# Blood Pressure Estimation through Photoplethysmography using Deep Learning in Clinical Setting: Critical Survey and Solutions

**François LaBerge**

**A Thesis**

**in**

**The Department**

**of**

**Computer Science and Software Engineering**

**Presented in Partial Fulfillment of the Requirements**

**for the Degree of**

**Master of Computer Science at**

**Concordia University**

**Montréal, Québec, Canada**

**May 2024**

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **François LaBerge**

Entitled: **Blood Pressure Estimation through Photoplethysmography using Deep Learning in Clinical Setting: Critical Survey and Solutions**

and submitted in partial fulfillment of the requirements for the degree of

**Master of Computer Science**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
*Dr. Juergen Rilling*

_____ External Examiner
*Dr. Juergen Rilling*

_____ Examiner
*Dr. Abdelhak Bentaleb*

_____ Supervisor
*Dr. Jamal Bentahar*

_____ Co-supervisor
*Dr. Louis-Philippe Fortier*

Approved by _____
Dr. Joey Paquet, Chair
Department of Computer Science and Software Engineering

_____ 2024

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

# Abstract

Blood Pressure Estimation through Photoplethysmography using Deep Learning in Clinical Setting: Critical Survey and Solutions

François LaBerge

Current solutions for blood pressure monitoring can be classified as invasive or non-invasive, both with drawbacks. Invasive blood pressure monitoring can lead to complications. Non-invasive blood pressure monitoring is intermittent which leads to missed episodes of hypertension and hypotension, also leading to complications. The state of the art for blood pressure monitoring through machine learning methods usually requires personalization, which is prohibitive in a clinical application. These proposed methods are generally not evaluated for clinical application. Datasets are usually split randomly, while a patient-wise split is required.

We first start by performing a survey of the literature to find candidate models for evaluation. These models are reproduced for evaluation alongside our proposed models. Popular input modalities from the literature are also reproduced with our proposed input modality. All combinations of models and input modalities are then evaluated against a patient-wise and random split. We perform a learning curve analysis to estimate how much data would be required to pass the AAMI standard.

The performance results establish that no model can provide calibration-free, non-invasive blood pressure monitoring using a single PPG site. The performance metrics show that our models and input modalities outperform the state of the art for random and patient-wise splits. Comparison against the models demonstrates that model complexity is insufficient to achieve better performance and that better preprocessing is a more efficient way to improve performance. The learning-curve analysis estimates that additional data could help achieve a model that passes the AAMI standard.

# Acknowledgments

I would like to express my deepest appreciation to my supervisor Dr. Jamal Bentahar. His guidance, support, and expertise were critical to my academic development and accomplishment of this thesis. I am deeply indebted to my co-supervisors Dr. Lawrence Leroux and Dr. Louis-Philippe Fortier. Their mentorship, technical insights, and interest in my research have been instrumental in achieving our goals. Their work ethic inspires me to strive for excellence.

I am deeply indebted to Ahmed Alagha for sharing his knowledge and invaluable advice in designing experiments and training deep learning models. I present my warmest thanks to Amélie Hébert for her unending support and help with spreadsheet formatting. Special thanks to Richard LaBerge and Etienne Fortier from Impack CPR for their professional support. Their counsel was essential to the expansion of my professional development.

I could not have undertaken this journey without my family and parents. Their continuous support was pivotal in starting and finishing my endeavors.

# Contents

# List of Figures

# List of Tables

# Acronyms

**AAMI** Association for the Advancement of Medical Instrumentation. 18, 20, 22, 47, 48

**ABP** arterial blood pressure. 1, 4, 7, 27, 28, 29, 30, 31, 32, 33

**BHS** British Hypertension Society. 18

**BP** blood pressure. 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 17, 18, 19, 20, 22, 26, 30, 31, 32, 33, 35, 36, 39, 45, 48, 49, 51, 52, 53

**DBP** disatolic blood pressure. 5, 8, 9, 12, 13, 18, 22, 24, 26, 30, 31, 32, 33, 41, 46, 47, 50

**ICU** intensive care unit. 27

**LED** light emitting diode. 7

**MAE** mean absolute error. 2, 8, 9, 12, 13, 18, 22, 24, 26, 41, 46, 47, 50

**ME** mean error. 2, 18

**ML** machine learning. 11, 17

**PAT** pulse arrival time. 9, 19, 20, 53

**PIR** PPG Intesity Ratio. 9

**PPG** photoplethysmography. 7, 8, 9, 10, 12, 14, 17, 18, 19, 20, 21, 22, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 48, 53

**PTT** pulse transit time. 9, 19, 20, 53

**PWA** pulse wave analysis. 8, 10, 19, 20

**RMSE** root mean squared error. 2, 18

**SBP** systolic blood pressure. 5, 6, 8, 9, 12, 13, 18, 22, 24, 26, 30, 31, 32, 33, 41, 46, 47, 50

**SD** standard deviation. 22, 41

**SQI** signal quality index. 29, 31, 32, 33

# Chapter 1

# Introduction

Blood pressure is one of the most common health metrics used to evaluate the health of a patient's hemodynamics. In itself, it measures the pressure exerted by the blood on the arterial walls. blood pressure (BP) monitoring can help characterize the circulatory system of a patient and help practitioners take adequate actions to maximize the chances of favorable outcomes. For instance, complications stemming from episodes of hypotension and hypertension can be counteracted if acted upon in a timely fashion.

Monitoring of arterial blood pressure (ABP) is critical during the perioperative period. It has been shown to be a good predictor of outcomes for patients. Episodes of hypotension during perioperative, intraoperative, and postoperative care have been associated with increased chances of myocardial infarction and death [58, 59].

## 1.1 Motivation

The *status quo* of BP monitoring solutions can be divided into two categories: non-invasive and invasive methods, both of which come with their drawbacks. Invasive BP monitoring can cause various complications with the patients. Complications include infection and sepsis [55]. Meanwhile, non-invasive BP is safer and does not cause complications. However, most implementations of non-invasive BP monitoring are non-continuous and cause intermittent readings. It has been shown that this type of monitoring can cause critical episodes of change in BP, such as hypotension, to

be outright missed. In these cases, proper action can not be taken, which reduces the chances of a favorable outcome for patients.

The state of the art has proposed some non-invasive continuous BP monitoring algorithms. Unfortunately, these algorithms usually require personalization to achieve an acceptable performance. Personalization is not applicable in a clinical or EMS setting, since labeled data for the patient is not available. As such, a third property of the algorithm is required for clinical use: calibration-free.

As such we are looking to develop a non-invasive, continuous, and calibration-free BP monitoring algorithm for use in a clinical setting. This algorithm could provide considerable improvement in outcomes for patients undergoing perioperative care.

The literature on this topic has proposed a multitude of methods for non-invasive, continuous BP monitoring. These solutions most commonly rely on machine learning and deep learning methods. Unfortunately, proposed methods are often difficult or impossible to properly compare due to several factors. The data used by the authors to develop their models is not always reproducible. In some cases, the dataset is not shared, making reproduction impossible. In other cases, the preprocessing of the dataset itself is not reproducible. Similarly, the implementation of the deep learning models themselves can not be reproduced. Code implementation for the proposed methods is seldom published and descriptions of the architecture are insufficient to properly reproduce it. Lastly, the performance of the methods is not always reported with the same metrics. In our review, works could report performance in terms of mean absolute error (MAE), mean error (ME), root mean squared error (RMSE), R-square, or a combination of these.

## 1.2 Contributions

In order to tackle the gaps identified in the literature and advance the development of a non-invasive, continuous, and calibration-free BP monitoring algorithm, we provide the following contributions:

- We first provide a comprehensive review of the literature through a survey.
- We reproduce and evaluate models from the literature from a concrete clinical perspective and publish our implementations on an open-source platform.

- We propose new deep learning architectures for our BP monitoring algorithm. Those architectures are evaluated against the reproduced models from the literature considering a concrete clinical context. Implementation for those models along with our automatic preprocessing pipelines and preprocessing methodology for PPG signals are also made open-source.
- We provide an analysis of the MIMIC-IV dataset to evaluate how much data is required to achieve a calibration-free BP estimation algorithm.

## 1.3 Organization

The organization of the thesis is as follows. In Chapter 2, we provide the background knowledge for BP and BP monitoring. We also present a literature review of BP estimation through machine learning. In Chapter 3, we present our survey, model reproductions, proposed models, preprocessing methodologies, and learning curve analysis. Lastly, in Chapter 4, we summarize our findings and suggest avenues for future research.

# Chapter 2

# Background and Literature Review

## 2.1 Background

Hemodynamics is the study of blood flow in the circulatory system. A major component of hemodynamics is BP. In the following subsection, we cover the background related to BP, by detailing its basic mechanisms and monitoring options.

### 2.1.1 Blood Pressure

To understand the importance of BP monitoring, we must first understand what is blood pressure, and why it is important to the circulatory system.

As with any other pressure, BP is a force over an area. In most cases, when practitioners want to measure blood pressure they are really interested in measuring ABP. Aortic blood pressure is the pressure of the blood in the aorta after exiting the left ventricle. Therefore, BP can be understood as the pressure exerted by the blood on the walls of the aorta. This pressure is generated by the heart itself. As the heart contracts to eject blood into the aorta, it generates a pressure wave that propagates through the circulatory system. It is interesting to note that this pressure wave propagates faster than the blood itself is moving. The speed is affected by the compliance of the arterial walls. The more compliant a vessel is, the slower the wave will propagate. For instance, pressure waves move slower in the aorta than the arteries [48]. Due to the intermittent contractions of the heart, BP is pulsatile in nature, meaning the pressure rises and falls back down within every heart cycle. Therefore, BP

is usually given by means of two measurements: the maximum and minimum pressure within a cycle. The maximum pressure is the systolic blood pressure (SBP) and the minimum pressure is the disatolic blood pressure (DBP). For historical reasons, BP is measured in millimeters of mercury (mmHg). It was first measured by measuring the distance by which the blood pressure displaced a column of mercury. We can therefore infer that BP is measured relatively to atmospheric pressure. For instance, normal blood pressure is usually said to be $120/80$ mmHg, where the first number denotes the SBP and the second denotes DBP.

The main purpose of the circulatory system is to provide the tissues of the body with oxygen and nutrients. The metabolic need of a tissue dictates how much oxygen and nutrients are required by the tissue. The metabolic need can increase or lower depending on various factors, such as the type of tissue and activity of the tissue. For instance, during heavy exercise, the metabolic need of skeletal muscle can increase by 20 times its resting baseline. In order to accommodate the variations in metabolic needs, the circulatory system increases the blood flow to the tissues with a higher metabolic need and lowers blood flow to the ones with a lower metabolic need. Meanwhile, the aortic blood pressure is kept remarkably constant. Fast changes in local blood flow are achieved by constricting or dilating the arterioles and capillaries for the receiving tissue. An interesting parallel can be drawn between blood pressure and Ohm's law. The blood pressure in a circulatory system can roughly be modeled by a simple electrical circuit, where the blood pressure acts as the voltage source, blood flow takes the role of current, and the arterioles and capillaries act as variable resistances. Therefore, the heart must supply sufficient cardiac output to fulfill the needs of the tissues. Tissues have a critical closing pressure that the heart must overcome in order to allow blood to flow. If the heart is unable to overcome this pressure, the blood flow to the tissues stops and can not fulfill its metabolic requirements.

### 2.1.2 Blood Pressure Monitoring

Blood pressure is one of the most commonly monitored characteristics of the circulatory system. Monitoring options are usually either classified as invasive or non-invasive monitoring.

Invasive blood pressure monitoring usually only comes in the form of an arterial line. To monitor BP with an arterial line, a catheter is inserted into the patient. This catheter is connected to a

tube containing a saline solution. Nowadays, the tube connects to a calibrated pressure transducer that can provide readings in mmHg. As the blood pressure changes, it pushes the saline solution. This motion in the saline solution is picked up by the pressure transducer and converted to a pressure reading. The catheter can be inserted in various arteries, but it is most commonly inserted in the radial artery [45]. Arterial lines are considered to be the gold standard of blood pressure monitoring. They provide an accurate and continuous reading of the pressure. Due to the invasiveness of catheterization, arterial lines can lead to complications. The most common complications are infection and sepsis [55].

Two types of non-invasive blood pressure monitoring methods exist; cuff-based methods and volume clamps. Inflatable cuff-based BP monitoring is the most well-known monitoring method and is extensively used for home monitoring. The method operates by placing a cuff around the arm of the patient. The cuff is inflatable, thereby increasing the pressure around the arm. Different techniques can be used to monitor the pulse around the cuff and detect when the inflatable cuff pressure is equal to the SBP and BP. Some techniques are automatic while others are manual. However, all these cuff-based methods hold the same advantages and drawbacks. The main advantage of this method is that it is non-invasive and therefore has little to no adverse health effects in most cases. The largest disadvantage is that the monitoring is not continuous, causing large time intervals between readings. These time delays can have negative repercussions on the care of patients. Turan et al. [68] found that common delays of 4 to 6 hours between readings can cause episodes of hypertension and hypotension to be outright missed by the monitoring. Intraoperative hypotension has been associated with higher risks of myocardial injury, acute kidney injury, and death [70], while hypertension shown to increase risks of myocardial injury [58], infarction [59, 54], acute kidney injury [65], and mortality [40, 37, 63]. Had these episodes not been missed, proper responses could have been taken by the medical personnel to minimize their adverse effects. The volume clamp method aims to overcome the issues of cuff-based methods. The volume clamp combines an inflatable cuff and a photoplethysmogram to correlate the changes in blood volume with blood pressure. This method is continuous and non-invasive, however, the accuracy of the readings is insufficient, the system is susceptible to motion artifacts, and is more expensive than other methods [38].

## 2.2 Literature Review

Since the 1970s, researchers and medical practitioners alike have searched for a way to monitor BP that was non-invasive and accurate. In more recent years a lot of research for non-invasive BP monitoring has been interested in experimenting with machine learning methods. These works have also mainly focused on using the signal from a photoplethysmogram as input. The literature for this domain of research can be divided into three sections: feature engineering, machine learning models, and data.

### 2.2.1 Feature Engineering

Feature engineering can make or break a machine learning solution. Blood pressure estimation is no different. In the BP domain, physiological signals are most often used as input. These can aptly be defined as time series. In the majority of cases, a photoplethysmography (PPG) signal is used, although some works use additional signals or features. Feature engineering for BP estimation can be divided into two types: manual and automatic.

#### A. Manual Feature Engineering

As previously noted, PPG plays an important role in BP estimation. PPG is a non-invasive measurement method widely used in clinical settings to measure various properties of the circulatory systems, such as heart rate, oxygen concentration, and respiration rate. Most importantly, PPG can detect variations in blood volume [3]. The method is typically implemented by using an light emitting diode (LED) to emit light on the skin of a patient, a photodiode then measures the amount of light reflected by the skin. The wide adoption of the method can be attributed to its variety, inexpensiveness, and ease of installation. The blood flow measured by PPG is directly affected by the ABP [36]. As such, PPG is an indirect measure of ABP, PPG shares many features of the ABP waveform. However, the relationship between the PPG and ABP waveform is not fully understood and can vary depending on the physiological characteristics of the patient's circulatory system and even the position of the PPG sensor. Most of the literature on non-invasive blood pressure measurement through machine learning has relied on extracting information from the PPG waveform.

