Cuffless Blood Pressure Estimation for Wearable Applications

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Biomedical Engineering

by

ROHIT RATHNAM N

15BMD0012

Under the guidance of

Prof. Sharmila N

Department of Sensor and Biomedical Technology

VIT,

Vellore.

April, 2019

DECLARATION

I hereby declare that the thesis entitled “Cuffless blood pressure estimation for wearable applications" submitted by me, for the award of the degree of *Bachelor of Technology in Biomedical Engineering* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Sharmila N.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date :

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "Cuffless blood pressure estimation for wearable applications" submitted by **Rohit Rathnam N (15BMD0012)**, **School of Electronics Engineering**, VIT University, for the award of the degree of *Bachelor of Technology in Biomedical Engineering*, is a record of bonafide work carried out by him under my supervision during the period, 01.12.2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : **Signature of the Guide**

Internal Examiner **External Examiner**

Head of the Department

Department of Sensor and Biomedical Technology

ACKNOWLEDGEMENTS

I take this opportunity to express my profound gratitude to my guide, Prof. Sharmila N for her exemplary guidance, monitoring and constant encouragement throughout this project. I would also like to thank the Head of Department of Sensor and Biomedical Technology, the Dean of School of Electronics Engineering, Dr. Harish Kittur, for their cordial support, and guidance. I am extremely grateful and would like to pay sincere and deepest gratitude to our Honourable Chancellor, Dr G. Viswanathan for providing all the students of this University all the required facilities and literally a place to learn and grow beyond boundaries. I would also like to thank the faculty members of School of Electronics Engineering, and the Omya Healthcare for their support and facilitation of the successful completion and execution of this project. I wish to express immense gratitude for my parents, without whom this project would not have been possible, for their constant support and encouragement throughout this endeavour.

Rohit Rathnam N

Executive Summary

Hypertension or increased arterial blood pressure is a chronic disease affecting a large part of the human population. The primary method for measuring blood pressure is through oscillometric methods using a sphygmomanometer. The disadvantage of this technique is that it requires a bulky cuff which restricts blood flow and hence cannot be used for continuous monitoring. In this project, we propose a cuffless technique to monitor blood pressure using only a pulse sensor. Most fitness tracking wearables in the market today already employ pulse sensing to detect heart rate, and the addition of blood pressure is a valuable addition to hypertensive subjects.

The methodology employed was the use of 2D convolutional neural networks (CNN) which acts as a regressor to estimate the systolic and diastolic blood pressure. A wavelet feature transform was performed on the signal conditioned photo plethysmography (PPG) signal to generate a scalogram, which was then provided as input to the CNN. The model was trained and tested on ICU data, and the best performing model was validated on test subjects. A hardware setup was designed and implemented to acquire the PPG signal and it was utilized to validate the algorithm.

CONTENTS Page No.

List of Figures

List of Tables

List of Abbreviations

ABP	Arterial Blood Pressure
API	Application Programming Interface
ADC	Analog to Digital Converter
BP	Blood Pressure
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CVD	Cardiovascular Disease
DAC	Digital to Analog Converter
DBP	Diastolic Blood Pressure
DSP	Digital Signal Processor
ECG	Electrocardiography
EEPROM	Electronically Erasable Programmable Read Only Memory
GPU	Graphics Processing Unit
ICU	Intensive Care Unit
IoT	Internet of Things
IRBP	Inhomogenous Resilient Backpropagation
LSTM	Long - Short Term Memory
MAP	Mean Arterial Pressure
MIMIC	Multiparameter Intelligent Monitoring for Intensive Care
ML	Machine Learning
MSB	Most Significant Bit
PPG	Pulse Plethysmography
RAM	Random Access Memory

SBP	Systolic Blood Pressure
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
UART	Universal Asynchronous Transmitter Reciever
USB	Universal Serial Bus
WHO	World Health Organisation

1. INTRODUCTION

1.1. OBJECTIVE

The primary objective of this project is to develop a novel algorithm to estimate blood pressure using pulse sensor data and test its efficiency on a prototype device. The methodology adopted uses wavelet transformed PPG data on a CNN to perform regression and obtain the systolic (SBP) and diastolic blood pressure (DBP). The project may be divided into two phases, namely:

- Phase 1 - Developing the deep learning algorithm and test it on ICU data
- Phase 2 - Developing a prototype device to test real world accuracy

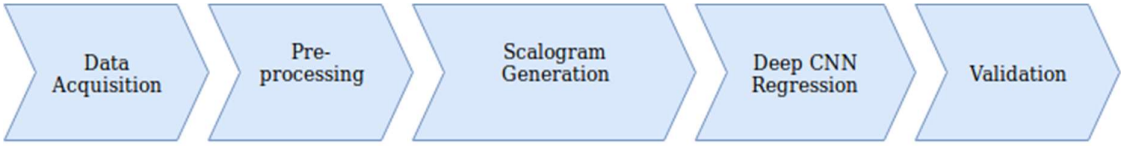

Figure 1.1 Project Outline

1.2. MOTIVATION

Cardiovascular disease (CVD) is a general term for conditions affecting the heart or blood vessels. It is usually associated with a build-up of fat deposits inside the arteries (atherosclerosis) and an increased risk of blood clots.

Cardiovascular diseases (CVD) are the number one cause of deaths in the world according the World Health Organisation (WHO) [1]. Hypertension or high blood pressure is a chronic physiological disorder plaguing nearly a billion people worldwide and hypertensive subjects are at a much higher risk of other cardiovascular diseases. Although hypertension cannot be completely cured, it can be controlled with diet, exercise and other lifestyle changes [2]. Routine feedback of the effectiveness of such lifestyle changes is instrumental in building a good routine. With the growing trend of wearables and fitness / healthcare monitoring devices penetrating the consumer market, more people want to be constantly notified about their health status, so that they may actively work towards a healthier lifestyle.

The measurement of BP through a sphygmomanometer is the gold standard for noninvasive blood pressure estimation. However, this approach has a variety of drawbacks, such as:

1. Bulky cuff which cannot be transported easily or worn discreetly for continuous or routine monitoring

2. Repetitive occlusion of brachial due to measurements resulting in numbness and reduced blood flow to arm
3. Requirement of bulky electronics with pneumatic systems (pump, valve, battery) to inflate and deflate the cuff

Photo plethysmography or PPG is an indirect measure of vascular flow and has been shown to be strongly correlated to the pressure wave or arterial blood pressure variation [3]. Time series data like audio signal processing has been accomplished using convolutional neural networks for classification of setting. Using an analogous approach, the PPG data is also implemented with similar methodology to try and estimate blood pressure. Using Convolutional Neural Networks the complex nonlinear function mapping blood flow to vascular flow was estimated and the systolic and diastolic blood pressure was estimated.

The PPG data can be acquired through the use of a wearable device, and the above-mentioned disadvantages of a BP cuff are avoided. Current wearable fitness trackers already employ a pulse sensor to measure the pulse rate of the individual. The rapid development of machine learning accelerated by high performance cloud infrastructure has allowed ML algorithms to be implemented on data from wearable and IoT enabled devices. Since there is no hardware upgradation of the wearable device required, this feature can be easily implemented into any existing device. The only infrastructure required to implement would be cloud infrastructure and software.

1.3. BACKGROUND

The major behavioral factors associated with heart disease and stroke are unhealthy diet, lack of physical activity, usage of tobacco and alcohol. The effects can be observed in individuals as elevated blood pressure, elevated blood glucose, raised lipid levels in the blood, and obesity. These intermediate risks factors can be measured in primary care facilities and indicate an increased risk of developing a heart attack, stroke, heart failure or other complications.

Ceasing the use of tobacco, reduction of sodium intake in the diet, eating sufficient amount of fruits and vegetables, regular physical activity and avoiding excess use of alcohol have been shown to reduce the risk of cardiovascular disease. [4] These function in addition to drug treatment can reduce the harmful effects of cardiovascular diseases. Through monitoring of blood pressure, hypertensive patients can keep a track on the positive influence of their lifestyle changes. Through this work, we intend to develop a cuffless approach for easy measurement of blood pressure through a wearable device.

The Blood Pressure (BP) is a periodic signal with the heart rate frequency. The upper bound of the blood pressure is called the Systolic Blood Pressure (SBP) while its lower bound is called the Diastolic Blood Pressure (DBP). The mean arterial pressure (MAP) is defined as the average of the blood pressure in a cardiac cycle. If SP is above 140 mmHg or DP is above 90 mmHg, it is considered hypertension that can damage internal body organs. The normal range of MAP is between 70 mmHg and 110 mmHg.

Hypertensive patients can measure their blood pressure from time to time, however, their blood pressure varies over time due to many factors such as food habits, mental situation or stress. Therefore, a continuous blood pressure monitoring seems necessary for accurate diagnosis and treatment of such patients. On the other hand, continuous records of BP also help healthcare providers to prescribe diet choices and medication for individual patients more accurately. The most accurate and common blood pressure measurement devices are sphygmomanometers, which utilize an inflatable cuff around the arm so that the BP can be measured with the height of a column of mercury. This method requires an inflatable cuff, which is inconvenient and prevents continuous measurements because the blood supply to the arm is cut off temporarily.

The basis of cuffless blood pressure estimation on the wave propagation theory for fluids,

which is founded on the natural relationship between the fluid pressure and wave propagation velocity. The theory implies that the blood pressure can be calculated from the heart beat wave velocity. There are a number of disadvantages associated with this method, such as the need for calibration for each person and the expiration of this calibration in short time intervals.

From a review of literature attempting to estimate blood pressure without the use of a BP cuff, two major approaches have been taken up by researchers:

1. Blood pressure estimation using PPG and ECG
2. Blood pressure estimation using PPG alone

From a wearable perspective, acquiring high quality, low noise ECG data is a difficult to process since it is vulnerable to a variety of noise sources. It is also inconvenient for a user to continuously wear adhesive electrodes, which may reduce adoption. Due to the above considerations, a PPG based approach was used to estimate the systolic and diastolic blood pressure.

