Non-Invasive Blood Pressure Estimation Using Photoplethysmography and IoT-ML Integration

Final Year Design Project

By

Ridwan Sharif

213-15-4381

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements forthe **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Dr. Arif Mahmud
Associate Professor
and Associate Head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by

Dr. Md. Zahid Hassan
Associate Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project, titled **"Non-Invasive Blood Pressure Estimation Using Photoplethysmography and IoT-ML Integration"** submitted by **Ridwan Sharif** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

BOARD OF EXAMINERS

Dr. Arif Mahmud — **Chairman**
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Fizar Ahmed — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Mushfiqur Rahman — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md. Manowarul Islam — **External Examiner**
Professor
Department of Computer Science and Engineering
Jagannath University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Arif Mahmud, Associate Professor and Associate head**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this projectnor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Dr. Arif Mahmud
Professor and Associate head
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Dr. Zahid Hassan
Associate Professor
Department of Computer Science and Engineering
Daffodil International University

Submitted by:

Ridwan Sharif
Student ID: 213-15-4381
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to **Dr, Arif Mahmud, Associat Professor and Associate Head**, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of **IoT and ML** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting themat all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Com- puter Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

The article examines the effectiveness of the cuffless blood pressure (BP) estimator based on photoplethysmography (PPG) signals, patient demographics, and machine learning. A prototype of IoT was created on the basis of an ESP32 microcontroller and a sensor of optical measurements of the maximal30102 to obtain physiological measurements. It was deployed with a full pipeline like signal processing, morphology and nonlinear features, feature selection (Mutual Information) with Recursive Feature Elimination (RFE) and SHAP analysis, and, finally, training of some machine learning models. CatBoost and Random Forest were optimized by randomizing some values. Although the approach was rigorous, predictive accuracy was lower than clinical. The highest performing models were those with a mean absolute error (MAE) of over 7 mmHg and R2R 2 of less than 0.25. These results reveal that superior machine learning algorithms are unable to combat low input quality. The makeup of the sensor, high sensitivity (maximum 15-15 ohms) and restricted waveform fidelity of the MAX30102 sensor was insufficient to capture the morphology features needed for accurate BP determination. Also, the sample size was not adequate, and the sample was not demographically balanced Abbott et al., 2019), which may further undermine the generalizability. The novelty of the work is the evidence of the limitation: revealing that cheap PPG sensors like the MAX30102 cannot be casually used to estimate BP. To enhance cuffless BP monitoring, future investigations need to utilize better quality sensors, larger, more heterogeneous datasets, and consolidated validation systems.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

This Chapter outlines the problem statement, the objective, the motivation and purpose of this research that is stated and the short outline of the methodology and how this report has been organized. Through this chapter the whole of the study can be glimpsed and primary understood.

1.1 Introduction

Hypertension, a persistently high blood pressure (BP), was a severe international health issue and the greatest modifiable cardiovascular (CVD) risk factor and is closely associated with myocardial infarction, stroke, heart failure, and chronic kidney disease [1]. It's largely being asymptomatic characteristic commonly allows its detection after the occurrence of a severe case, which makes hypertension to be referred to as the silent killer. The auscultatory or oscillometric sphygmomanometry, which is the current gold standard in the measurement of BP, has existed for more than 100 years, but they are just a point in time, episodic measure of hemodynamic status [1]. The gap in these intermittent measurements made in clinical settings is subject to the so-called white-coat hypertension effect, does not account for important determinants of BP variability, including nocturnal BP changes and post-awake BP spikes, and cannot comprehensively assess the range of BP variability [2]. In contrast, invasive arterial line monitoring provides continuous BP data that is critical to informing critical-care interventions, but it is too invasive (establishing an arterial cannula), and associated with infection and thrombosis risk, as well as patient discomfort, to be used routinely or in an ambulatory setting to profile patients [3]. Hypertension and other types of clinical deficits highlight the necessity of non-invasive and continuous BP monitoring technology, producing longitudinal and personalized data[4].

In the current report, the design, methodological development, and initial assessment of a system that aims to fulfil this requirement via the combination of wearable sensors, the Internet of Things (IoT), and Artificial Intelligence (AI) will be outlined. The system consists of a wearable cuff device based on strain gauges and a central cloud-based analytics engine. The wearable unit measures beat-by-beat cuff deflection signal over a 24-hour wearing process, and all of the beat-by-beat signals are transmitted to the cloud engine to produce uninterrupted estimates in BP values and ensemble BP trends with the associated confidence intervals. These ensemble trajectories represent a time series of the whole picture of BP change that gives clinicians and patients a full longitudinal profile, which is a critical advance in the field of hypertension.

1.2 Motivation

The technological environment has become ready to resolve the issue of continuous and non-invasive monitoring of BP. The motivation behind this project is based on three technological pillars, namely: the omnipresence of wearable sensors, the interconnectedness of the Internet of Things, and the analytical ability of Artificial Intelligence.

To start with, photoplethysmography (PPG) is one of the dominant sensing modalities in the consumer electronics market. PPG is a non-invasive, less expensive, and simple optical method to measure any volumetric change in blood in peripheral circulation[5]. Previously used for clinical pulse oximetry purposes only, PPG sensors have since then become the most common sensor found in practically endless wearable technology such as smartwatches and fitness trackers, offering real-time heart rate, heart rate variability, and blood oxygen saturation data[6]. The similarity between the waveform of the PPG and the arterial blood pressure gives evidence that information about BP is incorporated into the PPG signal, making it an excellent opportunity to retrieve this information [7].

Second, the IoT (Internet of Things) can be used to provide the necessary framework to turn data captured by such wearable sensors into useful health analytics. The functionality of IoT architecture allows the smooth and secure gathering of real-time physiological data using individuals, which is wirelessly transferred to the cloud-based system to be stored and analyzed [4]. Such connectivity erases geographical boundary conditions, making it easy to conduct remote patient monitoring (RPM) and ensuring that healthcare providers monitor the health conditions of the patients without necessarily being at their locations, thus enhancing accessibility and cutting down the cost of healthcare delivery [8].

Last, Artificial Intelligence (AI) is most useful, especially Machine Learning (ML) and Deep Learning (DL), the analytical engine needed to extract an understanding of the linear and non-linear relationship between the PPG signal, Demographic info and blood pressure. Whereas simpler approaches had to contend with signal noise and variations over time, sophisticated AI algorithms are now able to learn good representations directly off unprocessed signals, and automatically learn those hidden details that are predictive of BP fluctuations [9]. These three technologies, auspiciously combined by the term synergy, PPG (ubiquitous sensing), IoT (pervasive connectivity), and AI (intelligent analysis), form a very powerful, yet timely window of opportunity to devise a new solution to cuffless, continuous BP estimation.

1.3 Objectives

The current study aims at conceptualizing and reporting an end-to-end system of non-invasive blood pressure estimation that is both technologically valid and clinically applicable and in accordance with established engineering principles. The particular goals of the project are as follows:

1. To Develop an pipeline to process raw PPG data from IoT device to extract features to predict blood pressure using machine learning model and display the results in web-interface.

2. To extract features from raw PPG signal data, that are useful for blood pressure prediction.

3. To compare and evaluate the performance of multiple Machine Learning and Deep Learning models on extracted features from IoT based raw PPG data collection for Blood pressure prediction.

4. To analyze the impact of feature extracted dataset from raw PPG signal data and sensor limitation on feature extraction and model performance.