Historically, a lot of feature engineering has been focused on manually extracting salient features from a single PPG signal. This type of feature extraction is called pulse wave analysis (PWA). It has the advantage of being cheaper and simpler since it only requires one sensor. The features are most often composed of time domain features, such as the height and width of the waveform at various key points. For example, El-hajj and Kyriacou [12] extracted 52 features from the PPG signal, its first derivative, and second derivative, as input features to an LSTM-based deep learning model. Their approach achieved a performance of 4.51 and 2.6 MAE for SBP and DBP respectively. Less frequently, PWA features are crafted from the frequency domain, such as the amplitude and frequency of FFT peaks. Li and He [32] only used the harmonics of the PPG signal as input for a GRNN, and reported 3.96 and 2.39 MAE for SBP and DBP respectively. Ma et al.[34] used manually extracted features in the time and frequency domain as input for a Transformer-inspired architecture and achieved estimation performance of $4.303 \pm 6.378$ and $3.134 \pm 4.489$ MAE for SBP and DBP. In Roy et al.[52], over 200 features were extracted from the PPG signal including features from continuous wavelet transforms, and were used as input for a deep learning model employing residual convolutional blocks and 2 fully connected layers. The model achieved performance results of $2.38 \pm 3.24$ MAE and $1.23 \pm 1.73$ for SBP and DBP. In [77] The amplitude and phase were obtained with an FFT and used for estimation with only 2 fully connected layers and achieved an impressive error of $0.06 \pm 7.08$ and $0.01 \pm 4.66$ MAE for SBP and DBP. Lastly, In Li et al.[32], the authors used the harmonics of the signal as input for a GRNN and achieved $3.96$ and $2.39$ MAE for SBP and DBP. In Pandey et al. [46], the authors mixed FFT-based features with time-domain features from the first and second derivative of the PPG signal and obtained estimation errors of $2 \pm 6.08$ and $1.87 \pm 4.09$ MAE for SBP and DBP. Even more rarely, statistical features are extracted from the signal [17, 52, 9]. They include metrics such as the mean, standard deviation, skewness, and kurtosis. Roy et al. [52] Included 11 statistical features alongside other time domain and autoencoder-based features. They calculated, standard deviation, mean absolute deviation, skewness, kurtosis, interquartile range, approximate entropy, spectral entropy, Hjorth complexity, Higuchi fractal dimension, and detrended fluctuation analysis. They then used these features as input for a deep learning model. Extensive research has been done in exploring PWA features. In their work, Dey et al. [10] extracted 233 different features from the PPG signal, which

included time domain, frequency domain, and physiological features. Their approach reported performances of 7.8 and 6.1 MAE for SBP and DBP respectively. Further miscellaneous features have also been explored, such as the heart rate [31, 79, 76, 33, 41], Teager–Kaiser energy [51, 41], K value [51, 34], energy profile [51, 41], PPG Intesity Ratio (PIR) [16], and margin factor [34].

Removing the requirement for using a single sensor, an approach that is often used is to use two PPG readings to measure the pulse transit time (PTT). The PPG signals are recorded at two different locations, and the PTT is derived by calculating the time taken by the pulse waveform to propagate from the first location to the other. The hope in using this method is that the PTT can help determine some physiological characteristics of the patient's circulatory system [43]. An important physiological property that can affect the relationship between blood volume and blood pressure is the ability of arteries to contract or distend in response to changes in pressure. This property is referred to as the compliance of the arteries. For instance, Yen et al. [79] Derived PTT by measuring PPG at the index and middle finger. Using other features derived from the PPG waveform, they trained a deep learning model to acquire pressure measurement from the PPG signal. They reported a performance of 0.17 and 0.52 MAE for SBP and DBP respectively.

A more popular, but similar approach to PTT is to use an ECG signal to measure pulse arrival time (PAT). An ECG measures the electrical activity in the heart. Some features of the ECG waveform have been associated with stages of the heart cycle. Most importantly, the R peak corresponds with the contraction of the ventricles. The PAT is defined as the time difference between the R peak of the ECG signal and the systolic peak in the PPG signal. Therefore, PAT approximates the waveform propagation time from the aorta to the PPG site. As noted by El-Hajj and Kyriacou [11], PAT and PTT are often wrongly used interchangeably in the literature. Li et al. [31] extracted 7 features from the PPG and ECG signal, including PAT, and reported 0.7357 and 0.5587 MAE for SBP and DBP respectively. In Farki et al. [16], the authors extracted 3 features from the PPG and ECG signals, including PAT. They reported a performance of 2.561 and 2.231 MAE for SBP and DBP respectively.

Some researchers have also suggested using demographic information from the patients as additional input to the models [75, 9, 56, 10]. This demographic information can act as an approximation to the physiological properties of subjects. Xing et al. [75] added the patient's weight and

BMI as features alongside time domain features. They concluded that including physiological information might improve the performance of BP estimation algorithms. These features typically include weight, height, age, sex, and BMI. Yang et al. [78] concatenated the subject's features to the automatically extracted features. They concluded that adding demographics features helps with estimation, with their best model achieving $4.43 \pm 6.09$mmHg and $3.23 \pm 4.75$mmHg MAE for SBP and DBP. Dey et al.[10] employed a different technique by partitioning their training set based on demographic thresholds. They found that this split helped with estimation performance. the partitioning scheme reduced the error, starting from $7.8 \pm 10.4$ and $6.1 \pm 7.1$ for SBP and DBP without the partitioning to $6.9 \pm 9$ and $5 \pm 6.1$ for SBP and DBP with the partitioning. These findings are supported by [2, 60, 41, 9, 56] who also used demographics information and achieved good performance. These results indicate that demographic information might help the models estimate the properties of the subject's circulatory system.

### B. Automatic Feature Engineering

Proper PWA requires the implementation of a complex algorithm to filter the signals, extract the heartbeats, and finally extract the features. Therefore, an increasingly popular approach is to rely on deep learning models to perform automatic feature extraction. The most popular input methodology is to split the PPG signals into segments, and subsequently rely on CNN or RNN layers to perform feature extraction [4, 5, 20, 47, 82]. The most popular segment sizes were 8 seconds [80, 7, 26, 66, 81] or 5 seconds [2, 15, 30, 53]. Schlesinger et al. uniquely picked a window of 30 seconds [56]. A less frequent approach is to split the signal into single cycles [71]. Shimazaki et al. [61, 60] have explored the efficiency of automatic feature extraction compared to manual features. In "Features Extraction for Cuffless Blood Pressure Estimation by Autoencoder from Photoplethysmography" [60] the authors compared the performance of manual PWA and demographics features against autoencoder-generated features. They reported a correlation coefficient (R) of $0.67$ and $0.72$ for conventional features and autoencoder features respectively, showing autoencoder features outperformed the conventional features. In their other work [61], the authors compared automatically extracted features through a CNN against a multiple regression analysis using conventional features. They concluded that the CNNs outperformed the conventional features. The correlation

coefficient for the CNN was $0.71$ and $0.63$ for the conventional features. Their results demonstrate the superiority of automatic feature extraction for BP estimation.

### 2.2.2 Machine Learning Models

A variety of models have been proposed to model a subject's hemodynamic system. Mathematical models can generally provide approximations for the properties of the system. More recently, researchers have explored machine learning models to determine if they could more accurately describe hemodynamic systems. For BP estimation, we can break down the models into classical machine learning models, deep learning models, or ensemble methods.

#### A. Classical Machine Learning

Several studies have explored classical machine learning algorithms for BP estimation. Classical machine learning (ML) algorithms usually require manual feature engineering. Various algorithms have been explored in the literature. The tested algorithms include XGBoost [17, 19], SVM [25, 39, 51], random forest [76, 41, 16], linear regression [74, 67, 10], KNN [44], and GPR [9]. Xie et al. [74] compared the performance of various ML models on a BP estimation task. They compared linear regression, artificial neural networks, decision tree regression, bagging regression, and random forest. Their results show that random forest outperforms other types of ML models. Other works in the literature have also shown good performance from random forest models [76, 41, 44, 16]. Yang et al. [78] also compared various ML algorithms against deep learning models. While they also concluded that random forest performed the best out of the classical ML models, they showed that deep learning models outperformed classical ones. These conclusions show that, while the classical machine model might perform acceptably, deep learning networks show more promise.

#### B. Deep Learning

Deep learning has been able to solve a wide variety of long-standing machine learning problems. This justifies its increase in popularity since the 2010s. As a result, it has been more extensively explored for BP estimation in recent years. The interfaceable nature of neural network models

makes it easy to create new types of layers. The following deep learning layers have been applied to BP estimation: dense, CNN, RNN, attention mechanisms, and self-attention.

With the popularity of Deep Learning, many studies have explored using deep learning models for BP estimation. As aforementioned, the superiority in automatic feature extraction and improved performance of deep learning architectures make it an enticing approach. Hence, deep learning might be able to provide continuous, calibration-free, non-invasive BP estimation. For instance, [4] and [53] both have used a very deep architecture inspired by U-net with good performance. [62] and [57] have both implemented models inspired by ResNet and observed good performance from the model. [57] and [71] used AlexNet for estimation, also with good performance. Additionally, Yang et al., in [78], compared deep learning models to classical machine learning models and showed that deep learning models consistently outperformed the classical ones. These results show that more complex deep learning models, both wider and deeper can improve estimation performance.

With the ever-increasing ubiquity of CNNs, studies have explored using them to automatically extract features from the PPG waveform. Their ability to extract minute features from the signal while keeping few parameters makes them a good candidate for time series tasks. For example, Schlesinger et al.[56] explored using CNNs to extract features from a PPG spectrogram. The results demonstrate that CNNs are capable of extracting the proper features from the signal. Works in the literature have used CNNs to perform feature extraction on PPG and ECG signals for BP estimation [7, 53, 23]. CNNs have been applied both with complimentary layers [80, 79, 81, 61, 73], such as pooling and batching layers, or plainly [15, 30, 66, 78, 56]. A residual CNNs architecture is also sometimes used [82]. In some cases, a popular model or architecture is used [62, 4, 26, 71]. For instance, Schrumpf et al. [57] used the AlexNet [28] and ResNet [21] architectures, which both heavily rely on CNNs to perform BP estimation from remote PPG signals. They reached an MAE of $15.2 \pm 9.11$ mmHg SBP and $8.52 \pm 4.92$ mmHg DBP with the AlexNet model. The ResNet model reached an MAE of $12.51 \pm 12.61$ mmHg SBP and $8.3 \pm 9.84$ mmHg DBP. In the literature, approaches using CNNs for automatic feature extraction from the PPG signal generally report good performance. These results indicate that CNNs are better at extracting relevant features in the PPG signal than manual feature extraction.

RNNs are a special type of neural network layers that are specifically designed to deal with sequential data. Regular RNNs can have trouble dealing with long-range dependencies. Two variants of the RNNs have been developed to overcome this issue: LSTMs and their simpler counterpart, GRUs. Since the PPG signal is a time series, they're a natural method to try for BP estimation. El-hajj and Kyriacou [12] used BiLSTM, BiGRU, LSTM, and GRU layers as feature extraction layers for BP estimation. They reported performances of $4.51$ mmHg and $2.6$ mmHg MAE for SBP and DBP, respectively, using BiLSTMs and GRUs layers. In [31] Li et al. used LSTMs and BiLSTMs and tested different numbers of layers of LSTM. They found that 4 LSTM layers performed the best with estimation performance of $0.7357 \pm 0.9579$ and $0.5587 \pm 0.6829$ MAE for SBP and DBP. RNNs are most often used in conjunction with CNNs. The CNNs perform initial fine-grained feature extraction and the RNNs extract the temporal features. These architectures usually use either LSTMs [80, 66, 81, 78] or GRUs [62, 15, 30, 57]. In [66] Tazarv et al. used a combination of CNN, 2 LSTM layers, and a fully connected layer. Their results show that their model outperformed previous works with performance of $3.7 \pm 3.07$ and $2.02 \pm 1.76$ for SBP and DBP. Furthermore, [20, 47, 23] all used RNNs in their architecture and reported good performance. These results could indicate the importance of modeling time dependencies for blood pressure estimation using PPG.

Because of the vanishing gradient problem, RNNs have trouble extracting relationships between inputs with long-range time dependencies. As such, attention mechanisms have been developed to help RNNs track patterns over long periods of time. Several papers have tried to apply this method for BP estimation. For example, in [26] Kim et al. used a ResUNet architecture and compared its performance with and without an attention-based skip connection. Their results show that the attention mechanism improves MAE by $0.81$mmHg for SBP and by $0.15$mmHg for DBP, with final estimation performance of $5.75 \pm 7.55$ and $2.73 \pm 4.23$ MAE for SBP and DBP. Other studies used an attention-based mechanism in their architecture such as, [2] and [7], Chen et al. [7] reported good performance using this method. This could indicate that, as previously, modeling time dependencies is important, but modeling time dependencies for longer periods of time can further help with estimation.

With the popularity of the Transformer architecture, a variation of the attention mechanism, self-attention, is further being used for time series analysis. A Self-attention layer can pay attention

to any output of its previous layer and is typically used within an encoder or decoder network. However, few works have explored self-attention layers as feature extractors for PPG signals [15, 26, 34]. In [26] Kim et al. compared a ResUNet architecture with and without self-attention, and concluded that self-attention provided the best performance improvement compared to an attention-based skip connection. The self-attention layer improved estimation performance by $1.88$mmHg for SBP and $0.67$mmHg MAE for DBP compared to a model without any attention mechanism. Their architecture using self-attention only boasted an estimation performance of $3.87 \pm 4.42$ and $2.06 \pm 2.48$ for SBP and DBP. Further taking inspiration from the Transformer, Ma et al. [34] architectured a model based on an Informer model. Their model, titled KD-Informer, predicted BP with a performance of $4.303 \pm 6.378$ and $3.134 \pm 4.489$ MAE for SBP and DBP. Moreover, [12] and [15] both employed a self-attention mechanism in their architectures and reported good performance. Considering these results, we can note again the importance of modeling long-range sequence dependencies and the superiority of self-attention mechanisms to detect those patterns.

### C. Ensemble Methods

A few studies have explored the use of ensemble methods for blood pressure estimation. For instance, [17] and [19] both used XGBoost to train a machine learning model, however, the two studies' results vary greatly. Fleischhauer et al. saw $5.799 \pm 7.481$ MAE for SBP for their best model and Haque et al. achieved $2.54 \pm 4.24$ with their best model. Other methods used in the studies could explain this variation. Sadrawi et al. [53] used a genetic algorithm to create an ensemble of deep learning models. Their ensemble model achieved performance of $-1.659 \pm 0.665$ and $0.665 \pm 2.03$ ME for SBP and DBP. In [10] Dey et al. split their training data by demographic properties, then trained linear regression models for each partition. The models were then aggregated by inferring a prediction value with each model, rejecting outlying predictions, and taking the mean of the remaining predictions to get a final output. Their ensemble model achieved $6.9 \pm 9$ and $5 \pm 6.1$ MAE for SBP and DBP. Finally, in [16] Farki et al. clustered the training models based on extracted features using K-means, then trained models for each cluster. Their total error was calculated as the mean of the performance on all the clusters. Their best model, which used gradient boosting as their base model, achieved good performance of $2.561$ and $2.231$ for SBP and DBP. Clustering

the data improved the performance greatly, the model without it achieved performance of $6.367$ and $6.276$ MAE for SBP and DBP. These results indicate that ensemble methods could further improve estimation accuracy.

### 2.2.3 Data

Machine learning methods can find patterns in training data. Thus, data plays a crucial role when training those models. As observed by El-Hajj [12], BP estimation performance seems to improve as the size of the dataset decreases. The size of the dataset seems to be the main driver of performance in the literature. This poses a major issue in creating a non-invasive blood pressure estimation algorithm as generalization performance will lower with dataset size. This observation coincides with Slapničar et al. [62] who observed the lowest errors were achieved on small selected subsets of data, and studies with larger datasets saw larger errors. While splitting the BP estimation values into ranges, Schrumpf et al. [57] observed that the error for a given range depended strongly on the amount of training data for that range. The authors also state that morphological differences between individuals are a main challenge for creating a generalized BP estimation neural network. This finding corroborates with the ones from Shimazaki et al. [61]. While studying the efficiency of using CNN for feature extraction, the authors noticed that the largest errors were incurred by subjects with very high or low SBP values. Taking these findings into consideration, both Ma et al. [34] and Slapničar et al. [62] recommend using a patient-wise train/test dataset split in order to get a more accurate generalization performance. Using this method would make sure no bias is introduced into the predictions by overfitting on the subject's morphology.

Some studies have attempted to address this issue with personalization, which consists of fine-tuning the model using a patient's labeled data. In effect, this calibrates the model for the subject. For example, Slapničar et al. [62] added personalization to their deep learning model, which reduced the prediction error from $15.41$ and $12.38$ MAE for SBP and DBP to $9.43$ and $6.88$ MAE for SBP and DBP. Leitner at al. [30] used a similar approach where patient-specific data was used to retrain some layers of their models, a technique borrowed from transfer learning. Personalization improved the performance of their model, starting from $4.59$ and $2.72$ MAE for SBP and DBP without personalization to $3.52$ and $2.2$ MAE for SBP and DBP with personalization. Lastly, Xing

et al. [76] created a personalized random forest algorithm by subtraction the mean error from the predictions of the uncalibrated model. Personalization can improve performance by a large amount and helps with the challenges we've highlighted in the previous paragraphs. However, this method requires the user to acquire a gold standard measurement through a cuff-based or invasive technique in order to personalize the model. Furthermore, the efficiency of personalization over long periods of time or multiple uses has not been explored.