On review of the literature, it was concluded that most of the works adopt a feature extraction approach and the features were fed into a simple model such as a support vector regressor. The issue with this approach is that it is not easily scalable for large scale implementation as most works used human annotated PPG signals, which is not feasible for commercial viability. Another common approach was the use of recurrent neural networks to perform a time series prediction type operation. The disadvantage here is that the network error becomes a function of time, as each time step depends on the previous. It requires frequent calibration to remain accurate enough, which is difficult since the calibration would require a BP cuff.

P. Li et. al [5] presents a novel wavelet neural network algorithm for the continuous and noninvasive dynamic estimation of blood pressure (BP). Complete BP waveforms are reconstructed based on PPG signals to extract systolic blood pressure (SBP) and diastolic blood pressure (DBP). To improve the robustness, Daubechies wavelet is implemented as the hidden layer node function for the neural network. this paper investigates an inhomogeneous resilient backpropagation (IRBP) algorithm to calculate the weight of hidden layer nodes. The IRBP improves the convergence speed and reconstruction accuracy. Multiparameter intelligent monitoring in Intensive Care (MIMIC) databases, which contain

a variety of physiological parameters captured from patient monitors, are used to validate this algorithm.

E. M. Moreno et. al [6] presents a system to estimate blood glucose level irrespective of patient characteristics and without the need for calibration. The architecture of the system consists of a photoplethysmograph sensor, an activity detection module, a signal processing module that extracts features from the PPG waveform, and a ensemble network that estimates the SBP, DBP and BGL values. Activity detection is performed through a linear classifier that classifies each frame as activity, which is then fed into an automaton. The consolidated feature vector is then tested on ridge regression, multilayer perceptron, support vector machine and random forest algorithm. The most effective results were obtained from random forest tests.

C. Sideris et. al [7] presents a method to estimate arterial blood pressure using recurrent neural networks. Long-short term memory units are used in the network to learn short term and long-term dependencies without overfitting. The blood pressure waveform is predicted from the photoplethysmography data of each patient. However, the model is not generalized to work for every individual and has to be tuned according to the subject.

K. Duan et. al [8] presents a feature exploration methodology to find the best features to accurately estimate arterial blood pressure from photoplethysmography data. Wavelet transform based filtering using symlet 8 and normalization is performed before generating 57 features which include pulse width, amplitude, area, peak approximate angle, derivatives and nonlinear combinations of the above. The best features are selected using the maximum information coefficient. Pearson's correlation coefficient is used to eliminate redundant features.

A. Gaurav et. al [9] presents a smartphone-based approach for cuffless blood pressure estimation. Samsung Galaxy Note 5 equipped with PPG sensor is used for data acquisition. Pan-Tompkins peak detection algorithm is used to detect the peaks and valleys from the PPG and BP signals obtained from the MIMIC - 2 database. Features are extracted from magnitude, temporal variation, acceleration of the PPG wave, and other non-linear cardiac ratios. Two different multi-layer perceptrons are trained for SBP and DBP prediction respectively, and a consensus mechanism is used to arrive at the mean arterial blood pressure.

Y. Kurylyak et. al [10] features a fully connected neural network approach for estimation of blood pressure. 15,000 samples are collected from the MIMIC 2 database for training and validation. 21 features were extracted based on the temporal and magnitude variation of the PPG wave in specific points of interest. The results obtained with low computational cost are better than those presented by works performing linear regression.

F. Lamonaca et. al [12] presents an approach to utilize the camera in smartphones to acquire photoplethysmography signal and characterize it for evaluation of blood pressure. An intensity analysis is performed on pixels to extract the PPG signal. Adaptive thresholding is applied to derive the PPG signal. Temporal and magnitude features are extracted the derived photoplethysmography signal and this is used as input to a feed forward neural network. MIMIC - 2 database is used to train the network with 15,000 sample windows of PPG and BP waves.

P. Su et. al [13] proposes a deep recurrent neural network approach to estimate blood pressure from photoplethysmograph data. The architecture consists of long-short term memory units with forward and backward flow of data. Bidirectional long short-term memory are couple with residual connections to avoid the vanishing gradient problem. Back-propogation through time is used to train the network to predict the future blood pressure using historical data. Multi-task training is implemented to get a better representation of the feature space.

The following table presents a summary of literature pertaining to cuffless blood pressure estimation using PPG signal.

Table 1.1 Summary of Literature Survey

Author	**Title**	**Methodology**	**Reported accuracy**	
			SBP	**DBP**
P. Li et. al (2015) [5]	Novel wavelet neural network algorithm for continuous and noninvasive dynamic estimation of blood pressure from	Wavelet hidden layer and inhomogeneous resilient backpropagation	2.32 ±2.91	1.92 ±2.47

	photoplethysmography			
E. M. Moreno et. al (2011) [6]	Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques	Ensemble neural network with finite state automaton	0.91	0.89
C. Sideris et. al (2016) [7]	Building Continuous Arterial Blood Pressure Prediction Models Using Recurrent Networks	Long-short term memory recurrent neural network	3.80 ±3.46	2.21 ±2.09
K. Duan et. al (2016) [8]	A Feature Exploration Methodology for Learning Based Cuffless Blood Pressure Measurement using Photoplethysmography	Support vector regression	4.77 ±7.68	3.67 ±5.69
A. Gaurav et. al (2016) [9]	Cuff-Less PPG based Continuous Blood Pressure Monitoring – A Smartphone based Approach	Fully connected neural networks	0.16 ±6.85	0.03 ±4.72
Y. Kurylyak et. al (2013) [10]	A Neural Network-based Method for Continuous Blood Pressure Estimation from a PPG Signal	Fully connected neural networks	3.80 ±3.46	2.21 ±2.09
Y. Liang et. al (2018) [11]	Photoplethysmography and Deep Learning: Enhancing Hypertension Risk Stratification	Transferred learning using pretrained CNN with Imagenet weights. Classifier to segregate between hypertensive, normotensive and hypotensive subjects	80.52% F1 score	
F. Lamonaca et. al (2013) [12]	Application of Artificial Neural Network for Blood Pressure valuation with Smartphones	Multilayer feed - forward neural network	<12 mm Hg	
P. Su et. al (2018) [13]	Long-term Blood Pressure Prediction with Deep Recurrent Neural Networks	Bidirectional deep RNN consisting of multilayered LSTM networks with residual connections	3.90	2.66

2. PROJECT DESCRIPTION AND GOALS

The various steps involved are described in the following flowchart:

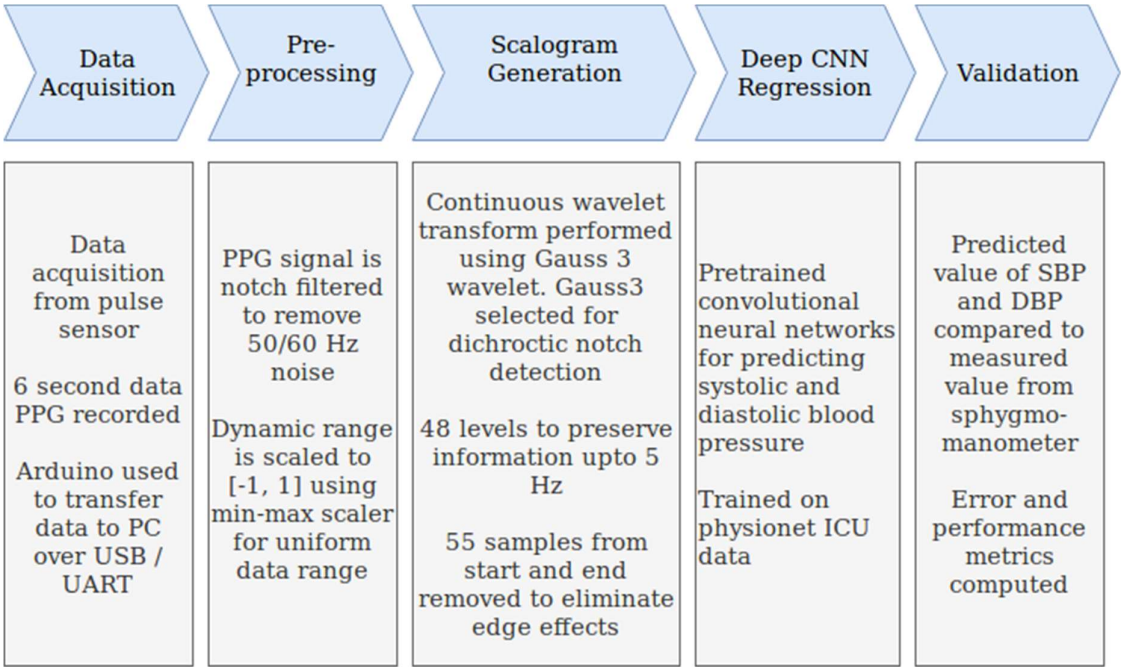

Figure 2.1 Detailed Project Outline

The aim of the project is to setup a machine learning enabled feature which can be implemented on any wearable device with a pulse sensor, that accurately estimates blood pressure accurately with ease.