1.4 Methodology

The engineering requirements of this project will be achieved by a step-by-step, phase-oriented methodological framework that is very close to an orthodox medical device development path. The labor process begins with the stage of strict research and ends with the post-development audit.

1. Gap Analysis and Literature Review: To come up with a strong theoretical background, a systematic review of the available academic and industrial literature will be conducted. Existing PPG-based blood pressure estimators will be interrogated in this review by critically reviewing their advantages and limitations, and hence informing the design of the proposed system.

2. Architecture Design and System Requirements: The gap analysis will be used to build on the results, leading to a comprehensive set of functional and non-functional requirements being formulated. A multi-layered system architecture focused on IoT will be defined, where the hardware, software processing, and communication protocols of each layer will be specified.

3. Development and training of AI models: An appropriate AI architecture will be chosen and set up. It will be trained and tested on locally and privately collected data from the IoT device developed, where the simultaneous PPG, Demographic info and cuff-based Blood Pressure electrical machine provide a foundation on which to test.

4. Integrating and Implementing Prototypes: The approach to the development of a proof-of-concept prototype will be described, including the process of developing sensor hardware, developing a web interface application, and establishing cloud-based backend services and model deployment.

5. Comparison Analysis and Performance Assessment: An elaborate evaluation plan will be outlined. The performance of the systems will be quantitatively evaluated with various accuracy standards by using a special test set separated from the privately collected data set. Taken together, the metrics will be compared to the state of the art of the extant literature.

1.5 Project Outcome

The key output of this work is a well-crafted technical report on an expert level which serves as a complete blueprint of a non-invasive, PPG-based, IoT-AI, Web integrated blood pressure monitoring system. The report provides a hierarchical structure of how the system is designed down to physiological principles up to detailed structure of hardware, software and AI modules. It gives it a tough data-based evaluation of the system performance against set clinical benchmark and a detailed review of compliance with key regulatory and engineering requirements.

1.6 Organization of the Report

The current report is divided into six major parts so that the issue under consideration could be logically organized and covered thoroughly.

In chapter one introduction outlines the clinical imperative, the technological case, and the main project objectives. After that in chapter two background discusses the physiological basis of PPG, a literature review of the relevant literature concerning methods of estimating BP and relevant research gaps. In chapter three Research methodology, the presentation of the methodology is carried out in detail the architecture of the system, specifications of the design, the design of the AI model, and the project plan are discussed. Chapter four Implementation and Results, the environment of implementation is outlined, and the performance of the system is assessed; the meaning of the results is explained. Later in chapter five Engineering Standards and Design Challenges, The five themes that are discussed include compliance with engineering standards, implications to the wider society and its ethics, complexity of the engineering factors that are involved, and the engineering relationship of the system to the clinical practice. Finaly in the last chapter, chapter six Conclusion, the most important results are stated, limitations of the project are mentioned, and the recommendations of prospective studies are provided.

Chapter 2
Background

The chapter offers essential background information, a preceding collection of precise investigations, a logical tabular representation of subsequent research, an evaluation of gaps, and a definitive conclusion. Through this chapter, the existing works and the innovation of this study can be preserved

2.1 Introduction

Firm consideration of the ability of photoplethysmography (PPG) to estimate blood pressure requires in-depth familiarity with the optical sensing technology as well as the physiological connection between the technique and the cardiovascular system. Photoplethysmography (PPG) is a non-invasive optical method aimed at the measurement of volume changes of blood in the microvascular bed of the tissue. The working principle is simple: a light-emitting diode (LED) shines on a part of the skin that may usually lie on the fingertip, earlobe, or wrist, and a photodetector detects the amount of incident light either passed through the tissue or reflected [10].

Light passing through the biological tissues is absorbed by changeable elements such as the skeletal system, melanin, and blood. Most importantly, the most intense light-absorbing characteristics are in the blood. Due to the pulsatile flow of blood in the arteries, the amount of blood in the tissue that is being illuminated changes cyclically and is correlated to each heartbeat. This dynamic blood volume regulation results in opposite changes in the light absorption in tissue, and this has been registered in the form of a photoplethysmography (PPG) waveform [5].

The PPG waveform shows two major components. The direct current (DC) component is made up of the dominant, slowly varying component, which is a reflection of the constant baseline tissue, bone, and non-pulsatile venous blood absorption of light. On top of this baseline is a far smaller AC component that is heart-synchronized and explains the pulsatile changes in arterial blood volume. The morphological data that is most relevant to cardiovascular dynamics can be found in the AC component, not the DC component. Despite morphological similarities between the photoplethysmography(PPG) waveform and the arterial blood pressure (ABP) waveform measured by an invasive catheter, the two are different physical quantities: PPG measures the change in blood volume, whereas ABP pressure. However, since the distension of the arterial walls and consequently the blood volume relies precisely on the pressure wave that travels through the arterial tree, PPG is a useful surrogate that contains embedded information about blood pressure [7].

2.2 Literature Review

BP measurement is a critical diagnostic indicator of cardiovascular health. Conventional systems, however, require the use of cuffs, which are usually fitted on the upper arm, and they are intrusive and uncomfortable. Genuine, cuffless methods have hence formed an active research field over the past number of decades. The review follows the history of cuffless blood pressure measurement methodology, distinguishing between three consecutive generations of methodologies: methods based on pulse transit time, traditional machine learning models using hand-designed features, and the latest paradigm based on machine learning.

Table 2.1: Summary of Literature Reviewed.

Author(s)	**Year**	**Title**	**Methodology**	**Key Findings**
Argüello-Prada, E.J. & Castaño Mosquera, C.D. [1]	2024	Exploring supervised machine learning models to estimate blood pressure using non-fiducial features of the photoplethysmogram (PPG) and its derivatives	GPR models combined with ReliefF feature selection using 57 non-fiducial features (statistical indexes, energy operators).	GPR models provided excellent predictions (RMSE < 0.8 mmHg for SBP, < 0.7 mmHg for DBP) using only 15 non-fiducial features, suitable for computationally constrained hardware.
Chakraborty, A. et al. [2]	2020	PPG-based automated estimation of blood pressure using patient-specific neural network modeling	Patient-specific Neural Network (NN) models using 4 selected time-plane PPG features.	Patient-specific approach eliminates need for correction factors, achieving high accuracy (mean error 0.46±2.63 mmHg SBP, 0.15±1.48 mmHg DBP).
Chandrasekhar, A. et al. [3]	2020	PPG Sensor Contact Pressure Should Be Taken Into	Experimental study on 17 subjects varying PPG sensor contact	PAT deviations due to CP changes can translate to BP errors of 11-20 mmHg,

		Account for Cuff-Less Blood Pressure Measurement	pressure (CP) and measuring its impact on Pulse Arrival Time (PAT).	highlighting a critical, often overlooked, confounding variable in cuffless BP measurement.
Dai, D. et al.[4]	2024	Non-Invasive Continuous Blood Pressure Estimation from Single-Channel PPG Based on a Temporal Convolutional Network Integrated with an Attention Mechanism	TCN with Convolutional Block Attention Module (CBAM) on the UCI dataset.	DBP estimation met the AAMI standard (MAE 2.12 mmHg); SBP did not (MAE 5.35 mmHg). TCN-CBAM outperformed other ML/DL models.
Chowdhury, M.H. et al. [5]	2020	Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques	Gaussian Process Regression (GPR) with 107 extracted features (demographic, time/frequency domain, statistical).	GPR with ReliefF feature selection yielded high accuracy (RMSE 6.74 SBP, 3.59 DBP). Complied with BHS Grade B.