# Chapter 3

# Blood Pressure Monitoring in a Clinical Setting

## 3.1 Introduction

Blood pressure management is an essential element of perioperative care. Intraoperative hypotension has been associated with higher risks of myocardial injury, acute kidney injury, and death [70], while hypertension shown to increase risks of myocardial injury [58], infarction [59, 54], acute kidney injury [65], and mortality [40, 37, 63]. Adequate BP monitoring can help achieve optimal outcomes for patients. Two methods are generally used to provide BP monitoring: non-invasive and invasive. Non-invasive BP monitoring is usually achieved using oscillometric methods by means of a sphygmomanometer cuff. This method has its drawbacks. It provides intermittent monitoring, every 1 to 5 minutes, which can cause critical delays in capturing changes in BP which can even be missed altogether [68]. Invasive BP monitoring is considered the gold standard and consists of inserting a catheter into a patient's artery. The invasive nature of this method can lead to complications such as vascular injury, bleeding, infection, and sepsis. [55]. Therefore, a continuous, non-invasive method of BP monitoring has long been sought after. More recently, PPG and ML have shown promising results in order to provide accurate, calibration-free, clinically usable, and non-invasive BP monitoring. We set out to determine if machine learning algorithms could produce an accurate, calibration-free, and non-invasive blood pressure monitoring from a single PPG signal.

Current state-of-the-art algorithms often require personalization or calibration of the model to the patient, which is undesirable in a clinical context.

Many deep learning architectures for BP estimation have been presented in the literature. Common layers to use for these architectures are combinations of CNNs and RNNs or GRUs and attention mechanisms [66, 12]. Others even used the phase and amplitude of the PPG signal as features for an artificial neural network [77]. These models usually pass the Association for the Advancement of Medical Instrumentation (AAMI) standard. Model performance is often compared against the AAMI and British Hypertension Society (BHS) standards to evaluate their practical applicability in a clinical setting. The AAMI standard operates on a pass/fail evaluation and states that SBP and DBP should be evaluated separately and should have a mean difference of $5$ mmHg and a standard deviation of $8$ mmHg. Even so, we can not determine if these results are applicable in a clinical setting. The dataset splits used to evaluate the performance of the algorithms often share data from the same recordings and patients. In some cases, the models are trained and evaluated on data from the same recording [66]. More often, the dataset is said to be split into fixed percentages of the dataset. These splitting schemes do not explicitly state the splitting strategy used, but can generally be assumed to be random splits. Again, this leads to shared recordings and patients across the splits [12, 77]. These evaluation methods can not provide accurate performance estimates for a calibration-free algorithm since the trained model has had the opportunity to train on the evaluated patient's data. Other works have attempted to keep the training and evaluation patient data separated. A patient-wise split is most common, but some works have proposed using a leave-one-subject-out split for better evaluation estimates [57, 62]. Unfortunately, these approaches do not qualify for the AAMI standard.

Works in this field are difficult to compare due to several factors. The datasets used for training and evaluation are rarely shared, and preprocessing methodologies can not always be reproduced. Likewise, code implementation for developed models is seldom published, and textual descriptions of the networks often lack details. Moreover, the reported performance is sometimes calculated as MAE, ME, RMSE, or R-squared. The combination of these three factors makes it troublesome to compare and reproduce studies. Our contribution to the literature is threefold:

(1) We first provide a comprehensive review of the literature through a survey.

(2) We reproduce and evaluate models from the literature for a clinical application and publish our implementations on an open-source platform.

(3) We propose new deep learning architectures evaluated against the reproduced models for a clinical context. Implementation for our models is also published.

These contributions help tackle the gaps we identified in the literature. The survey and reproduction allow comparing models under the same dataset and implementation conditions, allowing for a better overview of the state-of-the-art in BP estimation. Sharing our implementation will also make it easier for future contributors to compare their works. We designed our proposed models to cover a wide range of popular deep learning layers. This selection helps us evaluate which type of layer architecture performs best for the task.

Section 3.2 presents a summary of the current literature on surveys for non-invasive blood pressure monitoring through machine learning. In Section 3.3, we conduct a survey to find reproducible state-of-the-art models. Section 3.4 describes our datasets, input modalities, custom models, and reproduction implementations used for our experiments. Section 3.5 presents the results of our experiments. In Section 3.6, We interpret our results and explain our arguments. Section 3.7 closes the chapter and describes avenues for potential future works.

## 3.2 Related Work

Previous reviews on non-invasive blood pressure monitoring have been conducted in the literature. We looked for related reviews on the search engines: Google Scholar, Sofia, and PubMed. We searched using the following keywords: “machine learning”, “arterial pressure”, “hypertension”, “monitoring”, “non-invasive”, and “review”. Our search yielded a collection of 9 review articles. We compare the reviews against 5 characteristics describing the aspects of BP estimation that was reviewed. The characteristics are as follows: (1) BP estimation devices, (2) classical machine learning algorithms, (3) deep learning algorithms, (4) single PPG waveform only, and (5) reproduction of the models. All reviews are shown against their evaluation characteristics in Table 3.1.

El-Hajj and Kyriacou [11] conducted a review covering PWA, PAT, PTT, pulse wave velocity, classical machine learning, and deep learning approaches. They concluded that the state-of-the-art

methods require calibration in order to provide acceptable results. Rastegar et al. [50] performed a review mainly focusing on the methods for non-invasive BP estimation, the reviewed methods covered PWA, PAT, and PTT. Their review also included an overview of deep learning algorithms. They concluded that PTT-based methods along with deep learning are the most promising methods for BP estimation. Ismail et al. [24] presented a review on the stat-of-the-art and commercial devices for non-invasive BP monitoring. They concluded that new methods can boast improved estimation performances. Kim et al. [27] presented a meta-analysis of non-invasive BP estimation methods compared to a gold standard. They selected non-invasive methods, as measured by commercial devices. Their analysis showed that no commercial devices met the AAMI standards for BP measurement. Quan et al. [49] reviewed cuff-based and cuffless non-invasive BP monitoring methods and presented a capacitive proximity sensing method for BP estimation. Their work concludes that a large and diversified population is required for accurate BP prediction. Stojanova et al. [64] reviewed cuff-less BP estimation methods, including devices. They conclude that PPG and ECG are the most common signals for non-invasive BP monitoring and that deep learning approaches provide promising results in terms of prediction accuracy. Le et al. [29] performed a review on sensors from which PAT and PTT can be determined. They further reviewed the state-of-the-art for PTT/PAT-based approaches. Hosanee et al. [22] reviewed the sites for single PPG monitoring, as well as, the dataset used in BP estimation studies. They recommended using a finger photoplethysmogram and improving the diversity of the datasets. Meidert and Saugel [38] presented techniques for non-invasive BP monitoring and reported the validity in a clinical setting. Their analysis shows that non-invasive methods cannot always be replaced with invasive methods.

Our survey conducts a review of deep learning methods for BP estimation using a single PPG waveform. To our knowledge, our survey is the only review focusing exclusively on deep learning models using only a single PPG signal as input. We further differentiate by being the only review including reproduction of the reviewed models, allowing for a more accurate comparison of the reviewed methodologies.

Table 3.1: Related review articles compared against our work

| Review | Device | Classical ML | DL | S. PPG Wav. | Reproduction |
|---|---|---|---|---|---|
| El-Hajj and Kyriacou [11] | No | Yes | Yes | No | No |
| Rastegar et al. [50] | No | Yes | Yes | No | No |
| Ismail et al. [24] | Yes | Yes | Yes | No | No |
| Kim et al. [27] | Yes | No | No | No | No |
| Quan et al. [49] | Yes | No | No | No | No |
| Stojanova et al. [64] | Yes | Yes | Yes | No | No |
| Le et al. [29] | No | Yes | Yes | No | No |
| Hosanee et al. [22] | No | No | No | Yes | No |
| Meidert and Saugel [38] | Yes | No | No | No | No |
| **Our work** | **No** | **No** | **Yes** | **Yes** | **Yes** |

## 3.3 Survey

We first begin this study by conducting a survey of the machine learning models used for blood pressure estimation in the literature. We specifically bring our attention to models that used a PPG signal as input. We conducted our survey by conducting a keyword search in various databases in order to find relevant articles. We then filtered those articles in order to retrieve a subset of architectures that were promising and comparable.

We employed a variety of keywords across different databases to retrieve our candidate architectures. Our searches were based on the following keywords “machine learning”, “deep learning”, “artificial intelligence”, “blood pressure”, “ABP”, “arterial pressure”, “hypertension”, “estimation”, “prediction”, “monitoring”, “PPG”, “photoplethysmography”, “Transformer”, “ensemble”, “convolution”, “attention”, “continuous”, “non-invasive”, and “cuffless”. Our searches were spread across 3 different search engines: Google Scholar, The PubMed search engine, and the Sofia[1] specialized search engine. Google Scholar and Sofia are both database aggregators searching across multiple journals. From those searches, the articles were manually reviewed and picked according to their relevancy. The articles were then annotated with different properties in order to further evaluate their relevancy. Some of those properties include the date of publication, the dataset used, the dataset size, the architecture of the models, and the reported performance. The resulting collection of models is published online and can be viewed the following link.

[1]The Sofia searches across multiple libraries and databases, such as the WorldCat database. https://sofia-biblios-uni-qc.org/en/

Various machine learning techniques have been proposed for estimating BP. We propose a taxonomy, shown in Figure 3.1, to help classify and understand the approaches.

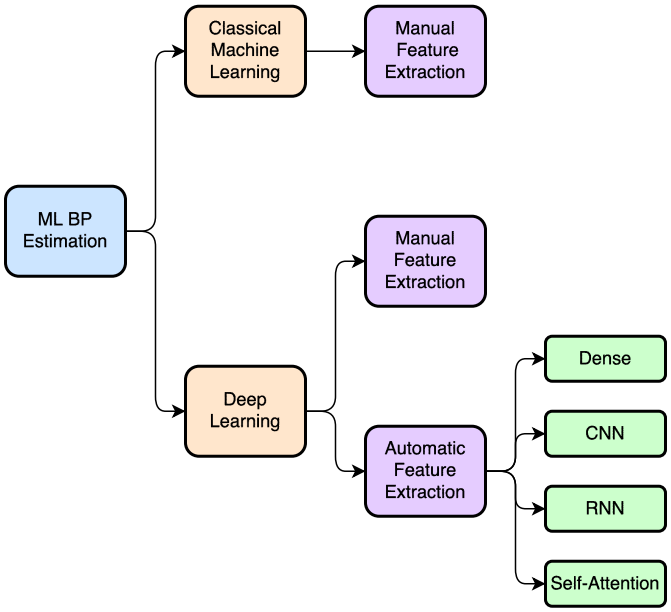
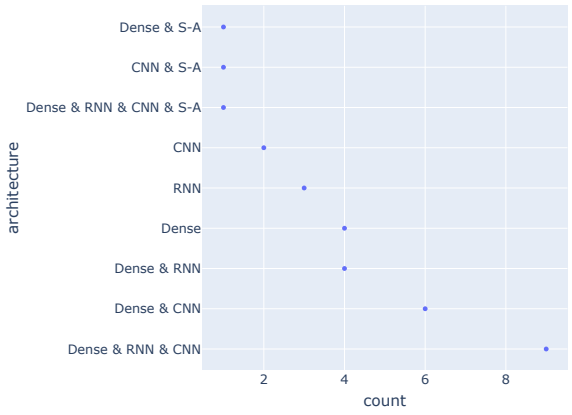



Figure 3.1: Taxonomy of approaches for BP estimation through ML

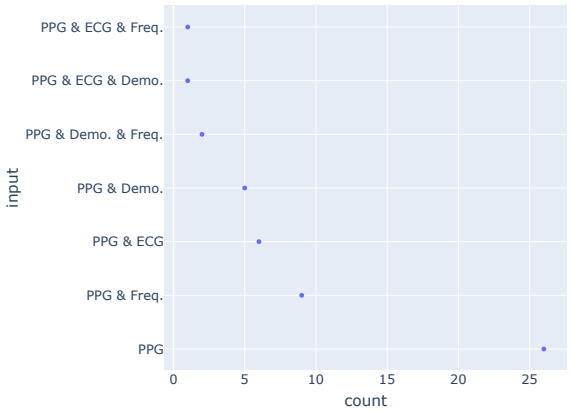
As part of the proposed approaches, a variety of learning algorithms, deep learning layers, and input features have been proposed. Figure 3.2 shows the popularity of each approach. We note that over half of the works proposed a deep learning architecture. Of these deep learning architectures, the most popular combination of layers was dense, recurrent, and convolutional layers. We also note that using only PPG signals was the most popular input type.

Figure 3.3 shows the distribution of the publication years of the collected works. The vast majority was published after 2018. A total of 113 proposed models were evaluated and reported using MAE. Of these models, the distribution of their reported performances is shown in Figure 3.4. Both the MAE and standard deviation (SD) performances are shown. The vertical line shows the requirement threshold for the AAMI standard. According to the reported performance, 36 models passed the AAMI standard for SBP and 54 for DBP. A total of 36 models passed the standard for both SBP and DBP, which represents roughly 32% of the proposed models.

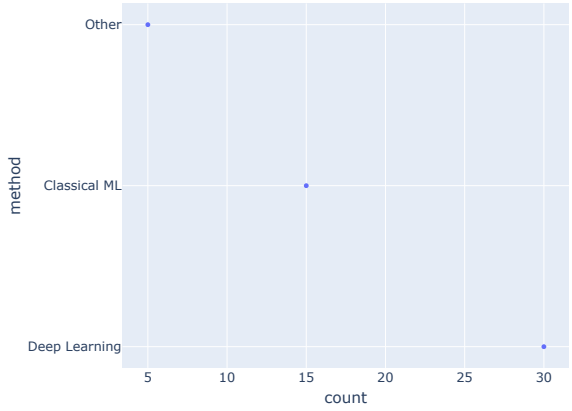
In order to find promising architectures from the candidates, we eliminated some articles by filtering with the annotated properties. First, we narrowed down the input data used by taking the




(a) Deep learning layers




(b) Input signals and features




(c) Learning algorithm

Figure 3.2: Distribution of the methods used in the collected articles

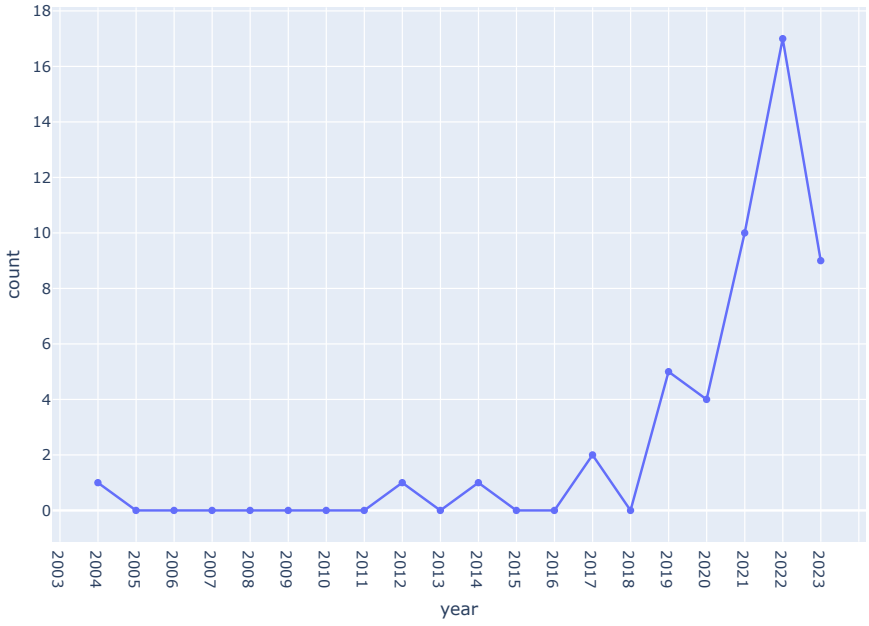
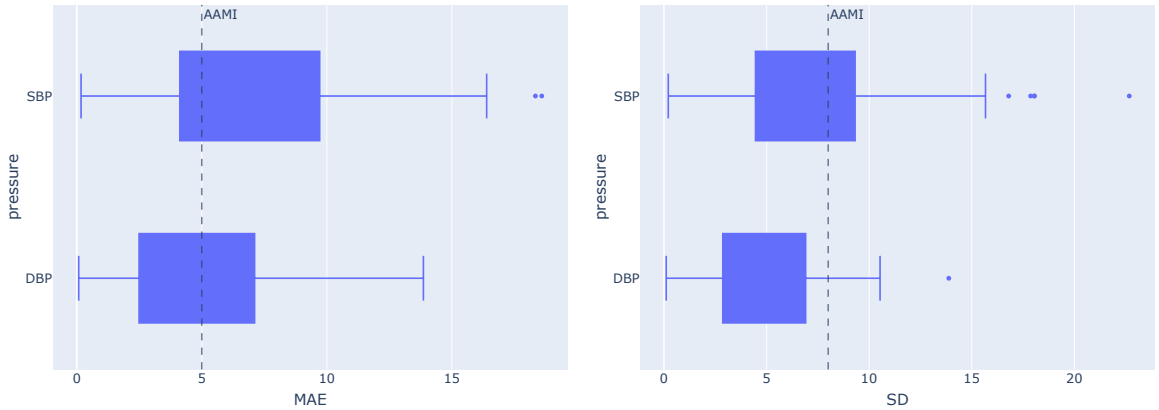



Figure 3.3: Distribution of years of the collected articles




(a) MAE (b) SD

Figure 3.4: Distribution of reported performances from the collected articles

articles that only used a PPG signal as input. This meant excluding the articles that used ECG and demographics information. Since features extracted automatically by CNN and autoencoders perform better than manually extracted features [60, 61], we narrowed down our candidates to models that used automatic feature extraction. We then excluded articles that used a self-collected dataset, as it would be harder to compare against their reported results. We kept only models trained on a MIMIC or UCI dataset. Finally, For the sake of comparison, we filtered to only keep models that predicted SBP and DBP values and measured prediction performance in MAE. Some articles did not report the performance results on its prediction and therefore had to be excluded. By the end of this elimination process, 8 articles remained which totaled 9 unique architectures. The resulting articles are presented in Table 3.2.