3. TECHNICAL SPECIFICATION

Hardware specifications:

Data acquisition hardware:

Table 3.1 Hardware Specifications

Component	Specifications
Sparkfun pulse sensor (amped)	Input Voltage 3-5.5V Output Voltage 0.3-V normalized at V/2 Power Consumption 3-4mA
Arduino Nano	Microcontroller ATmega328p Architecture AVR Operating Voltage 5 V Flash Memory 32 KB, 2 KB used by bootloader SRAM 2 KB Clock Speed 16 MHz Analog Pins 8 EEPROM 1 KB Input Voltage 7-12 V (Vin pin) Power Consumption 19 mA
Laptop	Processor Intel Core i7 4th Gen Architecture x86 - 64 bit RAM 4GB DDR3 Clock Speed 1.6 GHz

Cloud hardware:

Table 3.2 Cloud Infrastructure Specifications

Component	Specification
Google Cloud Virtual Machine	Processor Intel Skylake generation 8 core RAM 16 GB 50GB persistent drive
Google Cloud Storage	20 GB object storage API access using gAuth

Googe Colab Jupyter notebook	2vCPU @ 2.2GHz RAM 14 GB GPU 14 GB Tesla P80 100GB disk

4. DESIGN APPROACH AND DETAILS

4.1. DESIGN APPROACH AND METHODS

The design is inspired by a research article [14] by Google scientists who used deep convolutional neural networks for audio scene classification on a Youtube audio clip dataset. The methodology utilizes a Mel spectrogram transformation to create a 2D image of the audio signal which is then fed into a CNN. Modified successful Imagenet models such as VGG, Alexnet, Inception etc. was used to classify into multiple labels of audio scenes.

Using an analogous approach, the similar time series PPG data is converted into an image using a continuous wavelet transform, to generate a spectrogram. This is then fed into a CNN to perform regression and estimate SBP and DBP.

Theoretical basis:

Photo plethysmography also known as pulse oximetry is a technique which measures the oxygen saturation of hemoglobin in the bloodstream by a mechanism called Beer-Lambert's law. This law states that the concentration of solute in a solution is logarithmically proportional to the absorbance of the solution [15]. PPG makes uses of low-intensity infrared green (IR) light tuned to the absorbance frequency of hemoglobin. When light travels through biological tissues it is absorbed by both venous and arterial blood. Since light is more strongly absorbed by blood than the surrounding tissues, the changes in blood flow can be detected by PPG sensors as changes in the intensity of light. This intensity of light is correlated to the voltage of the photodetector. The voltage signal from PPG is proportional to the quantity of blood flowing through the blood vessels. Even small changes

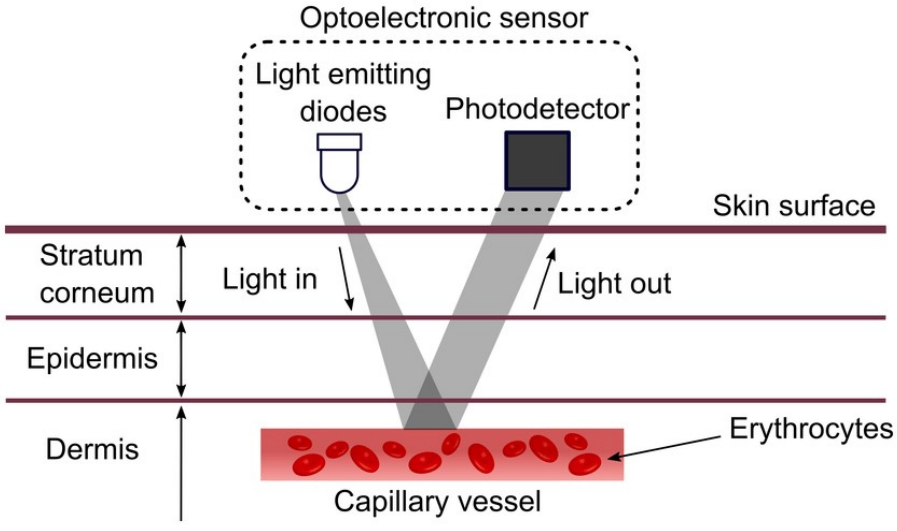

Figure 4.1 Mechanism of Photoplethysmography [25]

in blood volume can be detected using this method, providing high precision.

Data acquisition:

Data acquisition is performed through the hardware setup involving a pulse sensor and Arduino connected to a PC over USB/UART. The pulse sensor has amplification and signal conditioning components in built into the module. The output is an analog signal of range 0.3 - 5V normalized to be centered at 2.5V.

The data pin of the sensor is connected to the analog pin of the Arduino. The microcontroller has a 10-bit ADC operating between 0-5V. This gives an effective resolution of 5/1024 = 0.0048V or 4.8mV. The ATmega328p microcontroller in the Arduino Nano uses a 10 bit [16] successive approximation analog to digital converter (SA ADC)

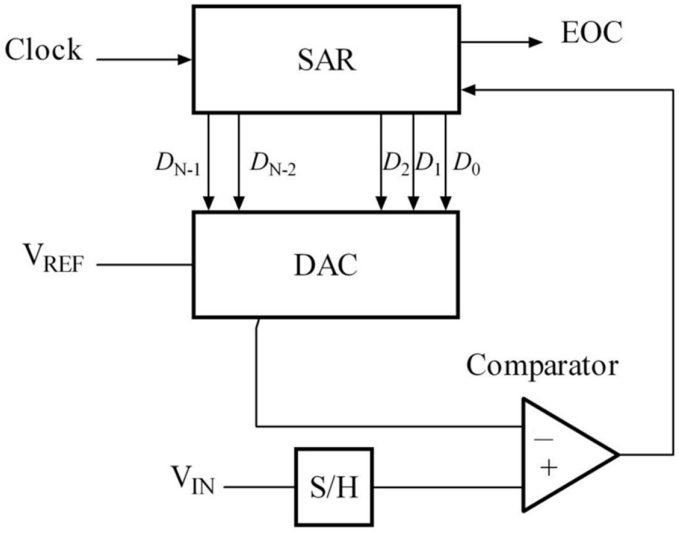

Figure 4.2 Successive Approximation ADC

The successive approximation analog-to-digital converter circuit typically consists of four chief sub circuits:

1. A sample and hold circuit to acquire the input voltage (Vin).
2. An analog voltage comparator that compares Vin to the output of the internal DAC and outputs the result of the comparison to the successive approximation register (SAR).
3. A successive approximation registers sub circuit designed to supply an approximate digital code of Vin to the internal DAC.

4. An internal reference DAC that, for comparison with VREF, supplies the comparator with an analog voltage equal to the digital code output of the SAR_{in}.

The successive approximation register is initialized so that the most significant bit (MSB) is equal to a digital 1. This code is fed into the DAC, which then supplies the analog equivalent of this digital code ($V_{ref}/2$) into the comparator circuit for comparison with the sampled input voltage. If this analog voltage exceeds Vin the comparator causes the SAR to reset this bit; otherwise, the bit is left as 1. Then the next bit is set to 1 and the same test is done, continuing this binary search until every bit in the SAR has been tested. The resulting code is the digital approximation of the sampled input voltage and is finally output by the SAR at the end of the conversion

The timer-counter module of the atmega controller is used to generate interrupts for sampling the ADC at accurate interval of 8ms for 125 Hz sampling frequency. The Atmega328p controller 8-bit timer0 is used to generate the timer interrupt. Prescaler of 1024 is used to get effective timer tick of 16000000/1024 = 15625 ticks per second. Therefore 125 Hz sampling requires a counter value of 15625/125 - 1 = 124.

Interrupts are functions which interrupt the normal execution of code and run the corresponding interrupt service routine. After this function completes execution, the controller continues from the instruction where it got interrupted. Using a timer interrupt allows for strict timing control, compared to the approach of adding a delay between sampling.

The data is sampled for a 6 second interval or 750 samples is saved and sent over UART to the PC. The Arduino board has an on-board USB to UART converter to interface with the USB port of the computer. This controller also serves as the programmer to dump code into the flash memory of the Arduino.

The values of measured voltages are copied from serial monitor and dumped to a .csv file.

Table 4.1 Pin Connections of Hardware Prototype

Function	Arduino Pin	Pulse Sensor Pin
V_{cc}	V_{cc}	V_{cc}
Ground	GND	GND
Signal	A0	Sig

Preprocessing:

The following preprocessing was applied on recorded data from hardware:

The raw .csv data was read and convolution with 10 Hz equiripple FIR low pass digital filter bank was performed on python. The filter coefficients are generated using MATLAB DSP toolbox filter design tool. The stop band gain is set at -40dB and the optimized filter is of order 44.

Convolution is performed between the filter coefficients and the raw data on python, using the NumPy library convolution function. The output array is cropped to be of same dimension as input raw data, and to remove edge effects. Although the wavelet transform eliminates high frequency noise, the data is filtered anyway for visualization purposes.

Min-max scaling is performed on the output filtered data to normalize the data to range [-1, 1]. The scaling formula is given by

$$2 \times \left(\frac{x - min(x)}{max(x) - min(x)}\right) - 1$$

Min-max is performed so that the amplitude differences do not affect the learning ability of the model. The baseline signal amplitude varies among individuals due to skin tone and thickness of the skin of the palm. Min-max scaling allows all data samples to be in the optimal range of [-1, 1] for efficient learning by the machine learning model.

The final data is saved as a .npy file. The file is then uploaded to a google cloud bucket for access by the Colab Jupyter notebook for further processing.

Data source

Primary Source of the data is the Physionet Multiparameter Database (MIMIC 2) [22][23]

The MIMIC II Waveform Database contains thousands of recordings of multiple

physiologic signals and time series of vital signs collected from bedside patient monitors in adult and neonatal intensive care units (ICUs). It is a companion to the MIMIC II Clinical Database, which contains detailed clinical information for many of the patients represented in the Waveform Database. The MIMIC II Waveform Database Matched Subset contains 4,897 waveform records and 5,266 numerics records from the MIMIC II Waveform Database, which have been matched and time-aligned with 2,809 MIMIC II Clinical Database records.

