2017.
- [19] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>
- [20] I. Guyon and A. Elisseeff, *“An Introduction to Feature Extraction”*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–25. [Online]. Available: https://doi.org/10.1007/978-3-540-35488-8_1
- [21] S. Krishnan and Y. Athavale, “Trends in biomedical signal feature extraction,” *Biomedical Signal Processing and Control*, vol. 43, pp. 41–63, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809418300399>
- [22] F. Murtagh, “Multilayer perceptrons for classification and regression,” *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0925231291900235>
- [23] M. Awad and R. Khanna, *“Support Vector Regression”*. Berkeley, CA: Apress, 2015, pp. 67–80. [Online]. Available: https://doi.org/10.1007/978-1-4302-5990-9_4
- [24] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artificial Intelligence Review*, vol. 39, no. 4, pp. 261–283, Apr 2013. [Online]. Available: <https://doi.org/10.1007/s10462-011-9272-4>
- [25] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [26] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999–7019, 2022.

Bibliography

- [27] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, “A Review of Activation Function for Artificial Neural Network,” in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)*, 2020, pp. 281–286.
- [28] H. Gholamalinezhad and H. Khosravi, “Pooling Methods in Deep Neural Networks, a Review,” 2020. [Online]. Available: <https://arxiv.org/abs/2009.07485>
- [29] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, “Impact of fully connected layers on performance of convolutional neural networks for image classification,” *Neurocomputing*, vol. 378, pp. 112–119, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231219313803>
- [30] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal Convolutional Networks for Action Segmentation and Detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003–1012.
- [31] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 07 2019. [Online]. Available: https://doi.org/10.1162/neco_a_01199
- [32] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167278919305974>
- [33] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun, “A comprehensive survey on pretrained foundation models: a history from BERT to ChatGPT,” *International Journal of Machine Learning and Cybernetics*, Nov 2024. [Online]. Available: <https://doi.org/10.1007/s13042-024-02443-6>
- [34] J. Schneider, C. Meske, and P. Kuss, “Foundation Models,” *Business & Information Systems Engineering*, vol. 66, no. 2, pp. 221–231, Apr 2024. [Online]. Available: <https://doi.org/10.1007/s12599-024-00851-0>
- [35] J. Y. Wagner, I. Negulescu, M. Schöfthaler, A. Hapfelmeier, A. S. Meidert, W. Huber, R. M. Schmid, and B. Saugel, “Continuous noninvasive arterial pressure measurement using the volume clamp method: an evaluation of the CNAP device in intensive care unit patients,” *Journal of Clinical Monitoring and Computing*, vol. 29, no. 6, pp. 807–813, Dec 2015. [Online]. Available: <https://doi.org/10.1007/s10877-015-9670-2>

Bibliography

- [36] A. B. Amado Rey, “Estimation of Blood Pressure by Image-Free, Wearable Ultrasound,” *Artery Research*, vol. 30, no. 1, p. 3, May 2024. [Online]. Available: <https://doi.org/10.1007/s44200-024-00048-6>
- [37] G. M. Drzewiecki, J. Melbin, and A. Noordergraaf, “Arterial tonometry: Review and analysis,” *Journal of Biomechanics*, vol. 16, no. 2, pp. 141–152, 1983. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0021929083900374>
- [38] T.-W. Wang, W.-X. Chen, H.-W. Chu, and S.-F. Lin, “Single-Channel Bioimpedance Measurement for Wearable Continuous Blood Pressure Monitoring,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.
- [39] T. Le, F. Ellington, T.-Y. Lee, K. Vo, M. Khine, S. K. Krishnan, N. Dutt, and H. Cao, “Continuous Non-Invasive Blood Pressure Monitoring: A Methodological Review on Measurement Techniques,” *IEEE Access*, vol. 8, pp. 212 478–212 498, 2020.
- [40] G. Wang, M. Atef, and Y. Lian, “Towards a Continuous Non-Invasive Cuffless Blood Pressure Monitoring System Using PPG: Systems and Circuits Review,” *IEEE Circuits and Systems Magazine*, vol. 18, no. 3, pp. 6–26, 2018.
- [41] F. Riaz, M. A. Azad, J. Arshad, M. Imran, A. Hassan, and S. Rehman, “Pervasive blood pressure monitoring using Photoplethysmogram (PPG) sensor,” *Future Generation Computer Systems*, vol. 98, pp. 120–130, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167739X18327729>
- [42] E. Martínez-Ríos, L. Montesinos, M. Alfaro-Ponce, and L. Pecchia, “A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data, journal = Biomedical Signal Processing and Control,” vol. 68, p. 102813, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1746809421004109>
- [43] E. Finnegan, S. Davidson, M. Harford, J. Jorge, P. Watkinson, D. Young, L. Tarassenko, and M. Villarroel, “Pulse arrival time as a surrogate of blood pressure,” *Scientific Reports*, vol. 11, no. 1, p. 22767, Nov 2021. [Online]. Available: <https://doi.org/10.1038/s41598-021-01358-4>
- [44] W. Chen, T. Kobayashi, S. Ichikawa, Y. Takeuchi, and T. Togawa, “Continuous estimation of systolic blood pressure using the pulse arrival time and intermittent calibration,” *Medical and Biological Engineering and Computing*, vol. 38, no. 5, pp. 569–574, Sep 2000. [Online]. Available: <https://doi.org/10.1007/BF02345755>
- [45] R. C. Block, M. Yavarimanesh, K. Natarajan, A. Carek, A. Mousavi, A. Chandrasekhar, C.-S. Kim, J. Zhu, G. Schifitto, L. K. Mestha, O. T. Inan, J.-O. Hahn, and