In addition to the model architectures, the datasets used to train the models are important to evaluate the performances reported by the authors. Special attention must be given to the number of subjects and total amount of data. For the sake of comparison, we evaluated the amount of data in hours of signal. In general, it is harder to achieve good performance on datasets with more subjects, but they will generalize better in a clinical setting. We believe the amount of subjects is the main factor that contributes to a model's generalization performance. The datasets employed by each article are shown in Table 3.3. We also include the dataset split used for evaluation. In the works of Slapničar et al., the impact of sharing patients in the training and test set is well shown. In their work, the authors first used a leave-one-subject-out dataset split to evaluate the performance of their

Table 3.2: Architectures of the remaining models after filtering

| No. | Title | Authors | Architecture |
|---|---|---|---|
| 1 | Blood Pressure Estimation from Photoplethysmogram Using a Spectro-Temporal Deep Neural Network | Slapničar et al. [62] | spectro-temporal ResNet inspired |
| 2 | An Estimation Method of Continuous Non-Invasive Arterial Blood Pressure Waveform Using Photoplethysmography: A U-Net Architecture-Based Approach. | Athaya and Choi [4] | U-Net inspired |
| 3 | A new deep learning framework based on blood pressure range constraint for continuous cuffless BP estimation. | Chen et al. [7] | RSPAN |
| 4 | Continuous Blood Pressure Estimation Using Exclusively Photopletysmography by LSTM-Based Signal-to-Signal Translation. | Harfiya et al. [20] | LSTM autoencoder |
| 5 | DeepCNAP: A Deep Learning Approach for Continuous Noninvasive Arterial Blood Pressure Monitoring Using Photoplethysmography. | Kim et al. [26] | ResUNet with attention-based skip connections and self-attention |
| 6 | Personalized Blood Pressure Estimation Using Photoplethysmography: A Transfer Learning Approach. | Leitner et al. [30] | CNN, GRU, Dense |
| 7 | A Deep Learning Approach to Predict Blood Pressure from PPG Signals. | Tazarv and Levorato [66] | CNN, LSTM, Dense |
| 8 | Assessment of deep learning based blood pressure prediction from PPG and rPPG signals. | Schrumpf et al. [57] | AlexNet |
| 9 | Assessment of deep learning based blood pressure prediction from PPG and rPPG signals. | Schrumpf et al. [57] | ResNet |

algorithm. In a leave-one-subject-out validation split, a single patient is first reserved for model evaluation and the rest are used for model training. Multiple runs like these are repeated until all subjects have been used for evaluation once. The performance results for all runs are then averaged together. They then showed that by adding a personalization step by training the model on some data of the evaluated patient, the performance improved from $15.41$ mmHg MAE to $9.43$ mmHg MAE for SBP and $12.38$ mmHg MAE to $6.88$ mmHg MAE for DBP. Their results show that BP estimation from PPG is extremely patient-dependent. In some cases, we were unable the recover an estimate of the recording hours in the dataset from the article. Those articles are noted with "-" in the recording hours column.

Table 3.3: Datasets of the remaining articles after filtering

| Authors | Validation Method | Subjects | Recording Hours |
|---|---|---|---|
| Slapničar et al. [62] | Leave-one-subject-out | 510 | 700 |
| Athaya and Choi [4] | 70/15/15 | 100 | 195 |
| Chen et al. [7] | 10-fold cross-validation | 1562 | - |
| Harfiya et al. [20] | 70/10/20 | 5289 | - |
| Kim et al. [26] | 10-fold cross-validation | 2064 | 374.43 |
| Leitner et al. [30] | 5-fold cross-validation per subject | 100 | 1000 |
| Tazarv and Levorato [66] | Leave-one-window-out, one model per patient | 20 | 1.66 |
| Schrumpf et al. [57] | 75/12.5/12.5, patient-wise | 5000 | - |

The candidate architectures will be reproduced and evaluated against a classical random split for comparison with their original works and against a patient-wise split to determine if their performance is adequate for a clinical context.

## 3.4 Materials and Methods

### 3.4.1 Dataset

For our experiments, we used physiological signals from a freely available dataset. The MIMIC datasets stand as the most widely used datasets in cuff-less blood pressure estimation. They are distributed freely by PhysioNet. For our research, we used the most recent version of the dataset: MIMIC-IV Waveform [42, 18]. The wide adoption of this dataset by works in the literature makes this the best option to enable comparison of our work with related works. The dataset is composed

of various physiological signals and measurements taken from critically ill intensive care unit (ICU) patients at the Beth Israel Deaconess Medical Center. The dataset contains 200 records spread across 198 patients. In their work, Slapničar et al. [62] note the importance of having a wide variety of subjects rather than pure hours of recording in order to achieve generalization. For our purposes, we are mainly interested in recordings of PPG and ABP. The datasets contain 2886.61 and 1081.23 hours of recording for PPG and ABP respectively. The signals are not cleaned and require significant preprocessing to extract valuable information. Both PPG and ABP are sampled at 125 Hz.

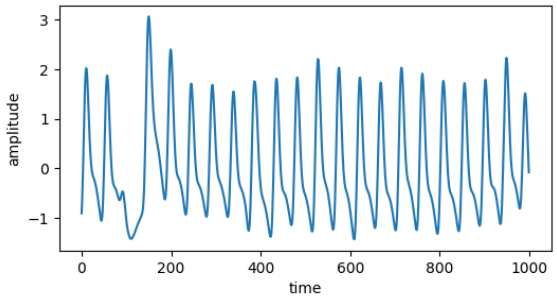
### 3.4.2 Preprocessing

Previous works in the literature usually use two input modalities for automatic feature extraction. (1) The input signal can be split into windows of constant length, for example, Tazarv and Levorato [66] uses rolling windows of 8 seconds with a stride of 2 seconds. (2) The input signals might be segmented into individual heartbeat [71].

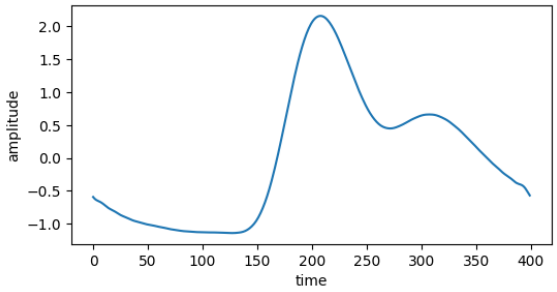
For this study, we use three input modalities, including two modalities from the literature and a new one. (1) Window of signal (Section B.), (2) single heartbeat (Section C.), and (3) sequence of heartbeats (Section D.). We chose these input modalities to improve the ease of comparison with other works. In the sequence of heartbeats, we create a 3-dimensional representation of our data using taking 2-dimensional single heartbeats. Ten subsequent individual heartbeats are concatenated together on a third axis. To our knowledge, this is the first work using such an input modality. Figure 3.5 shows examples for each input modality.

Since the input signals are noisy, we must first clean and filter our input data before using it as training data for our models. Some data cleaning steps are common to all preprocessing pipelines, they are detailed in Section A. The input modalities are implemented as three fully automatic preprocessing pipelines.

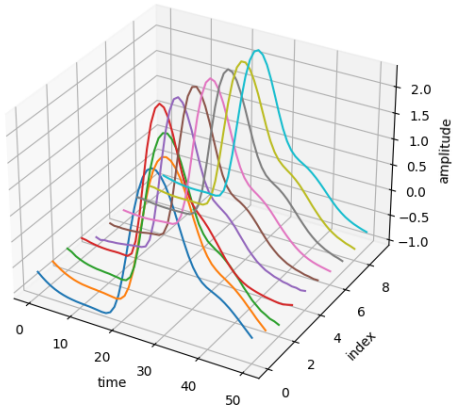
BP estimation from PPG is extremely patient dependent [62]. Therefore, sharing patients between the training and evaluation sets can drastically change the performance of the model. Keeping this in mind, we compare performance on two dataset splits, the first dataset split is a patient-wise split and the second is a random split. For the patient-wise split, we first split the patients into a 70/15/15% scheme before preprocessing to make sure no information on the patients is shared




(a) Window




(b) Heartbeat




(c) Heartbeat sequence

Figure 3.5: Input modality examples

between the dataset splits. For the random split, we first preprocessed the patient's data, the resulting data was then merged into a common dataset, this dataset was shuffled, and finally split into a 70/15/15% scheme so that patient information could be shared between the three sets.

## A. Cleaning

As mentioned in the previous sections, the PPG and ABP signals from the MIMIC dataset contain a lot of noisy data. The signals must be cleaned and filtered to remove low-quality and noisy inputs. The preprocessing steps in this section are reused in all input modality preprocessing pipelines, but their order and hyperparameters might vary from one pipeline to another. We mostly rely on preprocessing steps from the literature to clean the data. In total, four steps are reused in all

pipelines: (1) Reject invalid recordings; (2) filter the signal; (3) filter the input using a signal quality index (SQI); and (4) filter based on pressure values.

(1) Rejection of invalid recordings. Some recordings in the dataset might not contain data for the PPG, ABP, or Both signals. We start all pipelines by rejecting these recordings. Similarly, some recordings contain missing values in their signal recording, which can cause numerical calculation errors in subsequent preprocessing steps. We remove these values from the signals while keeping the remainder of the signal. In the case the entire signal is missing, we repeat the first step of rejecting recordings with no signal.

(2) Signal filtering. The PPG and ABP signals might contain some high-frequency and low-frequency signal noise. This noise can be removed with a signal processing filter. We rely on the literature to choose an appropriate filter. Butterworth low-pass and band-pass filters are the most often used. We reproduce Tazarv and Levorato [66]'s filtering steps. A 2nd-order band-pass Butterworth filter is applied to the PPG signal with a lower bound of 0.1 Hz and an upper bound of 8 Hz. For the ABP, we use a 2nd-order low-pass Butterworth filter with a cutoff of 5 Hz. This step is applied after rejecting invalid signals as described in the previous paragraph.

(3) SQI Filter. In some cases, the PPG and ABP signals contain sections where the signal is dominated by various artifacts and is unusable. In most cases, these artifacts are generally motion artifacts or noise artifacts [13]. Further artifacts could include flat lines from the sensor being removed or flat peaks from unknown sensor issues [62]. To remove these noisy inputs, we filter the input based on an SQI. Elgendi et al. [13] compared the performance of different signal metrics when used as an SQI. They found that skewness performed the best, as such we use skewness as our SQI in our work. SQI filtering is applied at a different preprocessing step depending on the input modality. It is applied after the data is shaped into its modality and is calculated on the PPG signal. We empirically found the best threshold values for filtering. For input windows, we use a lower bound of 0.35 and an upper bound of 0.8. For single heartbeats, we found that the bounds were better extended to a lower bound of 0.5 and an upper bound of 2. For sequences of heartbeats, we rejected the whole sequence if

a single heartbeat in the sequence was out of the bounds for a heartbeat.

(4) Filter pressure. After adding the output pressure values to the signal, we found some extreme BP values that should only occur in extreme circumstances. We decided to remove those inputs, as often done in the literature [66, 23]. This step helps further improve the quality of the input data. It is applied after adding the pressure labels to the input data. We keep inputs where the SBP and DBP pressures are between 30 mmHg and 230 mmHg. For a sequence of heartbeats, we keep inputs where all the SBP and DBP in the sequence are within 30 mmHg and 230 mmHg.

### B. Window

Separating the input signal into segments of constant length is one of the most common input modality for BP estimation from PPG. This method is often used in other machine learning tasks dealing with time series, therefore it is natural to extend this method to BP estimation. Additionally, the preprocessing for this input modality is easy to implement. The resulting input segment will usually contain many heartbeats. This enables the model to extract information from the temporal nature of the PPG waveform. Rolling window is usually used in conjunction with automatic feature extraction. Since the heartbeats are not isolated, it makes it harder to extract manual features.

We implement window preprocessing in our research to make it easier to compare our results with other results from the literature. We base our preprocessing on Tazarv and Levorato.'s [66] work, using windows of 8 seconds with a stride of 2 seconds, resulting in an overlap of 6 seconds. We preprocess the PPG and ABP recordings at the same time. The steps go as follows:

(1) Rejection of invalid recordings. This step is executed as described in Section A.

(2) Filter signal size. Some recordings in the dataset are very short, as such we must very that at least one time window fits in the recording. Therefore, we filter out the signals with less than 8 seconds of recording.

(3) Signal filtering. We remove frequency noise from the signals as described in Section A.

(4) Sliding window. We use a rolling window algorithm to create our signal windows. As previously described, we use segments of 8 seconds with a stride of 2 seconds.

(5) SQI Filter. We filter out noisy segments with an SQI score. As described in Section A., we keep the windows with an SQI between $0.35$ and $0.8$.

(6) Add blood pressure. Using the windows from the ABP signal, we extract the maximum and minimum values of the input window. The maximum value is used as the SBP target and the minimum value is used as the DBP target. The ABP signal is rejected after this step.

(7) Filter pressure. With the BP target values, we filter our out-of-bounds pressure as described in Section A.

(8) Standard scaling. To make sure the extracted features focus on the shape of the waveform and to generalize to multiple PPG devices, we apply standard scaling to each input window.

### C. Heartbeat

Extracting individual heartbeats from the signals is also one of the most commonly used input modalities in BP estimation. This method is often employed for manual feature extraction of the PPG waveform. In our work, we use this input modality as an automatic feature extraction strategy, letting the machine learning model extract its latent features from the input heartbeat. This method is unique to BP estimation but bears some resemblance to tokenization of text data. The heartbeat segments can interpreted as embeddings which make up the input sequence that is the PPG signal. This method bears a few disadvantages. Notably, the temporal aspect of the PPG waveform is mostly lost. This method is also harder to implement since finding the boundaries between heartbeats can be technically challenging. Luckily, Elgendi et al. [14] described an algorithm for systolic peak extraction from PPG. An implementation is freely available in the NeuroKit2 [35] python library.

We implement the heartbeat input modality to make it easier to compare our results with other papers and to determine whether the waveform contains sufficient information for BP estimation. We preprocess with the following steps:

(1) Rejection of invalid recordings. This step is executed as described in Section A.

(2) Signal filtering. We remove frequency noise from the signals as described in Section A.

(3) Split heartbeats. To split the signals into segments containing a single heartbeat, we use Elgendi et al.'s [14] algorithm. The algorithm finds the location of each systolic peak. We then find the minimal values between two consecutive peaks and use these points as the boundaries for heartbeat segments. Finally, we resample the segments to 400 samples to obtain inputs with constant size.

(4) SQI Filter. We filter out noisy segments with an SQI score. As described in Section A., we keep the heartbeats with an SQI between 0.5 and 2.

(5) Add blood pressure. As with input windows, we extract the maximum and minimum values of the heartbeat segment. The maximum value is used as the SBP target and the minimum value is used as the DBP target. The ABP signal is rejected after this step.

(6) Filter pressure. With the BP target values, we filter our out-of-bounds pressure as described in Section A.

(7) Standard scaling. To make sure the extracted features focus on the shape of the waveform and to generalize to multiple PPG devices, we apply standard scaling to each heartbeat segment.