The UCI Machine Learning repository contains preprocessed signals of PPG, ABP and ECG for cuffless blood pressure estimation, derived from MIMIC 2 [17]

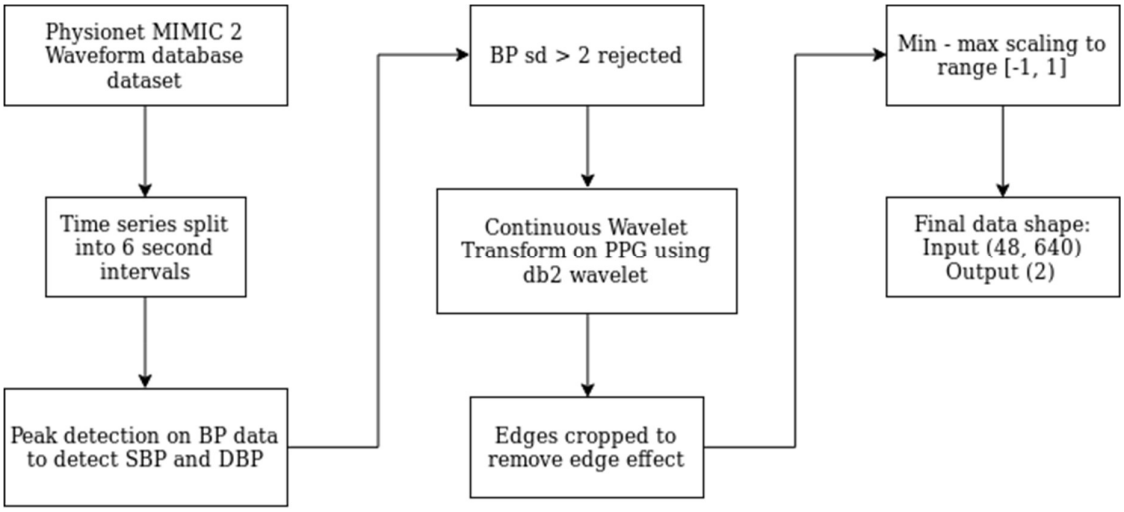

Figure 4.3 Block Diagram of Preprocessing

The following preprocessing was applied on ICU data from the database:
The data set was downloaded from UCI machine learning repository containing a preprocessed PPG, ECG and arterial blood pressure dataset extracted from the Physionet MIMIC 2 ICU database. All the data was recorded at 125 Hz sampling frequency and the data is a floating-point absolute value of voltage in millivolts. The ABP value is in mm. Hg.

The continuous time series data for 1000 patients is split into 6 second intervals. The systolic and diastolic blood pressure is identified for each pulse by a peak detection. This involves computing the 3-point derivative at each point. If the derivative is 0 or if the derivative changes sign it is an indicator of maxima or minima. When the derivative goes from positive to negative, it is annotated as a maxima and if the derivative goes from negative to positive, it is annotated as minima. The upper peak or maxima is systolic blood

pressure value and the lower peak or minima is the diastolic blood pressure.

Sample windows with more than standard deviation of 2 in systolic and diastolic blood pressure are eliminated for wide variation in blood pressure. Since the fluctuation in BP is high, the model might not be able to predict accurately. The average of SBP and DBP is saved to the dataset. No further processing was done on the PPG data as such because it was already filtered for AC interference and high frequency noise.

Scalogram generation

Wavelet transform refers to decomposing signals with finite length or fast decay of the oscillation waveform that is the mother wavelet. The continuous wavelet transforms (CWT) is a formal tool that provides an overcomplete representation of a signal by letting the translation and scale parameter of the wavelets vary continuously. the continuous wavelet transform is a convolution of the input data sequence with a set of functions generated by the mother wavelet.

The Continuous Wavelet Transform (CWT) is used to decompose a signal into wavelets. Wavelets are small oscillations that are highly localized in time. While the Fourier Transform decomposes a signal into infinite length sines and cosines, effectively losing all time-localization information, the CWT's basis functions are scaled and shifted versions of the time-localized mother wavelet. The CWT is used to construct a time-frequency representation of a signal that offers very good time and frequency localization.

One of the main applications of wavelet transform is image compression. The advantage of using wavelet-based coding in image compression is that it provides significant improvements in picture quality at higher compression ratios over conventional techniques. Since wavelet transform has the ability to decompose complex information and patterns into elementary forms, it is commonly used in acoustics processing and pattern recognition, but it has been also proposed as an instantaneous frequency estimator.

Moreover, wavelet transforms can be applied to the following scientific research areas: edge and corner detection, partial differential equation solving, transient detection, filter design, electrocardiogram analysis, texture analysis, business information analysis and gait analysis. Wavelet transforms can also be used in Electroencephalography (EEG) data analysis to identify epileptic spikes resulting from epilepsy.

Continuous Wavelet Transform (CWT) is very efficient in determining the damping ratio of oscillating signals (e.g. identification of damping in dynamic systems). CWT is also very resistant to the noise in the signal.

A scalogram is a representation of the continuous wavelet transform coefficients of a signal. pyWavelet python library is used for CWT. The 1D pulse signal is converted into a 2D image or scalogram using continuous wavelet transform. This provides us with a feature representation which contains time domain as well as frequency domain features, called as time-frequency analysis. This also allows us to convert a 1-dimensional time series into a 2-dimensional signal for input to a convolutional neural network. The scalogram data can be visualized as a grayscale colormap. The scalogram generated is min-max scaled to range [-1, 1] to improve response from neural network.

Daubechies 2 wavelet is suggested to be used by literature but since continuous wavelet transform could not be performed on the discrete wavelet, Gauss 3 wavelet was used for CWT due to similarity to PPG waveform and dicrotic notch. Wavelet decomposition levels 1 - 48 chosen to preserve 0.01 - 5 Hz signal. The frequency corresponding to wavelet decomposition levels was calculated on python using the pyWavelets library.

55 samples from start and end were deleted to remove edge effects. Edge effect is caused due to the finite length signal which has an abrupt ending.

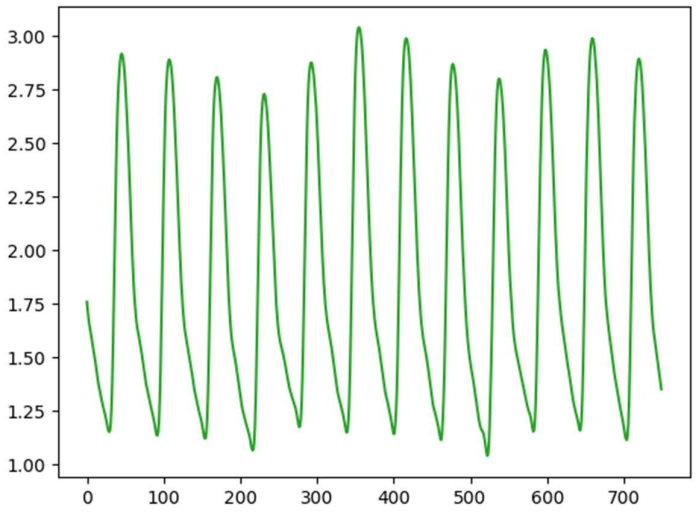

Figure 4.4 Sample Pulse Plethysmograph Signal

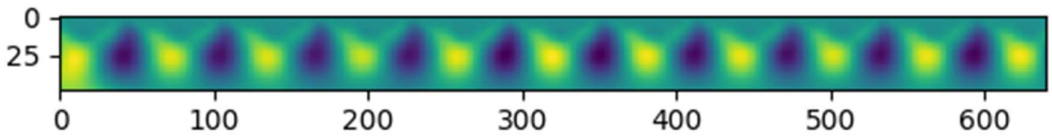

Figure 4.5 Sample Scalogram Data

Deep CNN regression

The functional unit of a neural network is called a neuron, which has weights and biases, as well as an activation function. The input data and weights are multiplied as a dot product and the bias is added to it. The output is then fed to the activation function. Some examples of activation functions are linear, sigmoid, tanh, relu, etc.

Feed-forward neural networks receive an input vector, and transform it through a sequence of hidden layer neurons. Each hidden layer is made up of a set of neurons that is fully connected to all neurons in the previous layer. The last fully-connected layer is called the output layer and this is the layer that performs regression with a linear activation function.

In the case of images, the number of pixels and associated weights for neurons increases very fast with each additional layer, hence a parameter efficient approach to this would be to subdivide the whole image and connect one neuron to each smaller portion. This approach results in the use of a convolving filter of size (n, m) which effectively reduces number of neurons per layer. Hence convolutional neural networks have convolutional layers which performs 2D convolutions on an input image data to extract a feature map of size n x m where (n, m) is the shape of the convolved filter.

Convolutional neural networks combine three architectural ideas to ensure some degree of shift of invariance: local receptive fields, shared weights and spatial/temporal subsampling. Each unit is locally connected in a small neighborhood of the previous layer. These local connections allow neurons to extract elementary visual features such ridges, end points, corners, curves, etc. These features are combined in the higher layers to get more complex features. [18]

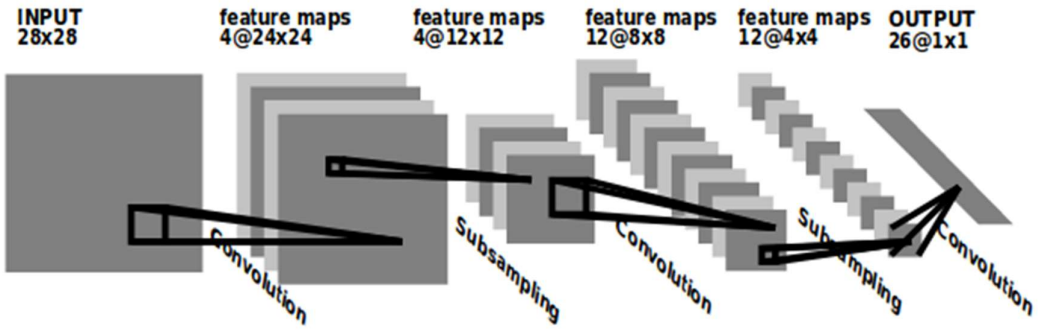

Figure 4.6 Architecture of Convolutional Neural Networks

Pooling layers are used to down sample the image and reduce the data dimensions. The common approach is max pooling, which retains the maximum value of a n x m map. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently and return a single value representing the map.