Kachuee, M. et al. [6]	2017	Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring	Algorithm based on Pulse Arrival Time (PAT) and other features from vital signals, using regression models.	Complied with the AAMI standard for DBP and MAP. Achieved BHS Grade A for DBP and Grade B for MAP.
Li, Y. et al. [7]	2020	Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model	Deep learning regression models using ECG and PPG for real-time SBP and DBP estimation.	The proposed model outperformed existing methods and achieved accurate estimation.
Li, Z. & He, W. [8]	2021	A Continuous Blood Pressure Estimation Method Using Photoplethysmography by GRNN-Based Model	Generalized Regression Neural Network (GRNN) using harmonics of the raw PPG signal as input.	Fulfilled AAMI and BHS Grade A standards (MAE 3.96 mmHg SBP, 2.39 mmHg DBP). Computationally efficient and can estimate the full BP waveform.

Liu, M. et al. [9]	2017	Cuffless blood pressure estimation based on photoplethysmography signal and its second derivative	SVM model using 14 new features from the second derivative of PPG (SDPPG) combined with 21 conventional features.	Enhanced BP estimation accuracy by 40% compared to Kurylyak et al.'s approach.
Brophy, E. et al. [10]	2021	Estimation of Continuous Blood Pressure from PPG via a Federated Learning Approach	Time-series-to-time-series GAN (T2T-GAN) with a federated learning approach for privacy.	Successfully modeled continuous ABP from PPG. Achieved MAP error of 2.95 mmHg after calibration. Showcased a privacy-preserving, scalable solution.
Tarifi, B. et al. [11]	2023	A Machine Learning Approach to the Non-Invasive Estimation of Continuous Blood Pressure Using Photoplethysmography	Four neural networks for SBP, DBP, MAP, and waveform prediction on MIMIC-III data.	SBP (MAE 5.26 mmHg) and DBP (MAE 2.96 mmHg) results met international standards. Novel separation of discrete value prediction from waveform shape prediction.

Wong, M. K. F. et al. [12]	2023	Applied machine learning for blood pressure estimation using a small, real-world electrocardiogram and photoplethysmogram dataset	Compared RF, SVR, Adaboost, and ANN on a small, noisy dataset (n=54) using ECG+PPG features.	SVR performed best on noisy data (MAE 6.97 mmHg), meeting BHS Level C criteria. Demonstrated feasibility on real-world ambulatory data.
Mohammadi, H. et al. [13]	2025	Cuff-less blood pressure monitoring via PPG signals using a hybrid CNN-BiLSTM deep learning model with an attention mechanism	Hybrid CNN-BiLSTM with attention mechanism on a large MIMIC-II dataset (2064 patients).	Achieved very high accuracy with MAE of 1.88 mmHg for SBP and 1.34 mmHg for DBP, underscoring potential for scalable wearable solutions.
Wang, W. et al. [14]	2021	Cuff-Less Blood Pressure Estimation via Small Convolutional Neural Networks	Modified LeNet-5 CNN trained on images created from PPG segments using Visibility Graph (VG).	Achieved Grade A under BHS protocol for both SBP (ME 0.18±7.46 mmHg) and DBP (ME 0.34±4.07 mmHg).

Slapničar, G. et al. [15]	2019	Blood Pressure Estimation from Photoplethysmogram Using a Spectro-Temporal Deep Neural Network	Custom ResNet architecture using 5s PPG waveform and its derivatives as input.	MAEs of 9.4 mmHg (SBP) and 6.9 mmHg (DBP) did not meet clinical standards, but the work was a valuable guide due to its large dataset (510 patients) and detailed methodology.
Kurylyak, Y. et al. [16]	2013	A Neural Network-based method for continuous blood pressure estimation from a PPG signal	An Artificial Neural Network (ANN) was trained on 21 manually extracted parameters from PPG signals in the MIMIC database.	Proved more precise than linear regression methods, achieving MAEs of 3.8 mmHg (SBP) and 2.2 mmHg (DBP).
Xing, X. & Sun, M. [17]	2016	Optical blood pressure estimation with photoplethysmography and FFT-based neural networks	ANN trained on FFT-derived features (amplitudes, phases) from normalized PPG signals.	Fulfilled BHS protocol, ranking Grade A/A. Fast and robust method using only PPG.

Dey, J. et al. [18]	2018	InstaBP: Cuff-less blood pressure monitoring on a smartphone using a single PPG sensor	Ensemble of prediction models based on demographic/physiological partitioning, using unique PPG features on a smartphone.	Achieved MAE of 5 mmHg (DBP) and 6.9 mmHg (SBP), an 18% and 11.5% improvement over ubiquitous models, respectively.
Tjahjadi, H. et al. [19]	2020	Noninvasive Classification of Blood Pressure Based on Photoplethysmography Signals Using Bidirectional Long Short-Term Memory and Time-Frequency Analysis	BLSTM network trained on time-frequency features (instantaneous frequency, spectral entropy) from PPG for BP classification.	Achieved high classification accuracy (97.33%), sensitivity (100%), and specificity (94.87%), outperforming CNN, KNN, and other methods.
Chen, S. et al. [20]	2019	A Non-Invasive Continuous Blood Pressure Estimation Approach Based on Machine Learning	SVR model optimized with a Genetic Algorithm (GA), using 14 features from ECG and PPG (including PTT and waveform characteristics)	High accuracy (error 3.27±5.52 mmHg SBP, 1.16±1.97 mmHg DBP), meeting AAMI and BHS Grade A criteria.

Liu, Z. et al. [21]	2020	Continuous Blood Pressure Estimation From Electrocardiogram and Photoplethysmogram During Arrhythmias	Random Forest Regression (RFR) model using 15 features from ECG and PPG to estimate BP during arrhythmias.	RFR model performed best, meeting AAMI and BHS Grade A standards even in arrhythmia patients (ME -0.04±6.11 SBP, 0.11±3.62 DBP).
Zadi, A.S. et al. [22]	2018	Arterial Blood Pressure Feature Estimation Using Photoplethysmography	Fifth-order autoregressive moving average (ARMA) models to relate PPG peaks/troughs to SBP/DBP.	ARMA models effectively tracked BP trends with low model errors (<5 mmHg during normal breathing, <8 mmHg during breath-hold), comparable to PTT-based methods.

2.3 Gap Analysis

A critical literature review indicates that the examined literature has identified a number of challenges and gaps that cannot be ignored in order to facilitate the implementation of cuffless blood pressure monitoring technologies in clinical practice. The proposed project is mainly concerned with addressing these gaps.