### D. Heartbeat Sequence

The heartbeat sequence consists of stacking individual heartbeats into a new axis, resulting in a two-dimensional input where the first axis indexes the heartbeats and the second axis indexes the signal. The objective of this modality is to restore the temporal aspect of the PPG waveform that was present in rolling windows and lost in single heartbeats. This modality also allows us to keep fine-grained analysis on a single heartbeat. As mentioned in C. the heartbeats can be viewed as embeddings for the time series. In this context, the semantics of the time steps change compared to how the PPG signal is generally interpreted in the literature. Individual time steps are usually considered to be the samples from the sensor. With this heartbeat sequence, the time steps are the heartbeats. We believe this is closer to how experts generally interpret the PPG signal. We use a

series of 10 heartbeats as it approximates the number of heartbeats in input windows while keeping the input at a reasonable size.

We implement a heartbeat sequence to determine if fine-grained analysis of the signal and analysis of its temporal aspect can be exploited at the same time. The preprocessing steps are as follows:

(1) Rejection of invalid recordings. This step is executed as described in Section A.

(2) Signal filtering. We remove frequency noise from the signals as described in Section A.

(3) Split heartbeats. To split the signals into segments containing a single heartbeat, we use Elgendi et al.'s [14] algorithm. The algorithm finds the location of each systolic peak. We then find the minimal values between two consecutive peaks and use these points as the boundaries for heartbeat segments. Finally, we resample those segments to 50 samples to obtain inputs with constant size.

(4) Sliding window. We use a rolling window algorithm to create our stack of heartbeats. We stack together 10 heartbeats with a stride of 1 heartbeat.

(5) SQI Filter. We filter out noisy segments with an SQI score. As described in Section A., we keep the sequence where all heartbeats' SQI are within 0.5 and 2.

(6) Add blood pressure. We extract the maximum and minimum values of the last heartbeat in the sequence. The maximum value is used as the SBP target and the minimum value is used as the DBP target. The ABP signal is rejected after this step.

(7) Filter pressure. With the BP target values, we filter our out-of-bounds pressure as described in Section A.

(8) Standard scaling. To make sure the extracted features focus on the shape of the waveform and to generalize to multiple PPG devices, we apply standard scaling to each heartbeat in the sequence.

### 3.4.3 Surveyed Model Reproductions

We implemented the surveyed models (Section 3.3) to the best of our ability. However, some implementation details required for the reproduction of the models are missing from their descriptions. In these cases, we made our best guesses to maintain the architecture to the spirit of their articles. We detail our modifications in Section 3.4.3. In addition to the models surveyed, we present various models using different layer types to evaluate which performs the best at feature extraction considering the different input modalities. (1) A simple MLP composed of fully connected layers, (2) A ResNet-inspired model using residual convolution layers, (3) a model stacking an RNN with an MLP for regression, and (4) a transformer encoder model using self-attention layers.

For the model proposed by Slapničar et al. [62], we implemented the version that only used the raw PPG signal as described in the article. The model introduced by Athaya and Choi [4] required some modifications. We added conditional zero padding of 1 at the expansion layers and employed a valid padding for the convolution layer when the shape of the lower level's input and the skip contraction block input did not match. This modification was necessary since a supplemental row of data is lost during the contraction layers when the input size is odd. Otherwise, we used a same padding for the convolution layer. We also removed the last two convolution layers of the model and replaced them with a Dense layer of output 2 in order to adapt the model to our regression problem. We also added a Dense layer of output 2 to the models proposed by Harfiya et al. [20] and Kim et al. [26] for the same reason.

The model introduced by Chen et al. [7] had some ambiguities with its hyperparameters and architecture. We used our best guesses to resolve these ambiguities and fill in the blanks. According to the description, the RFPASB block includes a skip connection, but its implementation is not possible as described. Due to the change in number of filters of the convolution layers from the input to the output, the dimensions of the input and output do not match and can not be added together. We added a convolution layer with a kernel of 1, strides of 2, same padding, and matching filters to implement the skip connection. In the hyperparameter table, the article states that the RFPASB modules use a stride of 2, but the specific convolution layer in the module where it is applied is not stated. We assumed it was added to the last $1 \times 1$ convolution layer in the multiscale large receptive

field. We also removed the global average pooling layer before the channel attention and spatial attention module since it would cause the feature map to output a 2D tensor while the subsequent layers require a 3D tensor. Moreover, The first MLP layer of the channel attention module uses a reduction ratio hyperparameter to determine the number of units as described by Woo et al. [72]. This hyperparameter is not specified, so we set it to 1. The last change concerns the soft threshold module. The given definition in the article is not well-defined. We changed it to the equation shown in Equation 1 to match the one given by Zhao et al. [83],

$$y_{soft} = \begin{cases} x - \delta & x > \delta \\ 0 & -\delta \leq x \leq \delta \\ x + \delta & x < -\delta \end{cases} \tag{1}$$

where $x$ is the feature map from the multiscale large receptive field and $\delta$ is the threshold from the mixed domain attention module.

Schrumpf et al. [57] used both AlexNet and ResNet for their work. The specific version of ResNet wasn't specified in the article. For simplicity, we chose to implement ResNet34. AlexNet and ResNet34 both use a final dense layer of 1000 units to output their prediction, again, we replaced them with dense layers of 2 units to adapt them to our regression problem.

### 3.4.4 Our Models

Several deep learning architectures have been evaluated for BP estimation in the past with varying degrees of success. Most commonly, the architectures usually rely on Multilayer Perceptrons (MLP), Recursive Neural Networks (RNN), Convolutional Neural Networks (CNN), or self-attention layers to perform feature extraction. Although those architectures have been explored, the difficulty in comparing results due to differences in datasets, preprocessing, and data splits makes it difficult to evaluate which performs the best at feature extraction. While MLP is the simplest model that can be applied to our dataset, RNN, CNN, and self-attention layers have all been applied to time series data and therefore can naturally be extended to PPG data.

## A. MLP

The MLP is the simplest model we evaluate. It consists solely of regular dense layers with activation functions. Although this model can prove efficient at extracting features, it could prove incapable of extracting long-range dependencies in the data due to the large number of parameters required. We ran a hyperparameter search on the architecture to find the architecture that performed the best. The architecture consists of 2 layers of 128 neurons with a ReLU activation for feature extraction and a final layer of 2 neurons with no activation for regression. The architecture can be viewed in Figure 3.6.

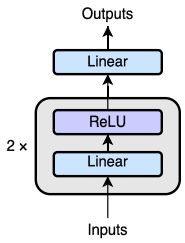



Figure 3.6: MLP architecture

## B. RNN-MLP

RNNs are a type of neural network specifically designed to deal with temporal data. It is natural to apply RNNs to PPG signals, which are a type of time series. Therefore, RNNs should be well capable of extracting information from the temporal aspect of the PPG waveform. RNNs have also been explored for BP estimation, but usually don't perform as well as convolution layers for feature extraction. Fairly often, RNNs are used in conjunction with CNNs. The CNN extracts fine-grained features from the waveform, while the RNN extracts its temporal aspect. In order to evaluate the RNN's performance as a feature-extracting layer, we use it without convolutional layers. We stacked our MLP (Section A.) at the end of the RNN to perform regression. As with our other models, we used a hyperparameter search to find the optimal parameters for the RNN. We found that 10 layers of GRU with 256 neurons performed the best. The architecture can be viewed in Figure 3.7.

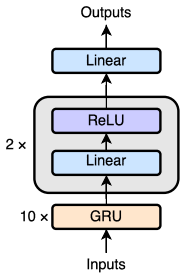



Figure 3.7: RNN-MLP architecture

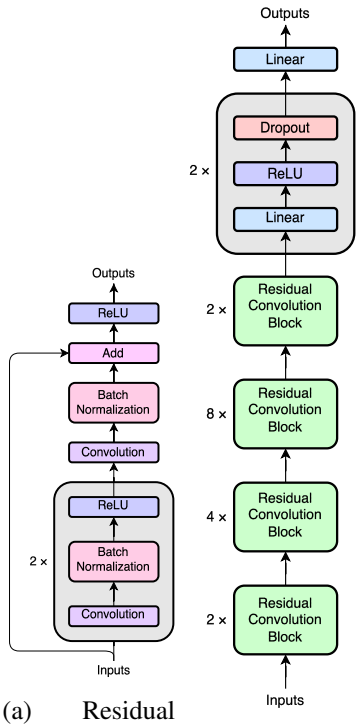
### C. Residual CNN

Convolution layers have often been used on PPG signals for automatic feature extraction. The restricted size of the kernel in convolution layers should encourage the model to extract local features from the signal while extracting the temporal aspect of the input data. Due to its popularity, a ResNet-like architecture is often used in place of regular convolutional layers. The ResNet architecture relies on regular convolution layers with residual connections to improve the optimization of very deep networks. Since those ResNet architectures usually perform best, our architecture also relies on convolutional residual layers. We ran a hyperparameter search on the architecture to find the architecture that performed the best. The optimal architecture was composed of 4 residual modules and a final regressor. The residual modules were composed of several residual blocks. The residual blocks were composed of 3 convolutional layers and a skip connection. All convolution layers used a kernel of size 3, same padding, and a ReLU activation function. The number of dimensions of the kernel changed depending on the input used. For window and heartbeat inputs we used 1-dimensional convolution layers and 2-dimensional convolutional layers for heartbeat sequence inputs. The convolution layers of the first residual blocks of each residual module used a stride of 2 to reduce the dimension of its output. The skip connection going around it then used a convolution layer with a kernel size of 1, stride 2, and no activation function to make the output sizes match. All convolution layers are followed by a batch normalization layer. The number of residual blocks and filters used in each module is shown in Table 3.4. The regressor was composed of 2 fully connected layers of 128 units with ReLU activation. A dropout layer with a rate of $0.01$ was added after each of the regressor layers for regularization. The first layer of the regressor was regularized with an L2 weight regularizer. Finally, a last fully connected layer of 2 units and no

activation function was added to provide the final regression output. The architecture can be viewed in Figure 3.8.

Table 3.4: Number of residual blocks and filters used in each residual module

| Module | Blocks | Filters |
|---|---|---|
| 1st | 2 | 64 |
| 2nd | 4 | 128 |
| 3rd | 8 | 256 |
| 4th | 2 | 256 |




(a) Residual Convolution Block (b) Residual CNN

Figure 3.8: Residual CNN architecture

## D. Transformer Encoder

In [69] ("Attention is all you need"), the authors present the Transformer architecture for machine translation. In this architecture, the model only relies on self-attention layers to extract features from its input data. The authors make the argument that self-attention layers can learn long-range dependencies while keeping a connection between all elements in the sequence. They also

demonstrate that self-attention layers find long-range dependencies in an input while keeping computations parallelizable, thereby providing advantages from both recursive and convolutional layers in a single-layer architecture. Considering these advantages, self-attention layers might perform well for feature extraction in a PPG signal. Although attention layers have been applied to BP estimation, very little research has been done on self-attention layers. From our survey, we only found 3 articles that employed self-attention [15, 26, 34]. Therefore, in our research, we reproduce the transformer encoder architecture and tune its hyperparameters to evaluate the performance of self-attention layers for automatic feature extraction of PPG signals. Again, we found optimal hyperparameters through a hyperparameter search. The optimal architecture consists of 3 encoder modules. The encoder modules are composed of one global self-attention layer with 4 attention heads and 2 fully connected layers to provide non-linearity. We used an embedding size equal to the size of the last dimension of the input. The embedding size was $1$ for the window and heartbeat input modalities and $50$ for heartbeat sequences. A ReLU activation function is applied after the first fully connected layer. The first fully connected layer uses $64$ neurons while the second's neuron count is equal to the embedding size. A skip connection is added around the self-attention layer and the two fully-connected layers. Layer normalization is applied after both skip connections. As with the RNN, we stack our MLP architecture on top of the encoder to perform regression. The architecture can be viewed in Figure 3.9.

### 3.4.5 Learning Curve Analysis

In order to evaluate the impact of dataset size and variability of patients on the performance, we performed a learning curve analysis on the dataset. The analysis was accomplished by sub-sampling the patients in the training set and keeping the validation and training set identical. We then trained our simplest model, the MLP, on the training set and evaluated it on the test set. We chose the heartbeat input modality for these experiments as it is the input modality that trains the fastest. We sampled the patients at increments of 10% from 10% to 100%, yielding the following sampling rates: 10%, 20%, 30%, ..., 100%. Each of those sampling rates was repeated 10 times and then averaged together to get an accurate performance estimate despite the random sampling.

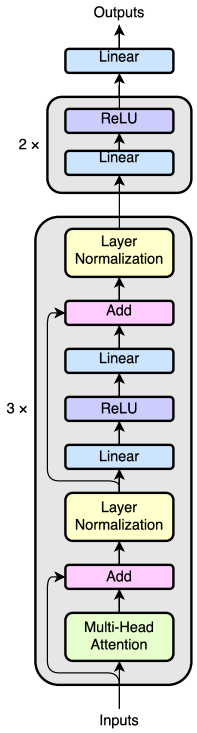



Figure 3.9: Transformer Encoder architecture

### 3.4.6 Training Settings

The models were implemented in Python 3.10 using TensorFlow 2.10 [1] and the Keras API [8]. The implementation for the models is made freely available at https://github.com/FGRCL/ML-BP-Estimation.

All models were trained under the same optimization parameters and kept to fairly common values. We trained all models for $50$ epochs with a batch size of $128$. Upon the end of training, we recovered the version of the model that performed the best on the validation set and evaluated this version on the test set. This method is used in place of early stopping. We further made sure that all models reached convergence during training. We opted for the Adam optimizer with a learning rate of $.0001$ and used mean absolute error as the loss function.

We ran our experiments on Compute Canada to run multiple experiments at the same time. The hardware for the experiments varied. We used 32 CPU cores and 1 GPU. The possible list of CPUs is given in Table 3.5. The GPUs used were NVIDIA P100 Pascal 12G, NVIDIA P100 Pascal 16G, and NVIDIA V100 Volta 32G.

Table 3.5: List of CPUs used to run experiments

| CPU |
|---|
| Intel E5-2683 v4 Broadwell @ 2.1GHz |
| Intel E5-2683 v4 Broadwell @ 2.1GHz |
| Intel E5-2683 v4 Broadwell @ 2.1GHz |
| Intel E5-2683 v4 Broadwell @ 2.1GHz |
| Intel E7-4809 v4 Broadwell @ 2.1GHz |
| Intel Silver 4216 Cascade Lake @ 2.1GHz |
| Intel Platinum 8160F Skylake @ 2.1GHz |
| Intel Platinum 8260 Cascade Lake @ 2.4GHz |

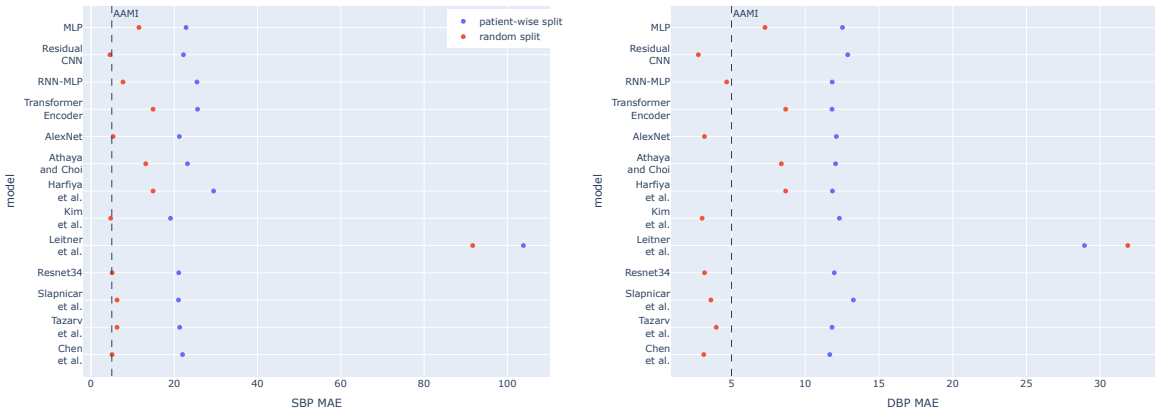
## 3.5 Results

We compare the results of our experiments across our different preprocessing methods, surveyed models, developed models, and dataset split. We measure the performance of our models in MAE and its SD for SBP and DBP. The performance of all tested models as evaluated on the test set can be viewed in Tables 3.6, 3.7, and 3.8. Table 3.6 shows the results using window preprocessing, Table 3.7 shows the results using heartbeat preprocessing, Table 3.8 shows the results using heartbeat sequence preprocessing. Each table is split into two sections, the first one shows our results with a patient-wise split, and the second shows our results using a random split. The tables also contain both our models and our reproduction of models from the literature as selected in our survey in Section 3.3. Our models are boldfaced and reproduced models are marked with a citation. We would like to emphasize that the results on models from the literature are from our experiments and not the results reported in their respective articles. Since reproduced models were incompatible with heartbeat sequence preprocessing, they are omitted from 3.8. Figure 3.14 presents the results of our learning curve analysis experiments. It displays the performance of the MLP on the test set with respect to the percentage of patients randomly sampled to be part of the training set. We measured the same performance metrics, i.e., MAE for SBP and DBP.