Dropout layers are used to reduce overfitting of the network. The activation of set percentage of random neurons is set to 0 so that the network can learn to more efficiently represent the feature space without co-evolving with specific neurons too much. Basically, dropout forces the network to learn a sparse representation of a complex space by randomly initializing neurons to weights 0. [19]

Batch normalization is another approach to reduce overfit of the network [20]. The outputs of the network are normalized to the range [-1, 1] using their minimum and maximum response with mean 0 and variance 1 so that the network learns faster. To increase the stability of a neural network, batch normalization normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. After this shift/scale of activation outputs by some randomly initialized parameters, the weights in the next layer are no longer optimal. The gradient descent algorithm undoes this normalization if it is a way for it to minimize the loss function.

Consequently, batch normalization adds two trainable parameters to each layer, so the normalized output is multiplied by a standard deviation parameter (gamma) and a mean parameter (beta). Batch normalization allows the gradient descent algorithm to do the denormalization by changing only these two weights for each activation, instead of losing the stability of the network by changing all the weights.

A loss function is used to quantify the error of the model. The model effectively tries to minimize the loss function over the dataset. For regression, the loss function used is mean squared error. It is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The error is propagated to the previous layer weights using backpropagation. The method used in this work is the Adam algorithm [21], which is an efficient version of gradient descent. It adapts the parameter learning rates based on the average first moment (the mean) and average of the second moments of the gradients (the uncentered variance). The algorithm calculates an exponential moving average of the gradient and the squared gradient. The parameters beta1 and beta2 control the decay rates of these moving averages. The learning rate given by alpha is another parameter to the optimizer.

Mean absolute error is used to visualize real world implication of error. It functions as the secondary error metric. Mean absolute error is chosen over root mean squared error because it is more robust to outliers. Mean squared error grows rapidly for large deviation, and hence the results are skewed by outliers since the error is weighted by its distance from the mean, whereas mean absolute error gives equal weightage to all the error terms

Callbacks are used to monitor the training of the model. Three callbacks were used in this work - model save, reduce learning rate on plateau, and early stopping. The save model callback saves the model weights to a google cloud storage bucket after every 10 epochs so that the progress is not completely lost in case of any errors or crash of program.

Reduce learning rate on plateau monitors the test data loss. If the loss does not decrease for given number of epochs (4), also known as the patience parameter, the learning rate is lowered by a factor of 0.5. This was used to fine tune the model. The early stopping callback stops the training and returns to the weights with best test data loss if the loss does not decrease after 10 epochs. This was used to prevent waste of computation hours.

The model is made on python using the keras library with tensorflow backend. This library produces GPU optimized code for training and testing on GPUs. The backend mathematical computations utilize CUDA, a proprietary C++ library for programming nVIDIA GPUs.

The machine learning code executes on a Jupyter notebook hosted by google colab. Jupyter notebook provides an API interface to execute python scripts on a server. Google Colab provides free access to a GPU to all for 12 hours at a time, to experiment with machine learning. The Jupyter notebook also has some basic linux VM functionality for ease of access and communication with other cloud resources.

Table 4.2 Model Architecture

Model 1	Model 2	Model 3
Input layer	Input layer	Input layer
4 - Conv2D - (2, 4) - same padding - ReLU - BN	4 - Conv2D - (2, 4) - same padding - ReLU - BN	4 - Conv2D - (3, 4) - same padding - ReLU - BN
8 - Conv2D - (2, 4) - same padding - ReLU - BN	8 - Conv2D - (2, 4) - same padding - ReLU - BN	8 - Conv2D - (3, 4) - same padding - ReLU - BN
Max Pooling (2, 2)	Max Pooling (2, 2)	Max Pooling (2, 2)
Dropout (0.2)	Dropout (0.2)	Dropout (0.2)
16 - Conv2D - (2, 4) - same padding - ReLU - BN	16 - Conv2D - (2, 8) - same padding - ReLU - BN	16 - Conv2D - (3, 8) - same padding - ReLU - BN
32 - Conv2D - (2, 4) - same padding - ReLU - BN	32 - Conv2D - (2, 8) - same padding - ReLU - BN	32 - Conv2D - (3, 8) - same padding - ReLU - BN
Max Pooling (2, 2)	Max Pooling (2, 2)	Max Pooling (2, 2)
Dropout (0.2)	Dropout (0.2)	Dropout (0.2)
64 - Conv2D - (2, 8) - same padding - ReLU - BN	64 - Conv2D - (2, 16) - same padding - ReLU - BN	64 - Conv2D - (3, 16) - same padding - ReLU - BN
128 - Conv2D - (2, 8) - same padding - ReLU - BN	128 - Conv2D - (2, 16) - same padding - ReLU - BN	128 - Conv2D - (3, 16) - same padding - ReLU - BN
Max Pooling (2, 2)	Max Pooling (2, 2)	Max Pooling (2, 2)
Dropout (0.2)	Dropout (0.2)	Dropout (0.2)
Fully Connected (512) - BN	Fully Connected (512) - BN	Fully Connected (512) - BN
Fully Connected (512) - BN	Fully Connected (512) - BN	Fully Connected (512) - BN
Fully Connected (512) - BN	Fully Connected (512) - BN	Fully Connected (512) - BN
Fully Connected (2)	Fully Connected (2)	Fully Connected (2)

Validation

The original dataset of ~10,000 samples was shuffled and split into 3 parts in the ratio 0.6:0.2:0.2. The model was trained on 60% data and tested on 20% data. After completion of training, the unseen 20% was used to validate the true accuracy on unseen data. This validation data serves as the basis of comparison between models. A large difference between test loss and validation loss indicates overfit by the model.

Hardware validation was performed using the setup described in the section on data acquisition. Simultaneously, the blood pressure using a BP cuff was measured as the gold standard reference.

Anonymized data was recorded from 10 subjects after their consent, no other details except the PPG signal and blood pressure was recorded to protect their privacy. The PPG data was then fed into the pipeline and the output BP values were compared to the gold standard value, and the error graphs were plotted.

4.2. CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

Input Parameters

After a survey of literature pertaining to cuffless blood pressure estimation, the main approaches shortlisted were blood pressure estimation using PPG and ECG, or blood pressure estimation using only PPG.

The criteria used to perform the constraint and tradeoff analysis are:

1. Ease of acquisition of signal
2. Accuracy of machine learning methods for that approach
3. Quality and noise levels of signal
4. Ease of implementation into current wearables

On consideration of above criteria, PPG was preferable over ECG because of the following reasons:

1. PPG signal is easier to acquire than ECG since it is an optical signal whereas ECG is electrical. It requires less noise reduction circuitry. PPG can be acquired from finger or wrist whereas ECG requires adhesive electrodes to be worn on the chest, with wires leading back to the wearable.

2. Accuracy was comparable in both approached so it was not a major criterion of difference. Even if the accuracy is slightly less in the PPG approach, it is acceptable since consumer end wearables are never intended to function as diagnostic tools, they merely function as early warnings or indicators for the patient to visit their healthcare provider.
3. ECG is affected by a variety of noises [22]:
    a. Power line interference (50 - 60Hz) - inductive and capacitive coupling with the electromagnetic waves produced by the power lines
    b. Baseline wander and motion artifact - caused by variances in the electrode-skin impedance variations induced by movement like vibrations, physical activity and breathing
    c. Electromyography noise (EMG) - caused by depolarization and repolarization of other muscles
    d. Instrumentation noise - caused by thermal noise and other noise sources added up at the instrumentation amplifier

4. On a survey of current wearable device, most of the popular wearables such as Apple's smart watch, Samsung Galaxy Watch Active, FitBit, Mi Band, etc. all have optical heart rate sensors which is basically the same as wrist PPG. This means that the blood pressure estimation feature can be implemented into the device with minimal hardware change if any, since all of the execution happens at the cloud server side. Although there are some niche wearables like Zephyr BioModule, Kenzen Patch, Qardiocore, etc. which offer ECG on a wearable device, their penetration of the consumer market is not that widespread, and hence PPG was chosen over ECG.

Dataset

The main criteria on which a data source is selected is:

1. Data source
2. Amount of noise
3. Ease of access
4. Size of dataset and variability

Based on these criteria, the options shortlisted were Physionet MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) Dataset, University of Queensland Vital Signs Dataset, or a custom dataset created from volunteers. On consideration of the following reasons, MIMIC 2 was considered as the best source for the dataset.

1. The MIMIC 2 database contains data from the ICU and nICU of a tertiary hospital from 2001 to 2008. The data is de-identified and the waveform database is freely available on registration. The waveform database contains the logs of bedside patient monitors, such as SpO_2, invasive arterial blood pressure, multiple lead configuration of ECG, etc. The Vital Signs dataset contains BP, ECG and SpO_2 data of patients under anesthesia. A custom dataset is flexible, the PPG and BP can be collected using a standard machine for various subjects. Since the target application is for real-time monitoring of mostly healthy patients during normal day to day activity, the anesthesia dataset and ICU dataset may not be suitable since underlying chronic illnesses are unknown. Custom database is the best in this case. since the source of data can be finely controlled.

2. Both the MIMIC 2 and Vital Signs dataset have a fair amount of noise. While using a custom dataset, each recording can be carefully monitored and the recording can be done in a controlled environment to avoid any noise. However, this is not a major constraint as such because it can be mitigated by filtering and removing extreme cases.