Bibliography

- R. Mukkamala, "Conventional pulse transit times as markers of blood pressure changes in humans," *Scientific Reports*, vol. 10, no. 1, p. 16373, Oct 2020. [Online]. Available: <https://doi.org/10.1038/s41598-020-73143-8>
- [46] E. Monte-Moreno, "Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques," *Artificial Intelligence in Medicine*, vol. 53, no. 2, pp. 127–138, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S093336571100056X>
- [47] A. Tazarv and M. 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)*, 2021, pp. 5658–5662.
- [48] M. Elgendi, R. Fletcher, Y. Liang, N. Howard, N. H. Lovell, D. Abbott, K. Lim, and R. Ward, "The use of photoplethysmography for assessing hypertension," *npj Digital Medicine*, vol. 2, no. 1, p. 60, Jun 2019. [Online]. Available: <https://doi.org/10.1038/s41746-019-0136-7>
- [49] C. Poon and Y. Zhang, "Cuff-less and Noninvasive Measurements of Arterial Blood Pressure by Pulse Transit Time," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 5877–5880.
- [50] P. Fung, G. Dumont, C. Ries, C. Mott, and M. Ansermino, "Continuous noninvasive blood pressure measurement by pulse transit time," in *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, 2004, pp. 738–741.
- [51] H. Gesche, D. Grosskurth, G. K uchler, and A. Patzak, "Continuous blood pressure measurement by using the pulse transit time: comparison to a cuff-based method," *European Journal of Applied Physiology*, vol. 112, no. 1, pp. 309–315, Jan 2012. [Online]. Available: <https://doi.org/10.1007/s00421-011-1983-3>
- [52] S. Gonz alez, W.-T. Hsieh, and T. P.-C. Chen, "A benchmark for machine-learning based non-invasive blood pressure estimation using photoplethysmogram," *Scientific Data*, vol. 10, no. 1, p. 149, Mar 2023. [Online]. Available: <https://doi.org/10.1038/s41597-023-02020-6>
- [53] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 4, pp. 859–869, 2017.
- [54] S. Yang, J. Sohn, S. Lee, J. Lee, and H. C. Kim, "estimation and validation of arterial blood pressure using photoplethysmogram morphology features in conjunction with pulse

Bibliography

- arrival time in large open databases"; *IEEE J Biomed Health Inform*, vol. 25, no. 4, pp. 1018–1030, Apr. 2021.
- [55] S. Maqsood, S. Xu, M. Springer, and R. Mohawesh, "A Benchmark Study of Machine Learning for Analysis of Signal Feature Extraction Techniques for Blood Pressure Estimation Using Photoplethysmography (PPG)," *IEEE Access*, vol. 9, pp. 138 817–138 833, 2021.
- [56] M. R. Shaikh and M. Forouzanfar, "Dual-Stream CNN-LSTM Architecture for Cuffless Blood Pressure Estimation From PPG and ECG Signals: A PulseDB Study," *IEEE Sensors Journal*, vol. 25, no. 2, pp. 4006–4014, 2025.
- [57] B. Kamanditya, Y. N. Fuadah, N. Q. Mahardika T., and K. M. Lim, "Continuous blood pressure prediction system using Conv-LSTM network on hybrid latent features of photoplethysmogram (PPG) and electrocardiogram (ECG) signals," *Scientific Reports*, vol. 14, no. 1, p. 16450, Jul 2024. [Online]. Available: <https://doi.org/10.1038/s41598-024-66514-y>
- [58] K. Gupta, N. Jiwani, and N. Afreen, "Blood Pressure Detection Using CNN-LSTM Model," in *2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT)*, 2022, pp. 262–366.
- [59] A. Paviglianiti, V. Randazzo, S. Villata, G. Cirrincione, and E. Pasero, "A Comparison of Deep Learning Techniques for Arterial Blood Pressure Prediction," *Cognitive Computation*, vol. 14, no. 5, pp. 1689–1710, Sep 2022. [Online]. Available: <https://doi.org/10.1007/s12559-021-09910-0>
- [60] J. Cheng, Y. Xu, R. Song, Y. Liu, C. Li, and X. Chen, "Prediction of arterial blood pressure waveforms from photoplethysmogram signals via fully convolutional neural networks," *Computers in Biology and Medicine*, vol. 138, p. 104877, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0010482521006715>
- [61] T. B. D. S. Costa, F. M. Dias, D. A. C. Cardenas, M. A. F. D. Toledo, D. M. D. Lima, J. E. Krieger, and M. A. Gutierrez, "Blood Pressure Estimation From Photoplethysmography by Considering Intra- and Inter-Subject Variabilities: Guidelines for a Fair Assessment," *IEEE Access*, vol. 11, pp. 57 934–57 950, 2023.
- [62] "IEEE Standard for Wearable Cuffless Blood Pressure Measuring Devices," *IEEE Std 1708-2014*, pp. 1–38, 2014.
- [63] E. O'Brien, B. Waeber, G. Parati, J. Staessen, and M. G. Myers, "Blood pressure measuring devices: recommendations of the European Society of Hypertension," *BMJ*,