Table 2.2: Gap Analysis Table

Features	**Argüello-Prada & Castaño [1]**	**Dai et al. [4]**	**Chowdhury et al. [5]**	**Mohammadi et al. [13]**	**Montesinos et al. [23]**	**Proposed System**
Calibration-free operation	No	No	No	No	No	Yes
Standardized validation framework	No	No	No	No	No (review highlights lack)	Yes
Robustness to motion artifacts	No	No	No	No (trained on ICU-style data)	No	Yes
Use of demographic data	No	No	Yes (age, gender used)	No	Yes (review suggests need)	Yes (Height, Weight, Age, Gender, BMI)
Security & privacy mechanisms	No	No	No	No	No	Yes (HIPAA/GDPR)

The apparent "accuracy arms race" seen in academic literature, where studies report incrementally better performance metrics, often masks these more fundamental issues. It can be misleading when a slightly smaller Mean Absolute Error (MAE) is pursued on a benchmark dataset when the methodology is not robustly validated or capable of being applied to the real world. The real problem is not only to obtain high accuracy under controlled conditions, but also to create a system that is reliable, safe, and clinically meaningful to various individuals in their everyday lives. This necessitates moving away from an algorithmic optimization-only approach towards a broader, systems-level approach that emphasizes strong validation, artifact robustness, and regulatory compliance

2.4 Summary

In the context of the background investigation, it is clear that the area of PPG-based blood pressure estimation is an active one with a very fast pace of development. The technological development of the dual-sensor PTT approaches to single-sensor, feature-based machine learning and, finally, an end-to-end deep learning are clearly evolving in the more sustainable and viable direction. The connection to IoT platforms also has the potential to bring these algorithmic breakthroughs to scalable, at-a-distance health surveillance. Nevertheless, this promise is checked by serious and continuing challenges. The dependence on calibration, absence of uniform validation procedures, susceptibility to motion artifacts and lack of connection with the larger physiological picture are among the greatest areas of research deficiency and clinical application. The proposed project aims to fill these gaps in the presentation of a comprehensive system approach that is not only founded on a state-of-the-art AI model, but is also rooted in the principles of rigorous evaluation, regulatory compliance, and secure system design

Chapter 3

Research Methodology

This chapter explains in detail the approach used to develop a cuffless blood pressure prediction approach based on IoT-based photoplethysmography (PPG) signal processing and machine learning models. It includes preprocessing of raw data, feature extraction, architecture of several machine learning and deep learning algorithms, and justification for selecting and comparing several modeling methodologies.

3.1 Methodology

3.1.1 Overview

The methodology used to develop a non-invasive blood pressure prediction system is proposed and it consists of several parts: IoT device, web interface, backend system and machine learning models. This methodology allows for prediction of blood pressure in real time according to photoplethysmography (PPG) signals, which are processed and analyzed in a sequence of well-defined steps [5][25].

In this paper, the IoT device, developed and programmed in C++, is used as the data acquisition unit, it is the interface between sensors (MAX30102) that sends PPG signals. Different hardware-based methods based on optical sensors have been validated in several studies for cuffless BP measurement [18][28][30]. The device is connected to the web system and sends the data it collects for further processing. The information is then processed through a web interface, which is created using HTML, CSS and JavaScript. Through this interface, users enter the demographic information (e.g., age, gender, height, and weight) and get the predicted blood pressure based on the trained machine learning models, which are related to multi-modal feature fusion strategies in literature [5][24].

The back-end application is written in Node.js technology and provides seamless communication between IoT device and Web interface. It helps the data to pass from the device to the website where the data is stored and processed. In addition, the backend also handles the interface between the inputs from the user and the predictive models, a common model in cloud-edge health monitoring schemes [27][30].

Once the data has been collected, it is passed through exception handling for the purpose of handling the errors that may occur during data collection, transmission, or processing. After this, the data is preprocessed, which means cleaning the data and preparing it for the next stages of the analysis. Statistical features such as skewness, kurtosis, and the number of peaks in the PPG signal are extracted during the feature extraction phase as these statistical features have exhibited their importance in predicting blood pressure [1][9][21]. Frequency-domain features derived from discrete Fourier transform (FFT)-based energy in different bands were also extracted, in line with previous research on the spectral basis of hemodynamic variability [4][15].

The processed data is then split into two portions, training data and test data. The machine learning models are trained using the training data while the test data is used for model evaluation. Similar validation process has been applied to GPM studies for BP estimation targeting avoidance of overfitting and for generalizability purpose.[2][20] Linear regression, Random forest regressor, ANN, SVR, KNN, Ada Boost Regressor, and Bidirectional Long Short Term Memory (BiLSTM) models are trained in Python on Google Colab. Each of the models is tested using commonly used regression metrics including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R2 score, to find the optimal model [10][11][13].

Once the best model is selected, it is implemented via the backend system, where it can be used to make real-time predictions of blood pressure based on new inputs from a user. The final prediction is presented in the web interface, from which users receive the predictive result of their blood pressure readings immediately [8][14]. This solution brings together hardware, software, and machine learning to produce a seamless platform for noninvasive and accurate blood pressure assessment.

This way, not only an efficient workflow is guaranteed from the data collection to the model deployment, but a modern best practice in IoT, web development, and machine learning, offering an effective and user-friendly solution for real-time health monitoring [23][26].

3.1.2 Proposed Methodology

The methodology used to develop a non-invasive blood pressure prediction system is proposed and it consists of several parts: IoT device, web interface, backend system and machine learning models. This methodology allows for prediction of blood pressure in real time according to photoplethysmography (PPG) signals, which are processed and analyzed in a sequence of well-defined steps [5][25].

In this paper, the IoT device, developed and programmed in C++, is used as the data acquisition unit, it is the interface between sensors (MAX30102) that sends PPG signals. Different hardware-based methods based on optical sensors have been validated in several studies for cuffless BP measurement.[8][28][30] The device is connected to the web system and sends the data it collects for further processing. The information is then processed through a web interface, which is created using HTML, CSS and JavaScript. Through this interface, users enter the demographic information (e.g., age, gender, height, and weight) and get the predicted blood pressure based on the trained machine learning models, which is related to multi-modal feature fusion strategies in literature [6][24].

The back-end application is written in Node.js technology and provides seamless communication between IoT device and Web interface. It helps the data to pass from the device to the website where the data is stored and processed. In addition, the backend also handles the interface between the inputs from the user and the predictive models, a common model in cloud-edge health monitoring schemes[27][30].

Once the data has been collected, it is passed through exception handling for the purpose of handling the errors that may occur during data collection, transmission, or processing. After this, the data is preprocessed, which means cleaning the data and preparing it for the next stages of the analysis. Statistical features such as skewness, kurtosis, and the number of peaks in the PPG signal are extracted during the feature extraction phase as these statistical features have exhibited their importance in predicting blood pressure. [1][9][21] Frequency-domain features derived from discrete Fourier transform (FFT)-based energy in different bands were also extracted, in line with previous research on the spectral basis of hemodynamic variability [4][15].