3. MIMIC 2 database requires registration with an academic email ID and agreement to not distribute the downloaded data without prior permission. Vital signs database is open to all and does not require any special registration or permission. A constructing a custom dataset would require written permission from all subjects and would be tedious to obtain a large group of people.

4. The MIMIC 2 database has 23,810 sets of recordings, the Vital Signs database has 32 sets of recordings, whereas is custom database may have a maximum of maybe 30-50 recordings. Since it is very difficult to obtain data from a wide variety of subjects without access to a hospital or other such institution, MIMIC 2 is considered as the source with maximum variety.

Model Parameters

1. Model parameters were chosen on the basis of previous works, survey of articles on best practices for machine learning, as well as trial of outcomes of different methods

2. Wavelet approach was chosen over the usual Mel spectrogram using Short Time Fourier Transform (STFT) because of the following reasons:
    a. PPG signal has a short bandwidth of 0.01 to 5 Hz. Since the Fourier transform is usually used with logarithmic frequency bands, this is not suitable as the output data size will be very small.
    b. Fourier transform is a representation purely in the frequency domain. Previous works [Duan] suggest that the position of dicrotic notch is an important feature, and the wavelet approach incorporates a kind of hybrid of both time domain and frequency domain features.
    c. Chen et. al presents that a simple wavelet transformed time series performs better than feature engineered STFT windows for CNN applications

3. The basic model architecture was adopted from Chen et. al [cnnscalogram]. In this

work, convolutional neural networks are used for acoustic scene classification of audio signals. The number of neurons in each layer and depth were varied to find the best performing model.

4. Adam optimiser, which is a form of gradient descent was used due to its efficiency in a wide range of machine learning applications.

5. Batch normalisation and dropout was used to prevent overfit. These design changes were made because the training loss was much higher than the validation loss in initial iterations of the model.

5. SCHEDULE, TASKS AND MILESTONES

The key milestones of the project is detailed in the following table:

Table 5.1 Work Splitup by Review

Review 1	**Review 2**	**Review 3**
Design outline	Selection of best model according to literature	Optimize model and hyperparameters
Selection of components	Build and test machine learning model	Validate results on hardware
Selection of dataset and preprocessing	Assemble prototype	Complete documentation

The project timeline is presented in the following Gantt chart:

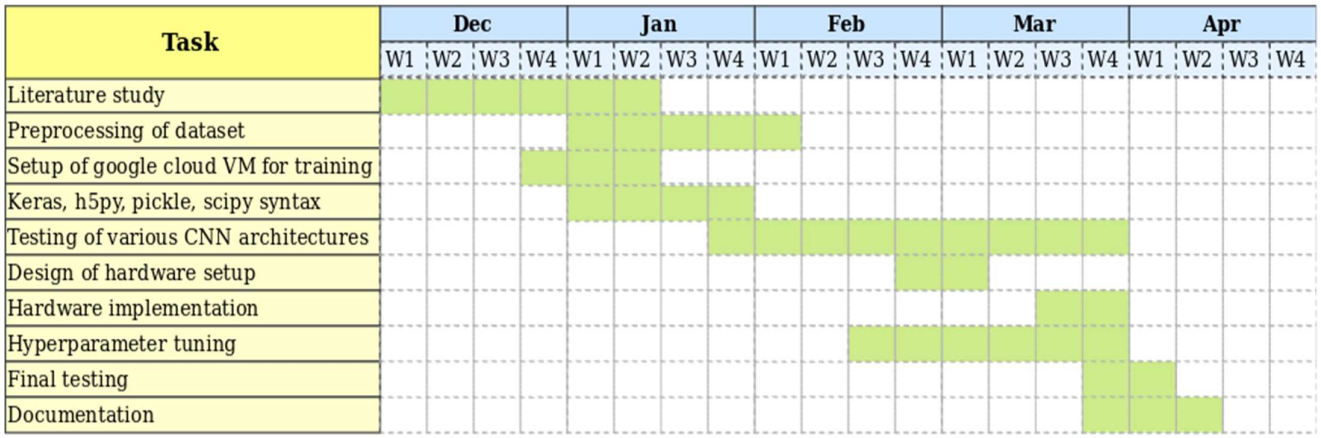

Figure 5.1 Gantt Chart of Tasks

6. PROJECT DEMONSTRATION

The data is collected from 10 subjects for real world validation using the hardware setup

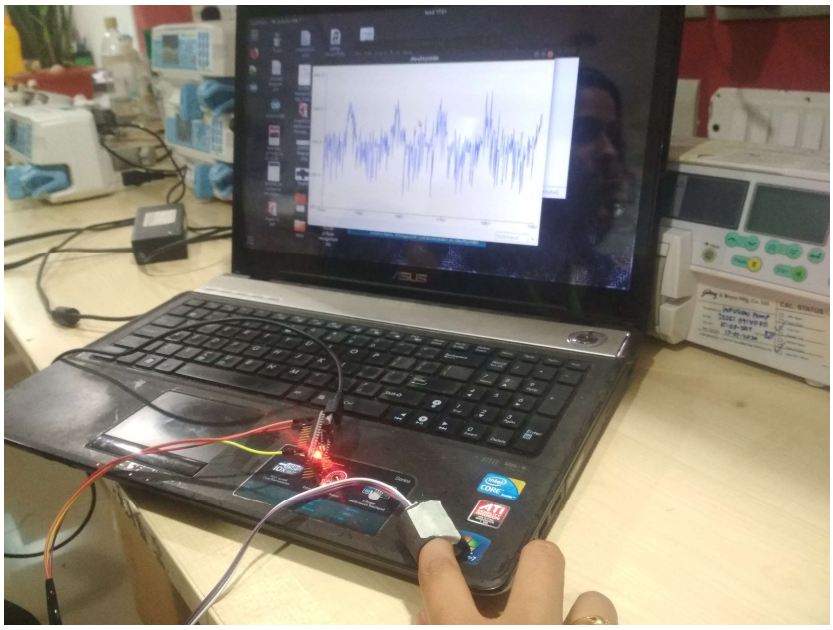

Figure 6.1 Data Acquisition Setup

The blood pressure is recorded using a standard blood pressure cuff in an X70 patient monitor manufactured by Omya Healthcare.

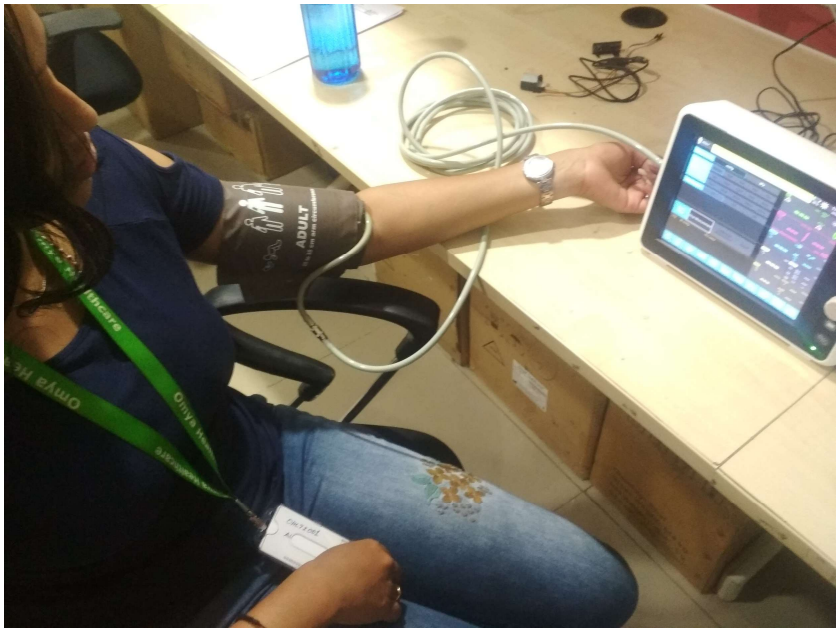

Figure 6.2 Blood Pressure Measurement Setup

The acquired signal is convolved with a low pass filter, and fed into the Jupyter notebook after wavelet transformation and normalisation as described in the previous sections. The results are tabulated below.