We note the large difference in performance between the patient-wise split and random split. For ease of comparison, this difference is shown in Figures 3.10, 3.11, and 3.12. On average, the random split provided an improvement of $13.802$ mmHg SBP MAE and $7.277$ mmHg DBP MAE. In context, the range of normal SBP spans $30$ mmHg, from $90$ mmHg to $120$ mmHg, and the range for DBP spans $20$ mmHg, from $60$ mmHg to $80$ mmHg. This means an MAE of $13.802$ mmHg SBP

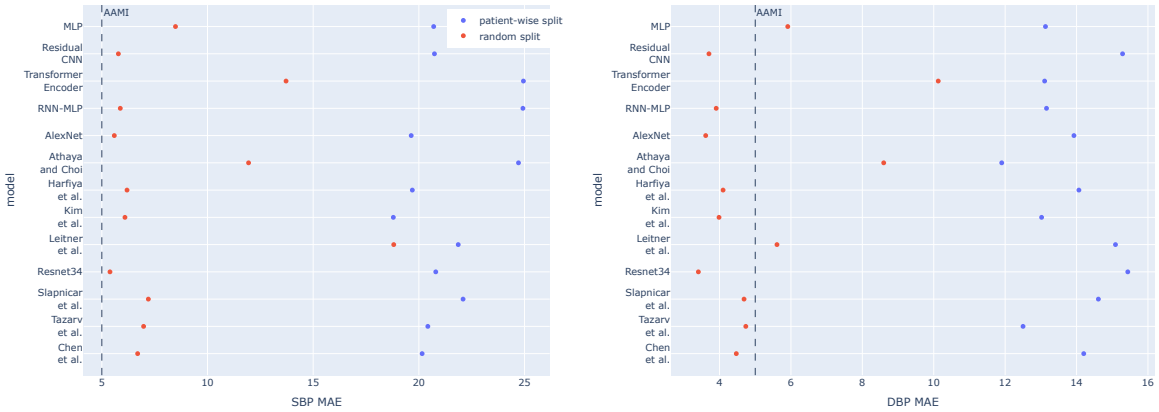
Table 3.6: Performance results of all models and splits with the window input modality

| | SBP (mmHg) | | DBP (mmHg) | |
|---|---|---|---|---|
| Model | MAE | SD | MAE | SD |
| | Patient-wise split | | | |
| Slapničar et al. [62] | 21.013 | 14.158 | 13.263 | 10.742 |
| Athaya and Choi [4] | 23.169 | 17.47 | 12.056 | 10.384 |
| Chen et al. [7] | 21.986 | 15.011 | **11.662** | **10.292** |
| Harfiya et al. [20] | 29.447 | 18.027 | 11.843 | 9.597 |
| Kim et al. [26] | **19.1** | **13.889** | 12.313 | 10.186 |
| Leitner et al. [30] | 103.847 | 20.44 | 28.935 | 14.732 |
| Tazarv and Levorato [66] | 21.301 | 16.772 | 11.815 | 9.677 |
| AlexNet [57] | 21.222 | 15.017 | 12.11 | 10.452 |
| Resnet34 [57] | 21.078 | 15.297 | 11.968 | 10.47 |
| **MLP** | 22.83 | 17.164 | 12.525 | 10.879 |
| **RNN-MLP** | 25.447 | 17.252 | 11.827 | 9.573 |
| **Residual CNN** | 22.205 | 15.56 | 12.88 | 10.367 |
| **Transformer Encoder** | 25.59 | 17.164 | 12.525 | 10.879 |
| | Random split | | | |
| Slapničar et al. [62] | 6.26 | 7.621 | 3.601 | 5.711 |
| Athaya and Choi [4] | 13.146 | 12.162 | 8.379 | 7.58 |
| Chen et al. [7] | 5.051 | 7.232 | 3.117 | 5.646 |
| Harfiya et al. [20] | 14.908 | 12.387 | 8.669 | 7.66 |
| Kim et al. [26] | 4.725 | 6.746 | 3.005 | 5.57 |
| Leitner et al. [30] | 91.673 | 21.105 | 31.872 | 13.742 |
| Tazarv and Levorato [66] | 6.255 | 8.155 | 3.964 | 6.21 |
| AlexNet [57] | 5.303 | 7.043 | 3.164 | 5.646 |
| Resnet34 [57] | 5.077 | 7.194 | 3.172 | 5.696 |
| **MLP** | 11.52 | 11.125 | 7.269 | 7.604 |
| **RNN-MLP** | 7.685 | 8.512 | 4.668 | 6.313 |
| **Residual CNN** | **4.56** | **6.452** | **2.752** | **5.377** |
| **Transformer Encoder** | 14.908 | 12.413 | 8.672 | 7.665 |



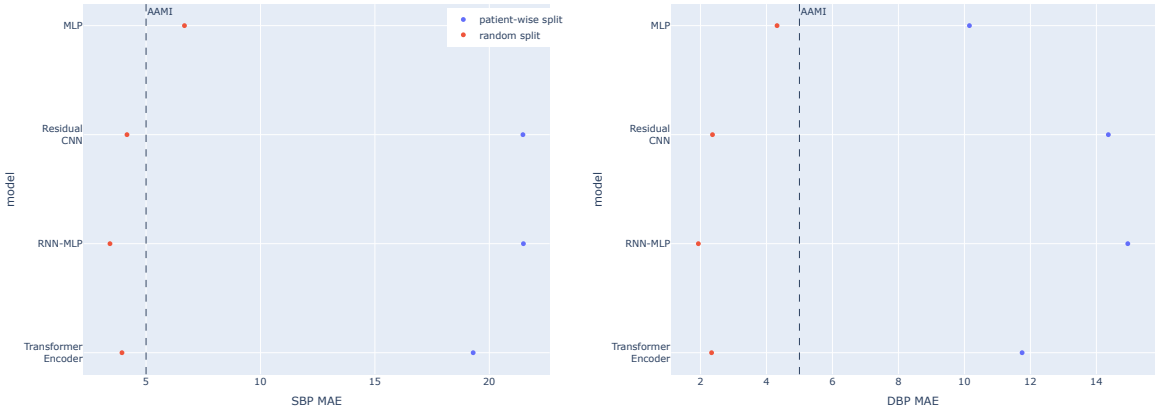

(a) SBP MAE (b) DBP MAE

Figure 3.10: Comparison of patient and random split with window preprocessing




(a) SBP MAE (b) DBP MAE

Figure 3.11: Comparison of patient and random split with heartbeat preprocessing




(a) SBP MAE (b) DBP MAE

Figure 3.12: Comparison of patient and random split with heartbeat sequence preprocessing

Table 3.7: Performance results of all models and splits with the heartbeat input modality

| | SBP (mmHg) | | DBP (mmHg) | |
|---|---|---|---|---|
| Model | MAE | SD | MAE | SD |
| | Patient-wise split | | | |
| Slapničar et al. [62] | 22.092 | 15.087 | 14.611 | 10.463 |
| Athaya and Choi [4] | 24.715 | 16.206 | **11.904** | **10.043** |
| Chen et al. [7] | 20.155 | 14.138 | 14.201 | 9.927 |
| Harfiya et al. [20] | 19.691 | 14.351 | 14.065 | 10.42 |
| Kim et al. [26] | **18.792** | **14.353** | 13.021 | 9.982 |
| Leitner et al. [30] | 21.863 | 16.69 | 15.09 | 10.422 |
| Tazarv and Levorato [66] | 20.425 | 15.859 | 12.501 | 9.735 |
| AlexNet [57] | 19.638 | 14.757 | 13.927 | 10.331 |
| Resnet34 [57] | 20.8 | 14.176 | 15.435 | 10.226 |
| **MLP** | 20.701 | 15.836 | 13.13 | 9.836 |
| **RNN-MLP** | 24.924 | 14.618 | 13.158 | 10.394 |
| **Residual CNN** | 20.738 | 14.565 | 15.289 | 10.632 |
| **Transformer Encoder** | 24.948 | 14.621 | 13.106 | 10.333 |
| | Random split | | | |
| Slapničar et al. [62] | 7.201 | 8.034 | 4.686 | 7.1 |
| Athaya and Choi [4] | 11.943 | 10.861 | 8.599 | 8.666 |
| Chen et al. [7] | 6.694 | 7.973 | 4.469 | 7.134 |
| Harfiya et al. [20] | 6.192 | 7.654 | 4.099 | 6.965 |
| Kim et al. [26] | 6.094 | 7.577 | 3.986 | 6.685 |
| Leitner et al. [30] | 18.812 | 25.476 | 5.609 | 8.218 |
| Tazarv and Levorato [66] | 6.977 | 8.313 | 4.736 | 7.418 |
| AlexNet [57] | 5.591 | 7.338 | 3.613 | 6.607 |
| Resnet34 [57] | **5.385** | **7.391** | **3.407** | **6.368** |
| **MLP** | 8.485 | 8.984 | 5.911 | 7.624 |
| **RNN-MLP** | 5.873 | 7.419 | 3.907 | 6.666 |
| **Residual CNN** | 5.783 | 7.331 | 3.704 | 6.446 |
| **Transformer Encoder** | 13.716 | 11.164 | 10.126 | 9.007 |

Table 3.8: Performance results of all models and splits with the heartbeat sequence input modality

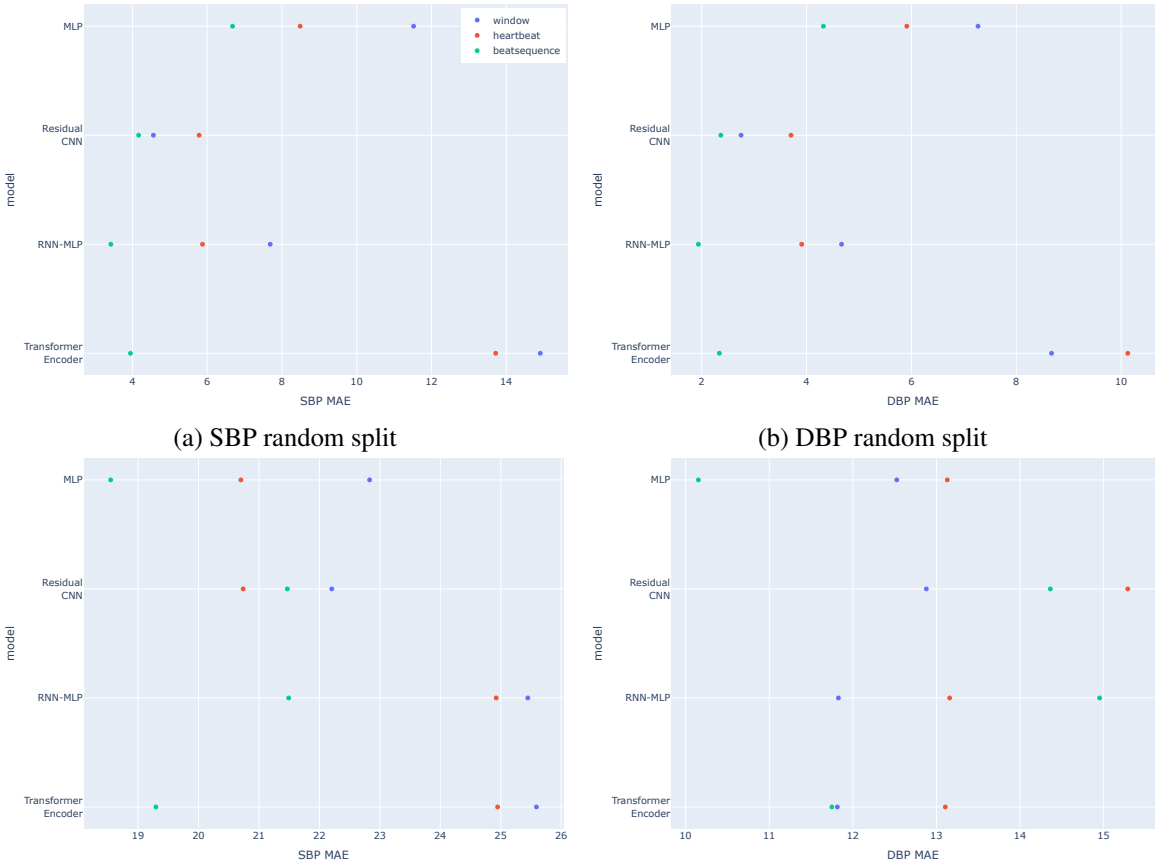
| | SBP (mmHg) | | DBP (mmHg) | |
|---|---|---|---|---|
| Model | MAE | SD | MAE | SD |
| | Patient-wise split | | | |
| **MLP** | **18.547** | **14.054** | **10.152** | **6.986** |
| **RNN-MLP** | 21.491 | 16.35 | 14.952 | 9.648 |
| **Residual CNN** | 21.469 | 14.545 | 14.362 | 8.266 |
| **Transformer Encoder** | 19.294 | 14.459 | 11.749 | 8.054 |
| | Random split | | | |
| **MLP** | 6.678 | 7.079 | 4.321 | 5.448 |
| **RNN-MLP** | **3.423** | **4.344** | **1.937** | **3.278** |
| **Residual CNN** | 4.166 | 5.382 | 2.365 | 4.271 |
| **Transformer Encoder** | 3.947 | 4.644 | 2.337 | 3.799 |

could cause a normal BP to incorrectly be measured as hypertension. Furthermore, while the random split provides adequate results, it can not be generalized to new patients. Algorithms evaluated with this approach do not provide an accurate estimation of the performance of a calibration-free algorithm. Instead, the patient-wise split provides a much more accurate estimate of performance for a calibration-free algorithm. In the literature, the dataset split used is often not explicitly mentioned. Our results can help us estimate which dataset split was used. Excluding the Slapničar, AlexNet, and ResNet models, we can note that the results from the random split are closer to the results reported in their respective articles. Hence, we believe it is a safe assumption that a random split is generally used to report the performance of machine-learning algorithms for BP estimation. Due to differences in the dataset and mostly preprocessing of it, our results do not match exactly the ones reported in the literature.

We can also make observations from the performance of the models with respect to their complexity. While our models generally use fewer parameters than the ones from the literature, their performance difference is small.

We observe the models which perform the best. Taking only into account the patient-wise split results that are clinically applicable, the model with the highest-rated performance is the MLP with heartbeat sequence preprocessing. The MLP is also the simplest of all the models tested. When comparing all models, irrespective of dataset split, the RNN-MLP using heartbeat sequence preprocessing with a random split performed the best.

We note the performance relative to input modalities. For ease of comparison, the performance of our models is shown across their preprocessing strategy in Figure3.13. Under this comparison, the heartbeat sequence preprocessing strategy generally performs the best, although it is sometimes bested by the other two strategies.




(a) SBP random split

(b) DBP random split

(c) SBP patient split

(d) DBP patient split

Figure 3.13: Comparison of patient and random split with heartbeat sequence preprocessing

We further notice the performance difference between the training set and our evaluation set. A comparison of the training set and test set for the performance in MAE for SBP and DBP is shown in Table 3.9. For a better overview, we calculated the mean of those metrics for all the models and input modalities. Most Notably we note the great gap in performance between the two sets for the patient-wise split. The performance gap is $14.609$ mmHg SBP MAE and $8.016$ mmHg DBP MAE. This gap is very large when compared to the same difference measures, $0.180$ mmHg SBP MAE and $0.156$ mmHg DBP MAE, obtained on the random split.

Table 3.9: Mean performance results on the training set and test set

| | SBP MAE (mmHg) | | | DBP MAE (mmHg) | | |
|---|---|---|---|---|---|---|
| Split | Training | Test | Diff. | Training | Test | Diff. |
| Random | 10.421 | 10.601 | 0.180 | 5.514 | 5.67 | 0.156 |
| Patient | 10.008 | 24.617 | 14.609 | 5.506 | 13.522 | 8.016 |

Under patient-wise dataset split, no model meets the requirements for the AAMI standard for SBP or DBP.

From the learning curve analysis results shown in Figure3.14, we notice that the performance for SBP MAE improves the larger the dataset is, but varies little with the other metrics.