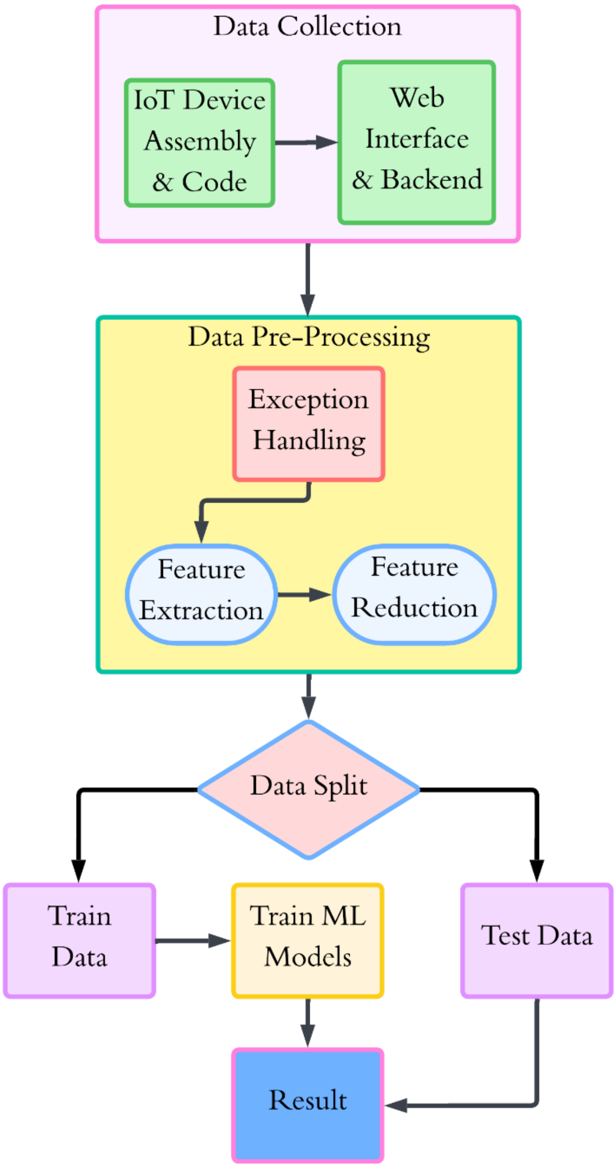


Figure 3.1: Proposed Methodology Diagram

The processed data is then split into two portions, training data and test data. The machine learning models are trained using the training data while the test data is used for model evaluation. Similar validation process has been applied to GPM studies for BP estimation targeting avoidance of overfitting and for generalizability purpose.[2][20] Linear regression, Random forest regressor, ANN, SVR, KNN, Ada Boost Regressor, and Bidirectional Long Short Term Memory (BiLSTM) models are trained in Python on Google Colab. Each of the models is tested using commonly used regression metrics including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R2 score, to find the optimal model [2][11][13].

Once the best model is selected, it is implemented via the backend system, where it can be used to make real-time predictions of blood pressure based on new inputs from a user. The final prediction is presented in the web interface, from which users receive the predictive result of their blood pressure readings immediately [8][14]. This solution brings together hardware, software, and machine learning to produce a seamless platform for noninvasive and accurate blood pressure assessment.

This way, not only an efficient workflow is guaranteed from the data collection to the model deployment, but a modern best practice in IoT, web development, and machine learning, offering an effective and user-friendly solution for real-time health monitoring [23][26].

3.2 Detailed Methodology and Design

3.2.1 Data Preprocessing, Features Extraction and Selection

The sensor provided raw PPG as samples of the waveforms on the infrared (IR) and red light channels. Approximately 409 time-series columns of data of the variability of changes in blood volumes across time were obtained by each measurement session. Similar to the majority of physiological signals, raw PPG was affected by motion artifact, ambient light artifact, and baseline wandering noise, which masked a hemodynamic signal. Therefore, much preprocessing was needed to use the data in machine learning models [15][25].

Data cleaning was the initial step of the preprocessing pipeline. Rows containing a place value error (e.g., -999 in heart rate or SpO2), rows with unfinished or damaged sensor data were filtered out. Label encoding was used in order to categorize categorical variables like gender numerically so that they can fit regression models.

After cleaning, some simple descriptors were obtained to give an overview of the raw waveforms. These were statistical characteristics of the mean, median, standard deviation, minimum value, maximum value, skewness, and kurtosis. Peak detection (e.g., number of peaks, mean peak amplitude) was also used to derive morphological information because it measures changes in heart rate and arterial stiffness [4]. Fast Fourier Transform (FFT) was used in the frequency domain to extract the most important frequency, total spectral energy, and band-limited powers in low-frequency (0.5-3 Hz) and high-frequency (3-6 Hz) bands [17]. Firstly, IR and red channels were examined but the red-channel characteristics and SpO2 values did not add any predictive values and were not used in the final dataset.

In order to improve the prediction power, a wider range of superior attributes was developed, which led to 76 descriptors per segment:

I. Morphological Characteristics: Amplitude, 25% and 75% width, rise and fall time, slopes, dicrotic amplitude and timing, area under the curve (AUC) and ratio rise and fall. They record compliance and pulse shape variations of BP [36][37].

II. Features of Heart Rate Variability (HRV): Inter-beat intervals (IBI), SDNN, RMSSD, and Poincare plot descriptors (SD1, SD2), which characterize autonomic control of vascular tone [38].

III. Spectral Features: Spectral centroid, spectral entropy, low-frequency (0.04-0.15 Hz)-spectral energy (associated with sympathetic activity) and high-frequency (0.15-0.40 Hz)-spectral energy (associated with parasympathetic activity).

IV. Entropy and Complexity Characteristics: Shannon entropy, sample entropy, approximate entropy, Hjorth parameters (activity, mobility, complexity), detrended fluctuation analysis (DFA), and Petrosian fractal dimension, which are measures of irregularity and self-similarity in PPG signals [39].

V. Signal Quality Metrics: Signal to noise ratio (SNR), valid beat percentage, detected peaks and feet, and quality indices, making it tough to be affected by motion artifacts [40].

In total 76 signal characteristics were obtained and after they were combined with demographic data (age, gender, height, weight, BMI) and vital signs (heart rate, SpO2), the data had 85 variables per observation.

Lastly, feature selection was done because of the imbalance in dataset size (536 records) and feature count. Nonlinear dependencies were ranked using Mutual Information (MI), Recursive Feature Elimination (RFE) created with LightGBM sequentially removed redundant features, and SHAP analysis was used to identify physiologically relevant predictors. This made sure that the model training data used in the end was based on strong interpretable and non-redundant features.

3.2.2 Models for Blood Pressure Prediction.

Linear Regression

Linear regression was used as a benchmark model, because it is readily interpretable and mathematically simples.[32] The approach approximates blood pressure as linear relation of input features:

$$\hat{y} = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \tag{1}$$

In which $\hat{y}$ the parameter that predicts BP, x_i are features based on PPG signals and demographics and β_i are constants that have been learned by minimizing the squared error. Although linear regression will hardly be able to address the nonlinearities of cardiovascular dynamics, it allows determining a new compelling point against which to compare the incremental improvements made by more advanced models.

Random Forest Regression

Random Forests (RFs) is a type of ensemble approach, which builds several decision trees throughout training and gives an average of all decision trees.[31] All the trees divide the input space into smaller parts through recursive subdivisions based on the features that achieve highest degree of variance reduction.

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}_{tree_i} \tag{2}$$

Thresholds on IR spectral energy or BMI can then be used to separate the feature space into subpopulations having different BP, e.g. as the training data used may have biases toward social media usage. RF averages predicted values across the large number of trees, which reduces the risk of overfitting individual relationships and is tolerant to noisy physiological predictors. Previous research has demonstrated that RFs can better predict PPG-based BP estimation since they can support nonlinear dependencies and heterogeneous contributions of a feature [11].

Gradient Boosted Trees (XGBoost)

Gradient Boosting and, in particular, the XGBoost algorithm enhance the quality of ensemble algorithms by creating trees in sequence such that each tree tries to regularize the mistakes that the ensemble has made so far [33]. According to the model the BP values can be predicted as:

$$\hat{y} = \sum_{m=1}^{M} f_m(x), \quad f_m \in \mathcal{F} \tag{3}$$

where f_m are regression trees. Regularization conditions do not favor very complex models, so the models can be generalized to different patients. Due to its capability to acknowledge both decipherability and high predictive capacity at the same time, as well as its adaptation to considerable contemporary datasets, including those provided by IoT sensor[23]. XGBoost has become a popular tool in the biomedical signal modelling domain.