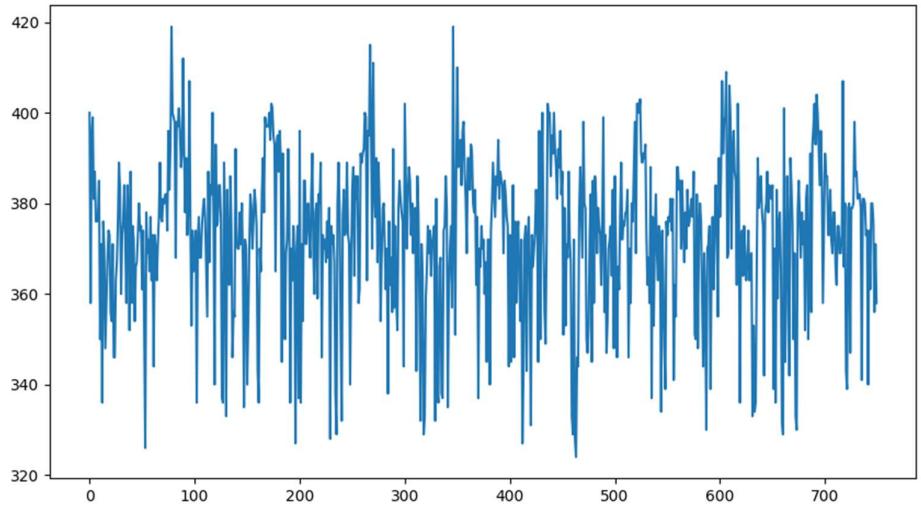

Figure 6.3 Noisy PPG Signal

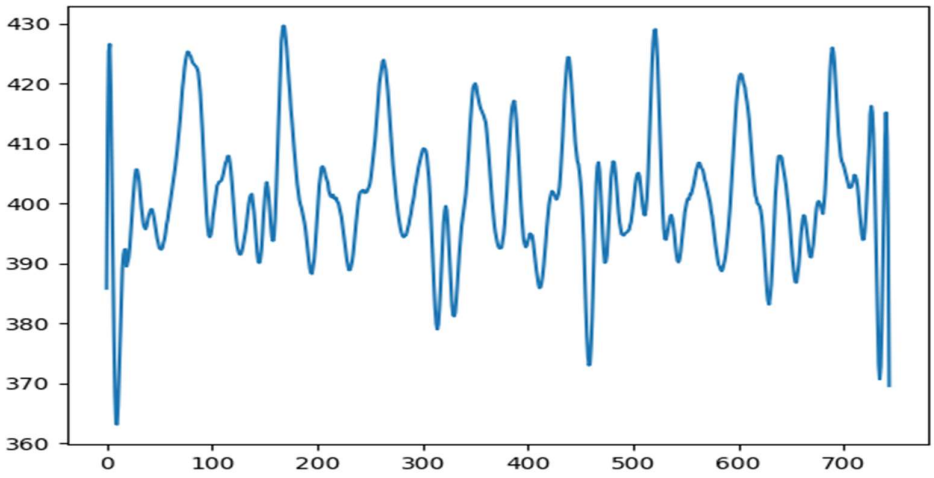

Figure 6.4 Low Pass Filtered PPG

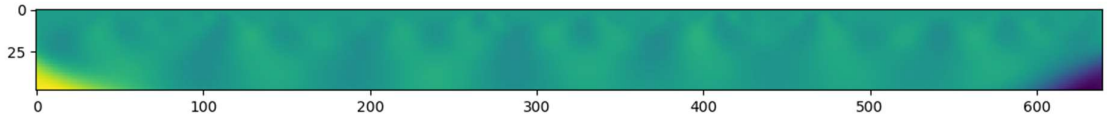

Figure 6.5 Scalogram Generated from Acquired Signal before Min – Max Scaling

Table 6.1 Hardware results

Subject	Actual		Estimated (rounded to nearest whole)		Error	
	SBP	**DBP**	**SBP**	**DBP**	**SBP**	**DBP**
1	134	88	138	91	-4	3
2	112	71	110	74	12	-7
3	109	72	103	79	6	-7
4	119	71	128	75	-9	-4
5	127	84	135	96	8	-12
6	102	86	104	81	-2	5
7	133	81	137	75	-4	-6
8	115	67	102	63	13	9
9	128	81	121	83	7	-2
10	135	92	140	83	-5	8

Mean Absolute Error for Systolic Pressure - 7.1

Mean Absolute Error for Diastolic Pressure - 6.3

7. RESULT AND DISCUSSION

Result:

Data distribution:

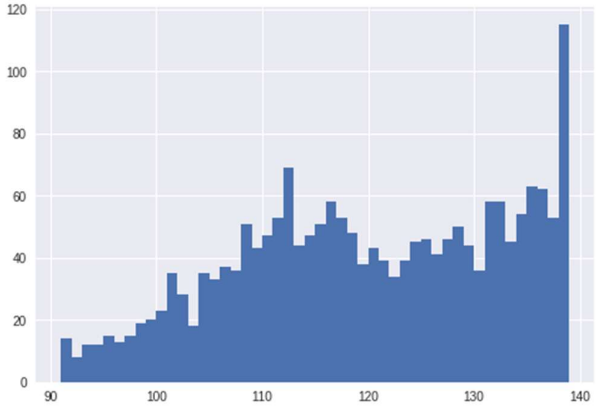

Figure 7.1 Data Distribution of SBP

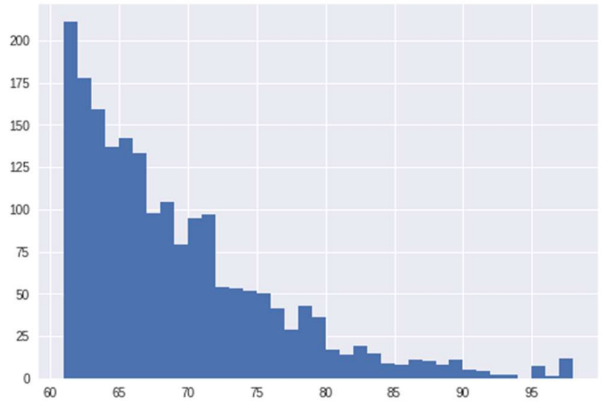

Figure 7.2 Data Distribution of DBP

Table 7.1 Model metrics

Model	Validation Loss	Mean Absolute Error
Model 1	74.57	6.21
Model 2	50.46	4.78
Model 3	60.04	5.40