(a) SBP MAE

(b) SBP SD

(c) DBP MAE

(d) DBP SD

Figure 3.14: Performance of the MLP in relation to the percentage of patients used for training with heartbeat input modality

## 3.6 Discussion

In this work, we set out to determine if machine learning could be used to develop a non-invasive, continuous, and calibration-free for BP monitoring. We reduced our scope to models which used a single PPG sensor, to minimize the cost of a potential system and maximize its ease of use. We further scoped down to deep learning algorithms since they performed better at feature extraction. More restrictions were detailed in the survey section (Section 3.3). We draw two main conclusions from our findings; Non-invasive, calibration-free BP monitoring remains an unsolved problem and the solution to this problem lies in the data rather than the model.

A non-invasive, calibration-free algorithm using a single PPG site that is applicable in a clinical setting has yet to be presented in the literature. As noted in the results section (Section 3.5), a patient-wise dataset split provides a more accurate estimate of performance for a calibration-free algorithm. A random split, the most common alternative, leads to sharing the patients between the training set and the test set. As stated by Xing and Sun [77], The largest challenge in estimating BP from PPG signal is learning the physiology of patients. As such, this sharing of patients leads to data leakage, which overestimates the performance of the algorithm in a calibration-free context. The employed dataset split strategy is not always mentioned in the literature. From our results, in Table 3.6, 3.7, and 3.8 we found that results from a random split were close to the performance reported in their respective articles. These results indicate that a random split is the most popular used split strategy in the literature, which is inapplicable to a calibration-free algorithm. The dataset splits from our surveyed models are reported in Table 3.3. Except for, Slapničar et al. [62] and Schrumpf et al. [57], we can determine that all surveyed models used a dataset split which shared patients between their training and test split. Slapničar et al. [62] further note the difficulty in comparing works in the literature due to variations in the datasets and evaluation metrics. Moreover, works in the literature rarely share their source code, making it more difficult to compare. Our work in reproducing models from the literature and comparing them solves these issues and allows for a more accurate comparison of approaches. We noticed from our patient-wise results that no models were able to meet the AAMI standards for BP monitoring. Therefore, this leads us to the conclusion

that no works in the literature have been able to present an algorithm for non-invasive, calibration-free BP monitoring.

We found that improvements in data preprocessing and quality are more effective than increasing the complexity of models. Most often, works in the literature propose a new deep learning architecture to provide BP monitoring. In contrast, we found that adding complexity to models seldom increases the performance of models by much. For instance, although the model by Kim et al. [26] usually performs the best for a patient-wise split and is also the most complex of our surveyed models, its performance improvement isn't very large compared to our simplest model, an MLP. Table 3.10 shows the improvement provided by the Kim et al. [26] compared to our MLP.

Table 3.10: Performance improvement of MAE going from an MLP to the Kim et al. [26] model

| | SBP MAE (mmHg) | | | DBP MAE (mmHg) | | |
|---|---|---|---|---|---|---|
| Preprocessing | MLP | Kim et al. [26] | Imp. | MLP | Kim et al. [26] | Imp. |
| Window | 22.83 | 19.1 | 3.73 | 12.525 | 12.313 | 0.212 |
| Heartbeat | 18.792 | 20.701 | 1.909 | 13.13 | 13.021 | 0.109 |

The performance difference ranges from 3.73 mmHg to 0.109 mmHg. The Kim et al. [26] model has 5.5B parameters, while the MLP has 145K parameters. In our opinion, this performance difference is too little to warrant the added complexity. Next, when comparing results across all our models, we note that the heartbeat sequence preprocessing performs the best. The RNN-MLP on a random split performs the best out of all our experiments. These observations lead us to the conclusion that improvements to the data preprocessing and cleaning are more efficient at improving performance than increasing the model complexity. We also noted that, while we let all models achieve convergence on the training set, there was a large gap between test error and training error on the patient-wise split dataset, this is usually an indication of overfitting. This gap can be seen in Table 3.9. Overfitting can generally be solved by increasing model capacity, increasing regularization, increasing the amount of data, or a combination of these. Still, considering our wide variety of models in architecture and complexity, the remaining course of action to narrow down this gap and

reduce training error to an acceptable level is to increase the amount of data. As seen in our learning curve analysis (Figure 3.14), SBP MAE can benefit from more data. We believe that the large performance gap between DBP MAE and SBP MAE causes the model to optimize entirely on the SBP output and mostly ignore the DBP output. This explains why DBP MAE did not see as much of a performance improvement from additional data. Following the findings of Brumen et al. [6], by fitting a power model on the SBP MAE curve, we can extrapolate the amount of data required to achieve an algorithm with less than $5$ mmHg SBP MAE. This simple extrapolation evaluated that roughly $21.51\times$ more data is required to meet this goal.

This is described in Equation 2 and 3.15 shows the fitted curve plotted against the gathered data.

$$y = 28.533 - 9.518x^{0.295} \tag{2}$$

where $y$ denotes the expected MAE for SBP and $x$ denotes the sample rate.

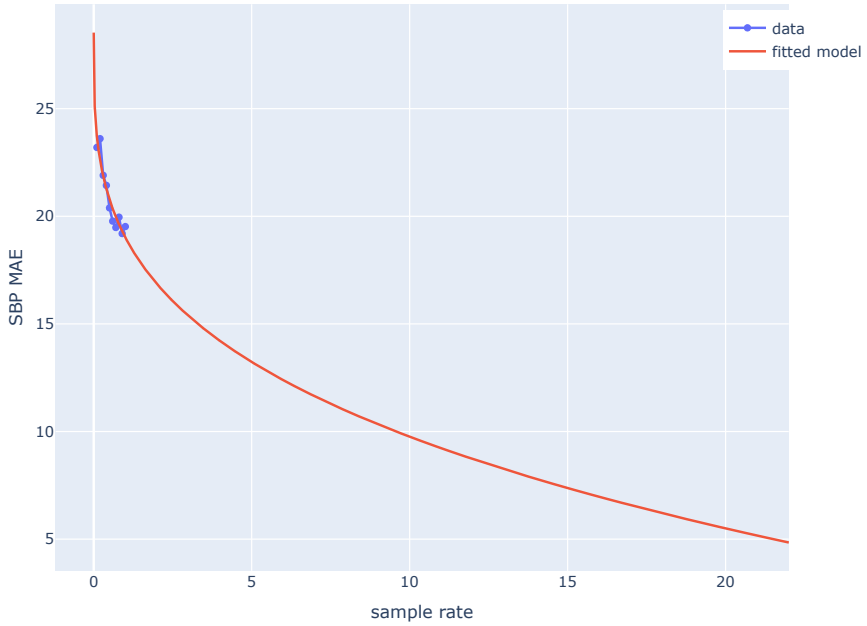



Figure 3.15: Fitted curved plotted against the data

With the entire training dataset currently consisting of 42 patients, this would mean that a training set of roughly 904 patients is required. In addition to low amounts of data, the variability of data is also usually poor. Works on this problem usually rely on 4 datasets: MIMIC, UQVSD, UCI, or self-collected. These datasets usually provide fairly low variability of data since all the data is

collected from a single hospital, and usually from patients under similar conditions. Therefore, we suggest that creating a larger dataset with much more variability is the most important task in order to solve non-invasive, calibration-free BP monitoring.

## 3.7 Conclusion

Our results seem to indicate that more data and better preprocessing have more to offer than increasing model complexity to attain acceptable performance for non-invasive, calibration-free BP monitoring from a single PPG signal that could be applicable in a clinical setting. Still, non-invasive, calibration-free BP monitoring is highly desirable to avoid the shortcomings of cuff-based and invasive BP monitoring. In light of our claims, we envision a few potential avenues to achieve this algorithm. The scope of this article was narrowed down to articles using a single PPG signal. This scope could be broadened to explore additional signals, such as ECG, or use demographic information. Additionally, we believe it is valuable to gather more data in order to validate our extrapolation (Equation 2) and increase model accuracy. Finally, We alluded to the low variability of current datasets. Increasing the variability of patients who compose the datasets could potentially improve the quality of data enough to achieve our goal.

# Chapter 4

# Conclusion and Future Work

## 4.1 Conclusion

According to our results, the state-of-the-art for BP estimation is unable to provide a non-invasive, calibration-free BP estimation algorithm. Our extensive survey of the literature helps provide a comprehensive overview of the literature BP estimation. By reproducing models from the literature, we were able to better compare and reevaluate these models using an appropriate split for a calibration-free algorithm. We showed that using a patient-wise split is essential to accurately estimate the performance of calibration-free algorithms. We also found that a majority of works in this domain do not use a patient-wise split. Furthermore, our results seem to indicate that increasing the complexity of deep learning networks does not provide considerable performance improvement over simpler models. Similarly, we estimated the impact of increasing the amount of patients in the dataset on the performance of the model. We determined that no amount of data could improve the performance of our model to an acceptable level.

Opposing these shortcomings, we also found that some of our input modalities and proposed models perform better than the literature. Our heartbeat sequence preprocessing methodology outperformed windows of signal which are commonly used in the literature. Moreover, our RNN-MLP using this preprocessing also outperformed the models from the literature. These two methods could provide performance improvements for machine learning models using a personalization approach while simplifying the models.

## 4.2 Future Work

While our results seem to show little hope for a non-invasive, calibration-free BP estimation algorithm, we note that only a small scope of the possible approaches have been explored.

First, our work focused solely on methods using a single PPG signal. As shown with PTT and PAT methods, adding additional signals to the algorithm might improve the performance. A first possibility could be to use two PPG signals with a deep learning network to approximate PTT approaches. On a similar note, deep learning architectures might be able to find additional relationships between PPG and ECG signals that PAT approaches miss. Renouncing additional physiological signals, we've noted that the literature shows that adding demographic information as inputs can improve performance. Deep learning approaches might be able to extrapolate enough morphologies from these features to achieve a calibration-free approach.

We've extrapolated that adding additional patients to the dataset can help achieve acceptable performance, for that reason, we believe that improvements could be made to the datasets used for BP estimation. First, it could be valuable to train on more data to validate our extrapolation and improve performance. Furthermore, our experiments were conducted only on the MIMIC dataset. This dataset is composed only of ICU patients from a single hospital. We believe that adding more variety to the dataset could improve the performance of our algorithm.

# Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Nicolas Aguirre, Edith Grall-Maës, Leandro J. Cymberknop, and Ricardo L. Armentano. Blood Pressure Morphology Assessment from Photoplethysmogram and Demographic Information Using Deep Learning with Attention Mechanism. *Sensors (Basel, Switzerland)*, 21(6):2167, March 2021.

[3] John Allen. Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, 28(3):R1, February 2007.

[4] Tasbiraha Athaya and Sunwoong Choi. An Estimation Method of Continuous Non-Invasive Arterial Blood Pressure Waveform Using Photoplethysmography: A U-Net Architecture-Based Approach. *Sensors (Basel, Switzerland)*, 21(5):1867, March 2021.

[5] Eoin Brophy, Maarten De Vos, Geraldine Boylan, and Tomás Ward. Estimation of Continuous Blood Pressure from PPG via a Federated Learning Approach. *Sensors (Basel, Switzerland)*,

21(18):6311, September 2021.

[6] Boštjan Brumen, Aleš Černezel, and Leon Bošnjak. Overview of machine learning process modelling. *Entropy*, 23(9), 2021.

[7] Yongyi Chen, Dan Zhang, Hamid Reza Karimi, Chao Deng, and Wutao Yin. A new deep learning framework based on blood pressure range constraint for continuous cuffless BP estimation. *Neural Networks*, 152:181–190, August 2022.

[8] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[9] Moajjem Hossain Chowdhury, Md Nazmul Islam Shuzan, Muhammad E. H. Chowdhury, Zaid B. Mahbub, M. Monir Uddin, Amith Khandakar, and Mamun Bin Ibne Reaz. Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques. *Sensors*, 20(11):3127, January 2020. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[10] Jishnu Dey, Aman Gaurav, and Vijay N. Tiwari. InstaBP: Cuff-less Blood Pressure Monitoring on Smartphone using Single PPG Sensor. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5002–5005, July 2018. ISSN: 1558-4615.

[11] 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*, 58:101870, April 2020.

[12] C El-Hajj and P.A Kyriacou. Cuffless blood pressure estimation from PPG signals and its derivatives using deep learning models. *Biomedical Signal Processing and Control*, 70:102984, September 2021.

[13] Mohamed Elgendi. Optimal Signal Quality Index for Photoplethysmogram Signals. *Bioengineering*, 3(4):21, December 2016. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[14] Mohamed Elgendi, Ian Norton, Matt Brearley, Derek Abbott, and Dale Schuurmans. Systolic Peak Detection in Acceleration Photoplethysmograms Measured from Emergency Responders in Tropical Conditions. *PLoS ONE*, 8(10):e76585, October 2013.

[15] Heesang Eom, Dongseok Lee, Seungwoo Han, Yuli Sun Hariyani, Yonggyu Lim, Illsoo Sohn, Kwangsuk Park, and Cheolsoo Park. End-To-End Deep Learning Architecture for Continuous Blood Pressure Estimation Using Attention Mechanism. *Sensors*, 20(8):2338, January 2020. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute.

[16] Ali Farki, Reza Baradaran Kazemzadeh, and Elham Akhondzadeh Noughabi. A Novel Clustering-Based Algorithm for Continuous and Noninvasive Cuff-Less Blood Pressure Estimation. *Journal of Healthcare Engineering*, 2022:e3549238, January 2022. Publisher: Hindawi.

[17] Vincent Fleischhauer, Aarne Feldheiser, and Sebastian Zaunseder. Beat-to-Beat Blood Pressure Estimation by Photoplethysmography and Its Interpretation. *Sensors (Basel, Switzerland)*, 22(18):7037, September 2022.

[18] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23), June 2000.

[19] Chowdhury Azimul Haque, Tae-Ho Kwon, and Ki-Doo Kim. Cuffless Blood Pressure Estimation Based on Monte Carlo Simulation Using Photoplethysmography Signals. *Sensors (Basel, Switzerland)*, 22(3):1175, February 2022.

[20] Latifa Nabila Harfiya, Ching-Chun Chang, and Yung-Hui Li. Continuous Blood Pressure Estimation Using Exclusively Photopletysmography by LSTM-Based Signal-to-Signal Translation. *Sensors (Basel, Switzerland)*, 21(9):2952, April 2021.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[22] Manish Hosanee, Gabriel Chan, Kaylie Welykholowa, Rachel Cooper, Panayiotis A. Kyriacou, Dingchang Zheng, John Allen, Derek Abbott, Carlo Menon, Nigel H. Lovell, Newton Howard, Wee-Shian Chan, Kenneth Lim, Richard Fletcher, Rabab Ward, and Mohamed Elgendi. Cuffless Single-Site Photoplethysmography for Blood Pressure Monitoring. *Journal of Clinical Medicine*, 9(3):723, March 2020. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[23] Bin Huang, Weihai Chen, Chun-Liang Lin, Chia-Feng Juang, and Jianhua Wang. MLP-BP: A novel framework for cuffless blood pressure measurement with PPG and ECG signals based on MLP-Mixer neural networks. *Biomedical Signal Processing and Control*, 73:103404, March 2022.

[24] Siti Nor Ashikin Ismail, Nazrul Anuar Nayan, Rosmina Jaafar, and Zazilah May. Recent Advances in Non-Invasive Blood Pressure Monitoring and Prediction Using a Machine Learning Approach. *Sensors*, 22(16):6195, January 2022. Number: 16 Publisher: Multidisciplinary Digital Publishing Institute.

[25] Syed Ghufran Khalid, Jufen Zhang, Fei Chen, and Dingchang Zheng. Blood Pressure Estimation Using Photoplethysmography Only: Comparison between Different Machine Learning Approaches. *Journal of Healthcare Engineering*, 2018:e1548647, October 2018. Publisher: Hindawi.

[26] Dong-Kyu Kim, Young-Tak Kim, Hakseung Kim, and Dong-Joo Kim. DeepCNAP: A Deep Learning Approach for Continuous Noninvasive Arterial Blood Pressure Monitoring Using Photoplethysmography. *IEEE Journal of Biomedical and Health Informatics*, 26(8):3697–3707, August 2022. Conference Name: IEEE Journal of Biomedical and Health Informatics.

[27] Sang-Hyun Kim, Marc Lilot, Kulraj S. Sidhu, Joseph Rinehart, Zhaoxia Yu, Cecilia Canales, and Maxime Cannesson. Accuracy and Precision of Continuous Noninvasive Arterial Pressure Monitoring Compared with Invasive Arterial Pressure. *Anesthesiology*, 120(5):1080–1097, May 2014.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[29] Tai Le, Floranne Ellington, Tao-Yi Lee, Khuong Vo, Michelle Khine, Sandeep Kumar Krishnan, Nikil Dutt, and Hung Cao. Continuous Non-Invasive Blood Pressure Monitoring: A Methodological Review on Measurement Techniques. *IEEE Access*, 8:212478–212498, 2020.