Support Vector Regression

Support Vector Regression (SVR) is an application of support vector machines as used in regression.[34] The SVR tries to approximate a fitting of a function that predicts blood pressure and lets be deviate by tolerance epsilon. The fitness is the minimization of the weight vector norm $\|w\|^2$ under:

$$| y_i - (w \cdot \phi(x_i) + b) | \leq \epsilon \tag{4}$$

where $\phi(x)$ projects features in a high dimension space through a kernel function. The radial basis function (RBF) kernel is especially well adapted for PPG signals, to the extent that it approximates nonlinear relationships across features. Past studies have already shown that SVR is indeed the appropriate approach in estimating the continuous BP that utilizes the features of PPG.[29]

K-Nearest Neighbors Regression

The KNN regression estimates the BP by averaging the values of the kkk nearest training examples in video space. Similarity was measured based on Euclidean distance between IR derived features and demographics. When X, O is known, Y is unknown: XO = YO.

Euclidean Distance equation:

$$d(x, y) = \sqrt{(\sum_{i=1}^{n} (x_i - y_i)^2)} \tag{5}$$

For predicting values like blood pressure, KNN performs regression using the following equation:

$$\hat{y} = \frac{1}{K} \sum_{i=1}^{K} y_i \tag{6}$$

KNN is more computationally expensive at inference time, but can utilize local structure within the data and can be used when the sample size is more than relatively small, as in this example. It has been used in small datasets to provide physiological signal regression results in areas where local neighborhoods capture individual variability[9].

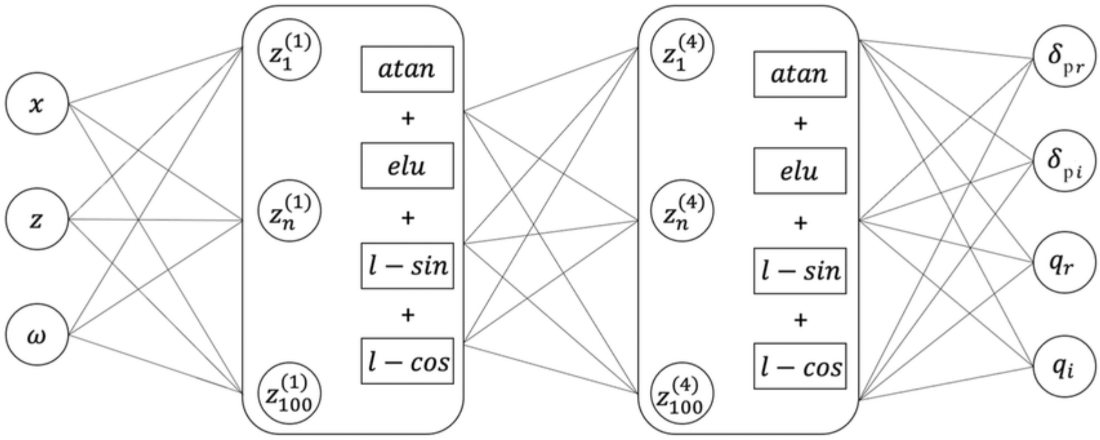


Figure 3.3: Illustration of KNN Architecture.

Artificial Neural Networks

Artificial Neural Networks ANNs) were used as multilayer perceptrons (MLPs) that had 64-128 neurons in two to three hidden layers and one output layer generating systolic and diastolic forecasts. The nonlinear transformations are used on each of the layers:

$$a^l = f(W^l a^{l-1} + b^l) \tag{7}$$

and in the case of the ReLU activation function, $f(\cdot)$ is generally. ANNs are also optimal to elicit nonlinear relationships between PPG morphology and BP, having been confirmed by the previous experiments predicting patient-specific BP using PPG[2]. In other architecture dropout layers were used to help discourage overfitting.

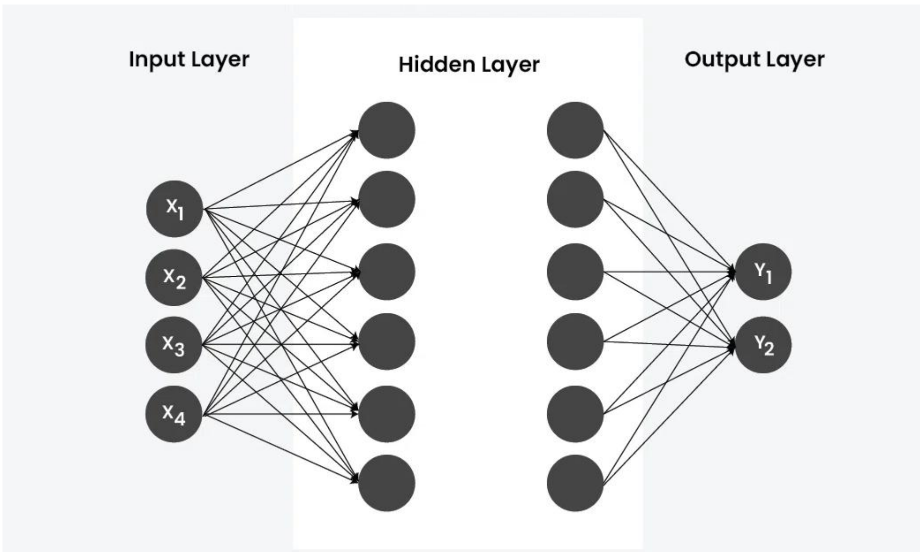


Figure 3.3: Illustration of ANN with dense layers Architecture.

Bidirectional Long Short-Term Memory Networks

As the PPG signals are inherently temporal signals, recurrent neural architectures were another option to consider as well. The Long Short-Term Memory (LSTM) nets learn long-term relationships across time series within a time sequence by storing memory cells controlled by input, forget, and output gates [35]. The BiLSTM version works on the sequence in two directions forward and backward, allowing a model to capture the context of the entire waveform. To compute the hidden state update, x_t will be used as input sequence, where:

$$C_t = f_t \odot C_{t-1} + i_t \odot \underline{C}_t, h_t = o_t \odot tanh(C_t) \qquad (8)$$

where it, f_t, o_t i_t , are input, forget and output gates, respectively. BiLSTMs have been shown to provide state estimates of cuffless BP using signal transitions that contain the dynamics of the vascular response in PPG signals.[13][19]

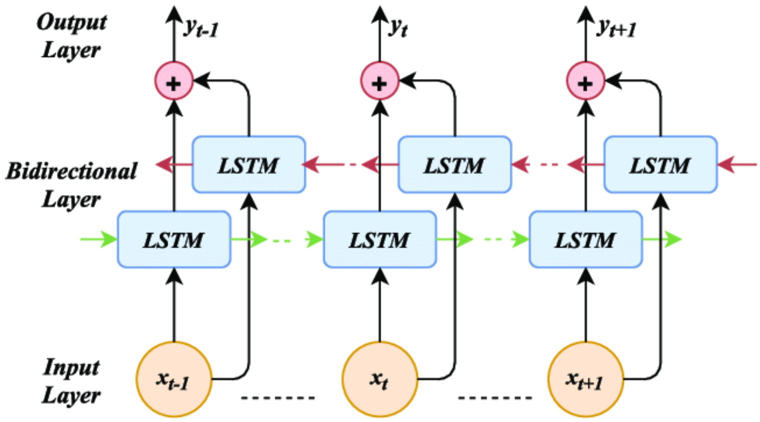


Figure 3.2: Illustration of BiLSTM Architecture.