Bibliography

- vol. 322, no. 7285, pp. 531–536, 2001. [Online]. Available: <https://www.bmj.com/content/322/7285/531.1>
- [64] E. O’Brien, J. Petrie, W. Littler, M. de Swiet, P. L. Padfield, K. O’Malley, M. Jamieson, D. Altman, M. Bland, and N. Atkins, ““the british hypertension society protocol for the evaluation of automated and semi-automated blood pressure measuring devices with special reference to ambulatory systems”,” *J. Hypertens.*, vol. 8, no. 7, pp. 607–619, Jul. 1990.
- [65] G. S. Stergiou, B. Alpert, S. Mieke, R. Asmar, N. Atkins, S. Eckert, G. Frick, B. Friedman, T. Graßl, T. Ichikawa, J. P. Ioannidis, P. Lacy, R. McManus, A. Murray, M. Myers, P. Palatini, G. Parati, D. Quinn, J. Sarkis, A. Shennan, T. Usuda, J. Wang, C. O. Wu, and E. O’Brien, “A universal standard for the validation of blood pressure measuring devices: Association for the Advancement of Medical Instrumentation/European Society of Hypertension/International Organization for Standardization (AAMI/ESH/ISO) Collaboration Statement, journal=Journal of Hypertension,” vol. 36, no. 3, 2018. [Online]. Available: https://journals.lww.com/jhypertension/fulltext/2018/03000/a_universal_standard_for_the_validation_of_blood.4.aspx
- [66] A. E. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, no. 1, p. 160035, May 2016. [Online]. Available: <https://doi.org/10.1038/sdata.2016.35>
- [67] B. Moody, G. Moody, M. Villarreal, G. D. Clifford, and I. Silva, “MIMIC-III Waveform Database (version 1.0),” 2020. [Online]. Available: <https://doi.org/10.13026/c2607m>
- [68] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000, [Online].
- [69] H.-C. Lee, Y. Park, S. B. Yoon, S. M. Yang, D. Park, and C.-W. Jung, “VitalDB, a high-fidelity multi-parameter vital signs database in surgical patients,” *Scientific Data*, vol. 9, no. 1, p. 279, Jun 2022. [Online]. Available: <https://doi.org/10.1038/s41597-022-01411-5>
- [70] W. Wang, P. Mohseni, K. L. Kilgore, and L. Najafizadeh, “PulseDB: A large, cleaned dataset based on MIMIC-III and VitalDB for benchmarking cuff-less blood pressure estimation methods,” *Frontiers in Digital Health*, vol. 4, 2023. [Online]. Available: <https://www.frontiersin.org/journals/digital-health/articles/10.3389/fdgth.2022.1090854>
- [71] N. Aguirre, E. Grall-Maës, L. J. Cymberknop, and R. L. Armentano, “Blood Pressure Morphology Assessment from Photoplethysmogram and Demographic Information Using

Bibliography

- Deep Learning with Attention Mechanism,” *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/6/2167>
- [72] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, “Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time,” in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1006–1009.
- [73] C. Carlson, V.-R. Turpin, A. Suliman, C. Ade, S. Warren, and D. E. Thompson, “Bed-Based Ballistocardiography Dataset,” 2020. [Online]. Available: <https://dx.doi.org/10.21227/77hc-py84>
- [74] Y. Liang, Z. Chen, G. Liu, and M. Elgendi, “A new, short-recorded photoplethysmogram dataset for blood pressure monitoring in China,” *Scientific Data*, vol. 5, p. 180020, 2018.
- [75] M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti, and L. Benini, “Robust Real-Time Embedded EMG Recognition Framework Using Temporal Convolutional Networks on a Multicore IoT Processor,” *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 2, pp. 244–256, 2020.
- [76] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [77] Y. J. (cainmagi), “Overview - MDNC Documentation — cainmagi.github.io,” <https://cainmagi.github.io/MDNC/>, [Accessed 05-03-2025].
- [78] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, “MOMENT: A Family of Open Time-series Foundation Models,” in *International Conference on Machine Learning*, 2024.
- [79] HPC Cineca, “Leonardo,” <https://www.hpc.cineca.it/systems/hardware/leonardo/>, [Accessed 10-03-2025].
- [80] CINECA Technical Portal, “Leonardo user guide,” <https://wiki.u-gov.it/confluence/display/SCAIUS/LEONARDO+User+Guide>, [Accessed 10-03-2025].