[30] Jared Leitner, Po-Han Chiang, and Sujit Dey. Personalized Blood Pressure Estimation Using Photoplethysmography: A Transfer Learning Approach. *IEEE Journal of Biomedical and Health Informatics*, 26(1):218–228, January 2022. Conference Name: IEEE Journal of Biomedical and Health Informatics.

[31] Yung-Hui Li, Latifa Nabila Harfiya, Kartika Purwandari, and Yue-Der Lin. Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model. *Sensors (Basel, Switzerland)*, 20(19):E5606, September 2020.

[32] Zheming Li and Wei He. A Continuous Blood Pressure Estimation Method Using Photoplethysmography by GRNN-Based Model. *Sensors (Basel, Switzerland)*, 21(21):7207, October 2021.

[33] Jing Liu, Shirong Qiu, Ningqi Luo, Sze-Kei Lau, Hui Yu, Timothy Kwok, Yuan-Ting Zhang, and Ni Zhao. PCA-Based Multi-Wavelength Photoplethysmography Algorithm for Cuffless Blood Pressure Measurement on Elderly Subjects. *IEEE journal of biomedical and health informatics*, 25(3):663–673, March 2021.

[34] Chenbin Ma, Peng Zhang, Fan Song, Yangyang Sun, Guangda Fan, Tianyi Zhang, Youdan Feng, and Guanglei Zhang. KD-Informer: Cuff-Less Continuous Blood Pressure Waveform Estimation Approach Based on Single Photoplethysmography. *IEEE Journal of Biomedical and Health Informatics*, pages 1–14, 2022. Conference Name: IEEE Journal of Biomedical and Health Informatics.

[35] Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. NeuroKit2: A python toolbox for neurophysiological signal processing. *Behavior Research Methods*, 53(4):1689–1696, feb 2021.

[36] Gloria Martínez, Newton Howard, Derek Abbott, Kenneth Lim, Rabab Ward, and Mohamed Elgendi. Can Photoplethysmography Replace Arterial Blood Pressure in the Assessment of Blood Pressure? *Journal of Clinical Medicine*, 7(10):316, October 2018. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[37] Edward J. Mascha, Dongsheng Yang, Stephanie Weiss, and Daniel I. Sessler. Intraoperative Mean Arterial Pressure Variability and 30-day Mortality in Patients Having Noncardiac Surgery. *Anesthesiology*, 123(1):79–91, July 2015.

[38] Agnes S. Meidert and Bernd Saugel. Techniques for Non-Invasive Monitoring of Arterial Blood Pressure. *Frontiers in Medicine*, 4, 2018.

[39] Elisa Mejía-Mejía, James M. May, Mohamed Elgendi, and Panayiotis A. Kyriacou. Classification of blood pressure in critically ill patients using photoplethysmography and machine learning. *Computer Methods and Programs in Biomedicine*, 208:106222, September 2021.

[40] Terri G. Monk, Michael R. Bronsert, William G. Henderson, Michael P. Mangione, S. T. John Sum-Ping, Deyne R. Bentt, Jennifer D. Nguyen, Joshua S. Richman, Robert A. Meguid, and Karl E. Hammermeister. Association between Intraoperative Hypotension and Hypertension and 30-day Postoperative Mortality in Noncardiac Surgery. *Anesthesiology*, 123(2):307–319, August 2015.

[41] Enric Monte-Moreno. Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques. *Artificial Intelligence in Medicine*, 53(2):127–138, October 2011.

[42] Benjamin Moody, Sicheng Hao, Brian Gow, Tom Pollard, Wei Zong, and Roger Mark. Mimic-iv waveform database, 2022.

[43] Ramakrishna Mukkamala, Jin-Oh Hahn, Omer T. Inan, Lalit K. Mestha, Chang-Sei Kim, Hakan Töreyin, and Survi Kyal. Towards Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice. *IEEE transactions on bio-medical engineering*, 62(8):1879–1901, August 2015.

[44] Govinda Rao Nidigattu, Govardhan Mattela, and Sayan Jana. Non-invasive modeling of heart rate and blood pressure from a photoplethysmography by using machine learning techniques. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 7–12, January 2020. ISSN: 2155-2509.

[45] Gregory Nuttall, Jennifer Burckhardt, Anita Hadley, Sarah Kane, Daryl Kor, Mary Shirk Marienau, Darrell R. Schroeder, Kathryn Handlogten, Gregory Wilson, and William C. Oliver. Surgical and Patient Risk Factors for Severe Arterial Line Complications in Adults. *Anesthesiology*, 124(3):590–597, March 2016.

[46] Rajeev Kumar Pandey, Tse-Yu Lin, and Paul C.-P. Chao. Design and implementation of a photoplethysmography acquisition system with an optimized artificial neural network for accurate blood pressure measurement. *Microsystem Technologies*, 27(6):2345–2367, June 2021.

[47] Jong-Uk Park, Dong-Won Kang, Urtnasan Erdenebayar, Yoon-Ji Kim, Kyoung-Chul Cha, and Kyoung-Joung Lee. Estimation of Arterial Blood Pressure Based on Artificial Intelligence Using Single Earlobe Photoplethysmography during Cardiopulmonary Resuscitation. *Journal of Medical Systems*, 44(1):18, December 2019.

[48] John E. Hall PhD. *Guyton and Hall Textbook of Medical Physiology*. Elsevier Health Sciences, May 2015. Google-Books-ID: 3sWNCgAAQBAJ.

[49] Xina Quan, Junjun Liu, Thomas Roxlo, Siddharth Siddharth, Weyland Leong, Arthur Muir, So-Min Cheong, and Anoop Rao. Advances in Non-Invasive Blood Pressure Monitoring. *Sensors*, 21(13):4273, January 2021. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

[50] Solmaz Rastegar, Hamid GholamHosseini, and Andrew Lowe. Non-invasive continuous blood pressure monitoring systems: current and proposed technology issues and challenges. *Physical and Engineering Sciences in Medicine*, 43(1):11–28, March 2020.

[51] Meng Rong and Kaiyang Li. A Blood Pressure Prediction Method Based on Imaging Photoplethysmography in combination with Machine Learning. *Biomedical Signal Processing and Control*, 64:102328, February 2021.

[52] Monalisa Singha Roy, Rajarshi Gupta, and Kaushik Das Sharma. BePCon: A Photoplethysmography-based Quality-aware Continuous Beat-to-Beat Blood Pressure Measurement Technique Using Deep Learning. *IEEE Transactions on Instrumentation and Measurement*, pages 1–1, 2022. Conference Name: IEEE Transactions on Instrumentation and Measurement.

[53] Muammar Sadrawi, Yin-Tsong Lin, Chien-Hung Lin, Bhekumuzi Mathunjwa, Shou-Zen Fan, Maysam F. Abbod, and Jiann-Shing Shieh. Genetic Deep Convolutional Autoencoder Applied for Generative Continuous Arterial Blood Pressure via Photoplethysmography. *Sensors (Basel, Switzerland)*, 20(14):E3829, July 2020.

[54] Vafi Salmasi, Kamal Maheshwari, Dongsheng Yang, Edward J. Mascha, Asha Singh, Daniel I. Sessler, and Andrea Kurz. Relationship between Intraoperative Hypotension, Defined by Either Reduction from Baseline or Absolute Thresholds, and Acute Kidney and Myocardial Injury after Noncardiac Surgery: A Retrospective Cohort Analysis. *Anesthesiology*, 126(1):47–65, January 2017.

[55] Bernd Volker Scheer, Azriel Perel, and Ulrich J. Pfeiffer. Clinical review: Complications and risk factors of peripheral arterial catheters used for haemodynamic monitoring in anaesthesia and intensive care medicine. *Critical Care*, 6(3):199, April 2002.

[56] Oded Schlesinger, Nitai Vigderhouse, Danny Eytan, and Yair Moshe. Blood Pressure Estimation From PPG Signals Using Convolutional Neural Networks And Siamese Network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1135–1139, May 2020. ISSN: 2379-190X.

[57] Fabian Schrumpf, Patrick Frenzel, Christoph Aust, Georg Osterhoff, and Mirco Fuchs. Assessment of deep learning based blood pressure prediction from PPG and rPPG signals. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3815–3825, June 2021. ISSN: 2160-7516.

[58] Daniel I. Sessler, Joshua A. Bloomstone, Solomon Aronson, Colin Berry, Tong J. Gan, John A. Kellum, James Plumb, Monty G. Mythen, Michael P. W. Grocott, Mark R. Edwards, Timothy E. Miller, Timothy E. Miller, Monty G. Mythen, Michael PW. Grocott, Mark R. Edwards, Gareth L. Ackland, Charles S. Brudney, Maurizio Cecconi, Can Ince, Michael G. Irwin, Jonathan Lacey, Michael R. Pinsky, Robert Sanders, Finton Hughes, Angela Bader, Annemarie Thompson, Andreas Hoeft, David Williams, Andrew D. Shaw, Daniel I. Sessler, Sol Aronson, Colin Berry, Tong J. Gan, John Kellum, James Plumb, Joshua Bloomstone, Matthew D. McEvoy, Julie K. M. Thacker, Ruchir Gupta, Elena Koepke, Aarne Feldheiser, Denny Levett, Frederic Michard, and Mark Hamilton. Perioperative Quality Initiative consensus statement on intraoperative blood pressure, risk and outcomes for elective surgery. *British Journal of Anaesthesia*, 122(5):563–574, May 2019.

[59] Daniel I. Sessler, Christian S. Meyhoff, Nicole M. Zimmerman, Guangmei Mao, Kate Leslie, Skarlet M. Vásquez, Packianathaswamy Balaji, Jesús Alvarez-Garcia, Alexandre B. Cavalcanti, Joel L. Parlow, Prashant V. Rahate, Manfred D. Seeberger, Bruno Gossetti, S. A. Walker, Rajendra K. Premchand, Rikke M. Dahl, Emmanuelle Duceppe, Reitze Rodseth, Fernando Botto, and P. J. Devereaux. Period-dependent Associations between Hypotension during and for Four Days after Noncardiac Surgery and a Composite of Myocardial Infarction and Death: A Substudy of the POISE-2 Trial. *Anesthesiology*, 128(2):317–327, February 2018.

[60] Shota Shimazaki, Shoaib Bhuiyan, Haruki Kawanaka, and Koji Oguri. Features Extraction for Cuffless Blood Pressure Estimation by Autoencoder from Photoplethysmography. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2857–2860, July 2018. ISSN: 1558-4615.

[61] Shota Shimazaki, Haruki Kawanaka, Hiroki Ishikawa, Koichi Inoue, and Koji Oguri. Cuffless Blood Pressure Estimation from only the Waveform of Photoplethysmography using CNN. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5042–5045, July 2019. ISSN: 1558-4615.

[62] Gašper Slapničar, Nejc Mlakar, and Mitja Luštrek. Blood Pressure Estimation from Photoplethysmogram Using a Spectro-Temporal Deep Neural Network. *Sensors (Basel, Switzerland)*, 19(15):E3420, August 2019.

[63] Wolf H. Stapelfeldt, Hui Yuan, Jefferson K. Dryden, Kristen E. Strehl, Jacek B. Cywinski, Jesse M. Ehrenfeld, and Pamela Bromley. The SLUScore: A Novel Method for Detecting Hazardous Hypotension in Adult Patients Undergoing Noncardiac Surgical Procedures. *Anesthesia and Analgesia*, 124(4):1135–1152, April 2017.

[64] Aleksandra Stojanova, Saso Koceski, and Natasa Koceska. Continuous Blood Pressure Monitoring as a Basis for Ambient Assisted Living (AAL) - Review of Methodologies and Devices. *Journal of Medical Systems*, 43(2):24, January 2019.

[65] Louise Y. Sun, Duminda N. Wijeysundera, Gordon A. Tait, and W. Scott Beattie. Association of Intraoperative Hypotension with Acute Kidney Injury after Elective Noncardiac Surgery. *Anesthesiology*, 123(3):515–523, September 2015.

[66] Ali Tazarv and Marco Levorato. A Deep Learning Approach to Predict Blood Pressure from PPG Signals, July 2021. arXiv:2108.00099 [cs, eess].

[67] X.F. Teng and Y.T. Zhang. Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach. In *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, volume 4, pages 3153–3156 Vol.4, September 2003. ISSN: 1094-687X.

[68] Alparslan Turan, Christine Chang, Barak Cohen, Wael Saasouh, Hani Essber, Dongsheng Yang, Chao Ma, Karen Hovsepyan, Ashish K. Khanna, Joseph Vitale, Ami Shah, Kurt Ruetzler, Kamal Maheshwari, and Daniel I. Sessler. Incidence, Severity, and Detection of Blood Pressure Perturbations after Abdominal Surgery: A Prospective Blinded Observational Study. *Anesthesiology*, 130(4):550–559, April 2019.

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[70] Michael Walsh, Philip J. Devereaux, Amit X. Garg, Andrea Kurz, Alparslan Turan, Reitze N. Rodseth, Jacek Cywinski, Lehana Thabane, and Daniel I. Sessler. Relationship between Intraoperative Mean Arterial Pressure and Clinical Outcomes after Noncardiac Surgery: Toward an Empirical Definition of Hypotension. *Anesthesiology*, 119(3):507–515, September 2013.

[71] Weinan Wang, Pedram Mohseni, Kevin L. Kilgore, and Laleh Najafizadeh. Cuff-Less Blood Pressure Estimation From Photoplethysmography via Visibility Graph and Transfer Learning. *IEEE journal of biomedical and health informatics*, 26(5):2075–2085, May 2022.

[72] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional Block Attention Module, July 2018. arXiv:1807.06521 [cs].

[73] Jiaze Wu, Hao Liang, Changsong Ding, Xindi Huang, Jianhua Huang, and Qinghua Peng. Improving the Accuracy in Classification of Blood Pressure from Photoplethysmography Using Continuous Wavelet Transform and Deep Learning. *International Journal of Hypertension*, 2021:1–9, August 2021.

[74] Qingsong Xie, Guoxing Wang, Zhengchun Peng, and Yong Lian. Machine Learning Methods for Real-Time Blood Pressure Measurement Based on Photoplethysmography. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pages 1–5, November 2018. ISSN: 2165-3577.

[75] Xiaoman Xing, Zhimin Ma, Mingyou Zhang, Xi Gao, Ying Li, Mingxuan Song, and Wen-Fei Dong. Robust blood pressure estimation from finger photoplethysmography using age-dependent linear models. *Physiological Measurement*, 41(2):025007, March 2020.

[76] Xiaoman Xing, Zhimin Ma, Mingyou Zhang, Ying Zhou, Wenfei Dong, and Mingxuan Song. An Unobtrusive and Calibration-free Blood Pressure Estimation Method using Photoplethysmography and Biometrics. *Scientific Reports*, 9(1):8611, June 2019.

[77] Xiaoman Xing and Mingshan Sun. Optical blood pressure estimation with photoplethysmography and FFT-based neural networks. *Biomedical Optics Express*, 7(8):3007–3020, August 2016. Publisher: Optica Publishing Group.

[78] Sen Yang, Yaping Zhang, Siu-Yeung Cho, Ricardo Correia, and Stephen P. Morgan. Non-invasive cuff-less blood pressure estimation using a hybrid deep learning model. *Optical and Quantum Electronics*, 53(2):93, January 2021.

[79] Chih-Ta Yen, Sheng-Nan Chang, Liao Jia-Xian, and Yi-Kai Huang. A Deep Learning-Based Continuous Blood Pressure Measurement by Dual Photoplethysmography Signals. *Computers, Materials & Continua*, 70(2):2937–2952, 2022.

[80] Chih-Ta Yen and Cheng-Hong Liao. Blood Pressure and Heart Rate Measurements Using Photoplethysmography with Modified LRCN. *Computers, Materials & Continua*, 71(1):1973–1986, 2022.

[81] Chih-Ta Yen, Jia-Xian Liao, and Yi-Kai Huang. Applying a Deep Learning Network in Continuous Physiological Parameter Estimation Based on Photoplethysmography Sensor Signals. *IEEE Sensors Journal*, 22(1):385–392, January 2022. Conference Name: IEEE Sensors Journal.

[82] Soheil Zabihi, Elahe Rahimian, Fatemeh Marefat, Amir Asif, Pedram Mohseni, and Arash Mohammadi. BP-Net: Cuff-less, Calibration-free, and Non-invasive Blood Pressure Estimation via a Generic Deep Convolutional Architecture, December 2021. arXiv:2112.15271 [cs, eess].

[83] Minghang Zhao, Shisheng Zhong, Xuyun Fu, Baoping Tang, and Michael Pecht. Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Transactions on Industrial Informatics*, 16(7):4681–4690, July 2020.