AdaBoost Regression

AdaBoost (Adaptive Boosting) is an ensemble learning algorithm that builds a strong learning model with the combination of multiple weak models (usually decision trees). Its use in regression would mean successively adding weak learners to the data and altering their weights to concentrate at misclassified data.

This is the final prediction of regression: a weighted average of all weak-learners gets predictions:

$$\hat{y} = \sum_{t=1}^{T} \alpha_t \cdot \hat{y}_t \qquad (9)$$

Where $\hat{y}_t$ is the prediction of the t-th weak learner, and **T** is the number of iterations.

AdaBoost Regression is often applied to decision tree splits (decision stump) because those are weak learners, but any simple regression model can also be used as the weak learner. The decision tree models in AdaBoost are shallow (usually of a depth no more than 1), that is, the algorithm produces many weak decision models that can be cumulatively refined.

CatBoost Model

CatBoost (Categorical Boosting) is an algorithm of gradient boosting presented by Dorogush et al. in 2018 [41]. It can be especially effective with small-to-medium datasets that contain mixed types of features, which is why it is applicable to biomedical signals. In contrast to other more traditional boosting algorithms (e.g., XGBoost, LightGBM), CatBoost uses ordered boosting to avoid target leakage and oblivious decision trees as base learners.

The split criterion at any level of an oblivious tree is the same, and thus the tree is symmetric and both efficient to train and infer. The loss function CatBoost minimizes is given a specific dataset x_i, y_i:

$$L = \sum_{i=1}^{N} \ell(y_i, F_{m-1}(x_i) + \eta \cdot h_m(x_i)) + \lambda \|h_m\|^2 \quad (10)$$

The ordered boosting technique guarantees that during each iteration, the CatBoost is trained on a permuted data set in which subsequent data entries would not affect previous forecasts. This mitigates overfitting which is paramount when dealing with small noisy data including PPG-derived features.

CatBoost is currently being used more in the biomedical and healthcare field of data analysis because it is highly generalized, converges quickly, and resists overfitting [41][42].

3.2.3 Hyperparameter Tuning

The hyperparameter tuning was carried out to maximize the model performance. With the limited size of dataset, the tuning process was limited to the most significant parameters, and overfitting was prevented by cross-validation.

Random Forest (RF) Tuning Hyperparameters

In the case of Random Forest, the parameters that were tuned were the following:

I. Number of Trees (n_estimators): 100-1000 were tested.

II. Maximum Depth (max_depth): This is limited to 3-20 to balance bias-variance.

III. Minimum Samples Split and Leaf: Modified to regulate overfitting.

IV. Maximum Features: $\sqrt{p}$ and $p/3$, where p is the number of features.

The best configuration was found through grid search with 5-fold cross-validation.

CatBoost Tuning Hyperparameters

In case of CatBoost, the following are the most important hyperparameters that were optimized:

I. Iterations (n_estimators): 500-2000, the size of the ensemble can be controlled.

II. Learning Rate (η): 0.01-0.3, between speed and stability.

III. Maximum Depth (depth): 4-10, which influences the complexity of the model.

IV. L2 Regularization (l2_leaf_reg): 1-10, to minimize overfitting.

V. Subsample: 0.6-1.0, to add randomness to improve generalization.

Randomized search with early termination (patience of 50 rounds) was used by us to prevent overtraining on noisy data. The CatBoost version was tuned in a way that provided significant stability over baseline models, and thus it is appropriate in this application.

3.2.3 Justification for Model Diversity

The complexity of the underlying physiological task is reflected in the fact that a large variety of models are tested. Nonlinear responses pertaining to vascular resistance, arterial compliance, and cardiac output act upon blood pressure, all of which are indirectly represented on the PPG waveform.[26] There is no single model family that has received universal approval as being best in this regard. Ensemble tree strains provide interpretability and predictability, kernel-based approaches provide nonlinear mapping flexibility, skeletal neural structures more effectively learn hidden temporal and structural structures. An overlay of these paradigms provides the methodological rigor of assuring highest probability of an architecture that can be generalized through subjects.

3.3 Project Plan

3.3.1 Task Allocation

This table depicts the timeline of the principal activities in each period of the project, from week 1 to week 37.

Table 3.1: Task Allocation Timeline

Tasks	Weeks																		
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
Literature Review & Problem Definition																			
IoT Device Assembly																			
IoT Device Code																			
Frontend Develop																			
Backend Develop																			
Ensure Data Flow																			
Data Collection																			
Data Preprocessing																			
AI Model Development																			
AI Model Deployment																			
Comparison Optimization, and Final Report Writing																			

The given table of task allocation timeline is the elaborated schedule of a project completion with 37 weeks and most important activities on its vertical axis. The initial steps of the project are the Literature Review and Problem Definition (weeks 1-5) and after that is IoT Device Assembly (weeks 5-13) that is composed of hardware assembly, as well as, coding. Week 11: Over the next weeks, the frontend is developed, it runs in parallel with APIs and backend, which will start in the paper week 13 and will be used to create APIs and to work through data flow (weeks 17-21). The period during which data collection process will occur is protracted and will cover the duration of weeks 17-29 where enough time will be available to collect and process data required by the machine learning models. The development of AI models begins in week 21 and extends to week 29 and then model deployment follows between week 29-33. In the last, the period between weeks 33-37 requires comparison, optimization, and scoring the final report, a process that manages results in refining and proper documentation. Such an organization enables the development to be mapped in parallel without excluding hardware, software, and machine learning undertakings hence flexibility in making adjustments in the whole project.

3.4 Summary

this research paper discusses the approach followed towards the creation of a non-invasive blood pressure (BP) monitoring system on the basis of an IoT-driven platform and machine learning models. The chapter starts with the description of the data collection procedure, in which physiological data had been collected with the use of a maximal of three data sensors, namely a MAX30102 PPG sensor with an ESP32 microcontroller. This sensor captured PPG signals and this was sent/transmitted through an IoT protocol to be processed and analyzed. This is followed by the discussion of the data preprocessing whereby the raw PPG data was preprocessed and irrelevant noise eliminated to make features be extracted more accurately. The PPG signals yielded key features, such as heart rate and SpO2, to be enabled to be used in model training. Another aspect that the methodology identifies is machine learning models that had been used in prediction of blood pressure such as Random Forest, AdaBoost, XGBoost, and BiLSTM. The data was preprocessed, and the models were trained and hyperparameters were tuned to order to maximize their performance. Model accuracy was assessed using various performance indexes as Mean Absolute Error (MAE), Root Same Square Error (RMSE), R-squared, and per cent errors (Mean Absolute Percentage error or (MAPE). Moreover, the chapter describes how ensemble techniques may be used to enhance the accuracy of predictions, through incorporating the strengths of several models. All the final models were considered to see their ability of generalization but mainly to improve their performance in real world cases, especially in wearable and portable litigating, BP. Finally, this chapter defines a holistic approach to combining IoT-collected data with machine learning-trained models with their performance assessment to develop a non-invasive and precise system of BP monitoring.

Chapter 4

Implementation and Results

This chapter presents the implementation of the IoT–AI blood pressure prediction system and provides a detailed analysis of the results. It begins with an overview of the environment setup for both IoT hardware and software, followed by a comparative analysis of machine learning and deep learning models.