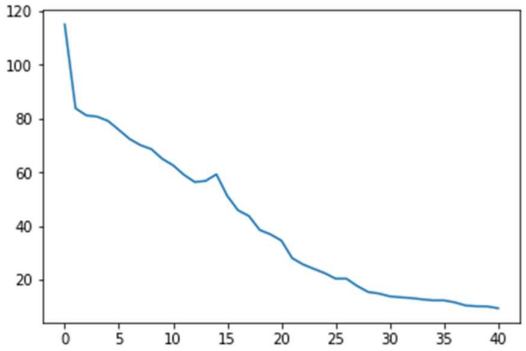

Figure 7.3 Training Loss

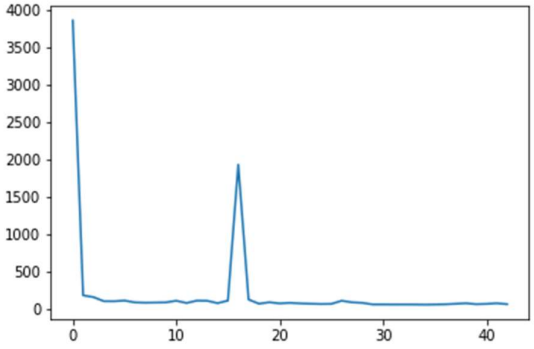

Figure 7.4 Validation Loss

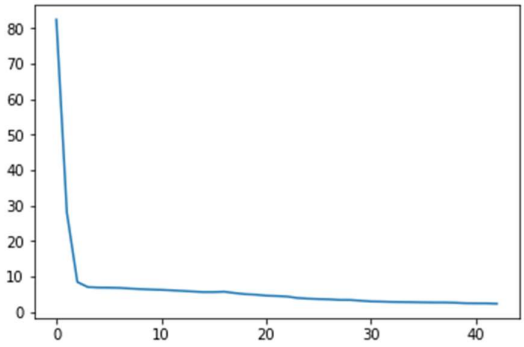

Figure 7.5 Training Mean Absolute Error

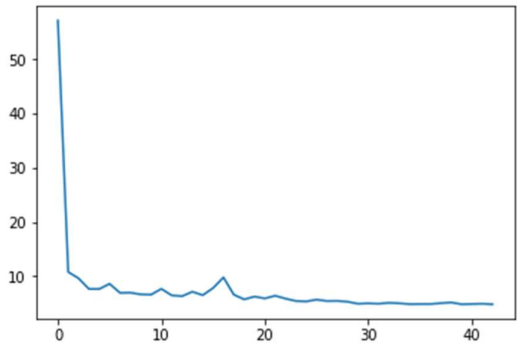

Figure 7.6 Validation Mean Absolute Error

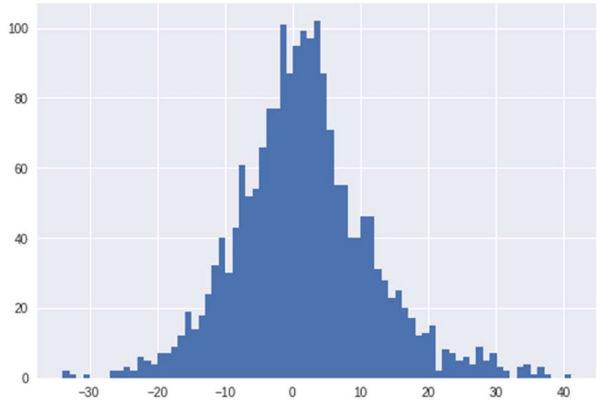

Figure 7.7 Model 1 Error Histogram of SBP

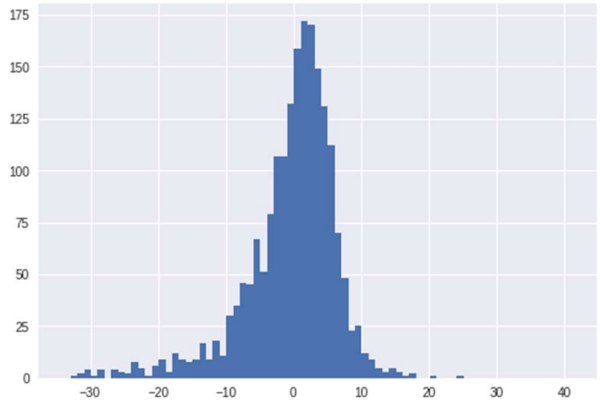

Figure 7.8 Model 1 Error Histogram of DBP

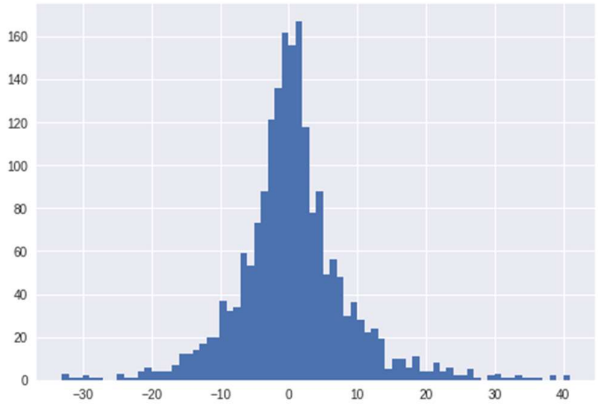

Figure 7.9 Model 2 Error Histogram of SBP

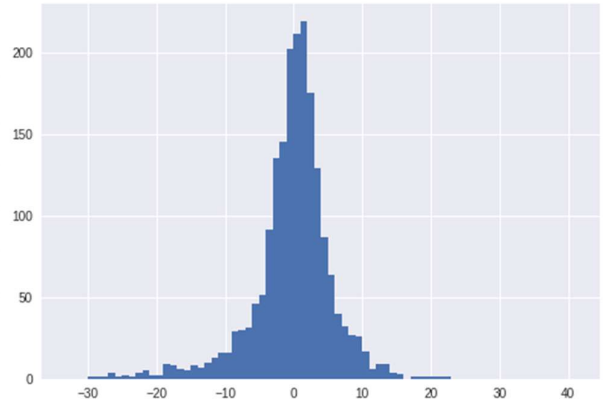

Figure 7.10 Model 2 Error Histogram of DBP

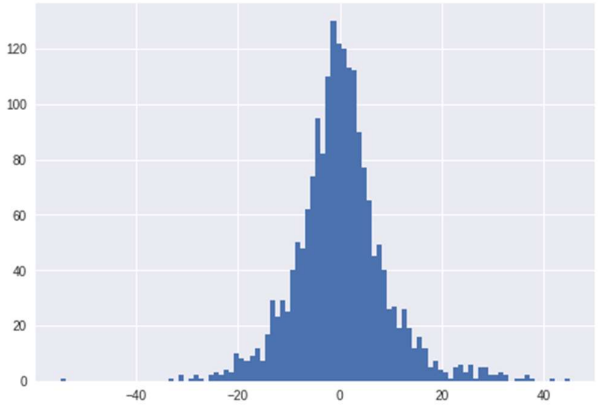

Figure 7.11 Model 3 Error Histogram of SBP

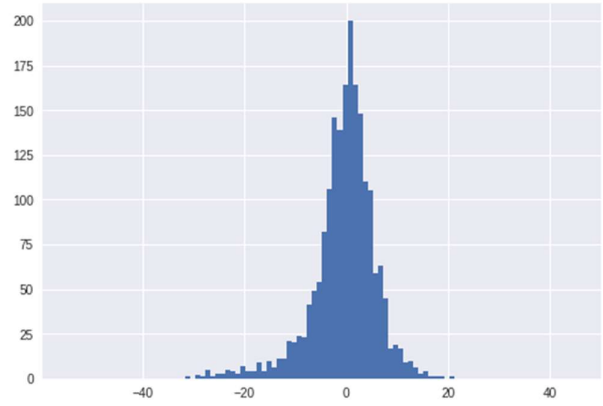

Figure 7.12 Model 3 Error Histogram of DBP

Result Analysis:

The results of the validation test indicate that model 2 provides the best fit of the three. Although these accuracies are not comparable to that presented by other literature, it is to be noted that this approach is much more scalable than other approaches which require human intervention to annotate the data. On analysis of possible sources of error, the following improvements can be made to improve model metrics:

1. Increase the size of dataset for more data variability
2. Algorithms to remove outliers in preprocessing
3. Include data of healthy individuals along with ICU data
4. Implement more complex CNN models like VGG, AlexNet, ResNet, Inception, etc.

The hardware utilized to acquire the PPG signal is also very rudimentary. Utilizing a better-quality sensor and proper construction of enclosure to protect from external light sources might result in a much better result than presented here.

This work was not intended to be a diagnostic tool, the output from the model serves as an indicator for users to check their blood pressure or visit their healthcare provider.

8. SUMMARY

In conclusion, this project involved the design and implementation of a cuffless blood pressure estimation algorithm using a pulse sensor. The pulse sensor data is transformed to a scalogram through a continuous wavelet transform, which is then fed into a convolutional neural network. The network performs regression to give the systolic blood pressure and diastolic blood pressure as output. The applications of this are in wearable devices, which already implement a pulse sensor to detect pulse rate. Through minimal or no change to the wearable hardware and cloud infrastructure, this feature can be setup in the wearable device. Cuffless blood pressure monitoring allows hypertensive patients to routinely check their blood pressure through their smart wearable on the go, at any location. The convenience allows for regular blood pressure checks and gives the user continuous feedback on the effect of their lifestyle and habits on their blood pressure so that they can incorporate positive changes to it.

References

[1] WHO, https://www.who.int/gho/ncd/risk_factors/blood_pressure_prevalence_text/en

[2] WHO, https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[3] M. Elgendi, "On the analysis of fingertip photoplethysmogram signals," Current cardiology reviews, vol. 8, no. 1, p. 14, 2012.

[4] Heart.org, https://www.heart.org/en/health-topics/high-blood-pressure/the-facts-about-high-blood-pressure/five-simple-steps-to-control-your-blood-pressure

[5] L. Peng, L. Ming, Z. Xu, H. XiaoHui, P. Bo, Y. ZhaoLin, C. HongDa, "Novel wavelet neural network algorithm for continuous and noninvasive dynamic estimation of blood pressure from photoplethysmography", Science China Information Sciences, 59(4), Sep. 2015

[6] E. Monte-Moreno, "Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques," Artificial Intelligence in Medicine, vol. 53, no. 2, pp. 127–138, Oct. 2011

[7] C. Sideris, H. Kalantarian, E. Nemati, M. Sarrafzadeh, "Building Continuous Arterial Blood Pressure Prediction Models Using Recurrent Networks" IEEE 2016

[8] K. Duan, Z. Qian, M. Atif, G. Wang, "A Feature Exploration Methodology for Learning Based Cuffless Blood Pressure Measurement using Photoplethysmography", The 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'16)

[9] A. Gaurav, M. Maheedhar, V. N. Tiwari, R. Narayanan, "Cuff-Less PPG based Continuous Blood Pressure Monitoring – A Smartphone based Approach", The 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'16)

[10] Y. Kurylyak, F. Lamonaca, D. Grimaldi, "A Neural Network-based Method for Continuous Blood Pressure Estimation from a PPG Signal", IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2013

[11] J. F. Kelleher, "Pulse Oximetry", Journal of Clinical Monitoring Vol 5 No I January 1989

[12] F. Lamonaca, K. Babe, Y. Kurylyak, D. Grimaldi, W. V. Moer, A. Furfaro, V. Spagnuolo, "Application of the Artificial Neural Network for blood pressure evaluation with smartphones," 2013 IEEE 7th International Conference on

Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), Berlin, 2013, pp. 408-412

[13]P. Su, X. Ding, Y. Zhang, J. Liu, F. Miao, N. Zhao, “Long-term Blood Pressure Prediction with Deep Recurrent Neural Networks”, IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), March 2018

[14] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, K. Wilson, “CNN architectures for large-scale audio classification”, Jan 2017

[15]Z. Xu, J. Liu, X. Chen, Y. Wang, and Z. Zhao, “Continuous blood pressure estimation based on multiple parameters from eletrocardiogram and photoplethysmogram by Back-propagation neural network,” Computers in Industry, vol. 89, pp. 50–59, Aug. 2017

[16]Sparkfun Product Datasheets - Atmega328p, https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf

[17]M. Kachuee, M. M. Kiani, H. Mohammadzade, M. Shabany, Cuff-Less High-Accuracy Calibration-Free Blood Pressure Estimation Using Pulse Transit Time, IEEE International Symposium on Circuits and Systems (ISCAS'15), 2015

[18]Y. LeCun, Y. Bengio, “Convolutional Networks for Images, Speech, and Time-Series”

[19]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, Journal of Machine Learning Research 15 (2014) 1929-1958

[20]S. Ioffe, C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”

[21]S. J. Reddi, S. Kale, S. Kumar, “On the convergence of Adam and beyond”, International Conference on Learning Representations, 2018

[22]Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages, June 2000

[23]M. Saeed, M. Villarroel, A.T. Reisner, G. Clifford, L. Lehman, G.B. Moody, T. Heldt, T.H. Kyaw, B.E. Moody, R.G. Mark. Multiparameter intelligent monitoring in intensive care II (MIMIC-II): A public-access ICU database. Critical Care Medicine 39(5):952-960, May 2011

[24]Y. Liang, Z. Chen, R. Ward, and M. Elgendi, “Photoplethysmography and Deep

Learning: Enhancing Hypertension Risk Stratification," Biosensors, vol. 8, no. 4, p. 101, Oct. 2018

[25]J. Moraes, M. Rocha, G. Vasconcelos, J. Vasconcelos Filho, V. de Albuquerque, and A. Alexandria, "Advances in Photopletysmography Signal Analysis for Biomedical Applications," Sensors, vol. 18, no. 6, p. 1894, Jun. 2018

[26]M. T. Hagan, H. B. Demuth, M. H. Beale, O. De Jesus, "Neural Network Design, 2nd Edition"

[27]Y. Liang, Z. Chen, R. Ward, and M. Elgendi, "Hypertension Assessment via ECG and PPG Signals: An Evaluation Using MIMIC Database," Diagnostics, vol. 8, no. 3, p. 65, Sep. 2018

[28]A. Suzuki and K. Ryu, "Feature Selection Method for Estimating Systolic Blood Pressure Using the Taguchi Method," IEEE Transactions on Industrial Informatics, vol. 10, no. 2, pp. 1077–1085, May 2014

[29]Wikipedia – Sequential Approximation Analog to Digital Converter https://en.wikipedia.org/wiki/File:SA_ADC_block_diagram.png#filelinks