4.1 Environment Setup

4.1.1 IoT Device Environment

The IoT board was created based on the ESP32 NodeMCU V1.3 development board, built in with a photoplethysmography (PPG) sensor, the MAX30102. Testing In early design experiments the device could be powered directly, with any power bank, via a USB-B connection, which served as a stable and convenient power source and did not need explicit circuitry. This arrangement minimized the work required to debug and provided continuous operation when testing data collection. After stability was achieved, the connections to power supply were changed to the rechargeable 18650 lithium-ion battery having its own holder and Type-C fast-charging module, enabling the device to be pushed into the acrylic enclosure to make it portable. The real-time monitoring and control of the device was carried out using 16x2 LCD display module with I2C adapter and push-button interface. Using WiFiManager library, the ESP32 was able to reach local Wi-Fi networks in real-time, allowing it to transmit data without hardcoding any credentials. Such an environment of flexible hardware provided a level of adaptability even within various testing conditions.

4.1.2 Web and Backend Environment

The web-based and backend infrastructure was deployed in a Node.js server with a connection to a MongoDB Atlas cluster. The backend supported a variety of API endpoints to receive incoming sensor data and store it in the ‘raw_data’ collection, then enrich with demographic information (age, gender, height, weight, BMI) and blood pressure records to add it to the ‘total_data’ collection. The backend was launched to Vercel which uses built-in encryption (TLS/HTTPS) to communicate securely. Vercel was also used to deploy the frontend web interface as another application to enable users to send inputs to the system. This normalization of the server and the client allowed the application to scale and maintain. In the deployment, the standards of RESTful API have been used to ensure interoperability between the components. MongoDB data segregation enabled effective dataset preparation to train a model, as well as live data integration with the MongoDB prediction API. It was a web environment with a strong backbone to facilitate communication between IoT and cloud and the web.

4.1.3 Model Development Environment

Google Colab Pro was the main training and evaluation environment selected, with the benefit of long runtime visibility and increased resource limit, including the use of a virtual or physical GPU based on the application. Popular Python libraries like scikit-learn, TensorFlow, Keras, and XGBoost were used to implement and run several regression and deep learning models, which later were tested in the environment. The preprocessing of data and visualization were conducted with pandas, NumPy, seaborn, and matplotlib. Besides cloud-based resources, Dell Inspiron 15 5510 laptop using Intel Core i7-11390H processor (8-core), 16 GB RAM, and Windows 11 home 64-bit was used to develop it primarily as a local resource. This laptop assisted in code writing in the back-end as well as integrating the websites along with verifying the initial data sets. The message-passing workflow between the compute-intensive parts, which was performed on the dual-environment workflow, Colab, and the local machine, the local machine provided efficiency in both experimentation and deployment. In combination with the hardware of the IoT and with a web system hosted by Vercel, this development setup at the model made the basis of the computations of the project

4.1.4 Environment Workflow

In this section the whole workflow of the system is shown and discussed in detail.

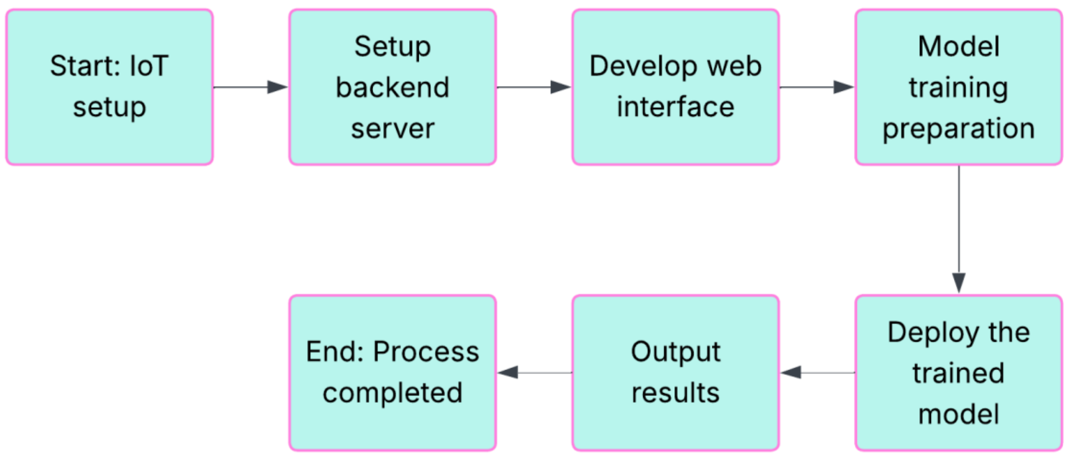


Figure 4.1: Workflow of IoT-AI Blood Pressure Prediction System.

The figure shows a flow chart of the steps undertaken sequentially during the implementation of IoT-AI based blood pressure prediction system. This will start with the solution of IoT where the ESP32 and MAX30102 sensor will be programmed to collect data. The second step after testing the hardware operational and finding that it functions performs the task of the backend server, which provides two-way communication between the IoT device, database, and web application. This is then balanced by development of the web interface, here users can add demographic information and receive system predictions.

Once the system infrastructure is in place, the process proceeds to model training preparation as defined by taking raw physiological data, processing it (i.e. preprocessing), preparing it to be used in machine learning activities. The trained model is subsequently deployed along the backend API, and it can be inferred in real time. The system then moves to the output results phase where the user can read the systolic and diastolic blood pressure as per their prediction through the web interface on the display. Lastly, the workflow ends with the end of a process and completes the circle between the acquisition of the IoT data, backend processing, AI prediction and end-user feedback.

4.2 Comparative Analysis

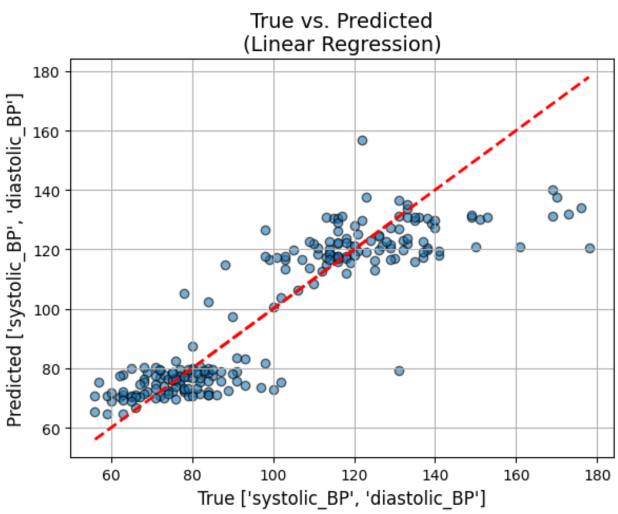
This part of the report summarizes and compares performance of diverse machine learning models that are used to predict systolic and diastolic blood pressure (BP). The training and testing data includes 536 samples: 18 X variables, and 2 target variables (systolic and diastolic BP). The data has been separated into a 80 percentage training set or 20 percent testing set. The analyzed models are Linear Regression, Random Forest, XGBoost, Artificial Neural Network (ANN)., BiLSTM., AdaBoost Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) Regression. Google Colab with various Python libraries specifically for ML and DL models building was used to train the models and analyze the results on important criteria such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), R2 and Mean Absolute Percentage Error (MAPE). Each model is analyzed in the following subsections in connection to several performance visualizations such as scatter plots, residual plots, and accuracy curves (Error CDF).

4.2.1 Linear Regression

Some of the least complex models that are utilized in regression tasks include Linear Regression. It supposes that the features (similar to independent variable) and the target (similar to dependent variable) vary in a linear manner. Systolic BP and diastolic BP are the target variables in this case.

True vs. Predicted (Linear Regression):

Figure 4.2: Ture vs Predicted(Linear Regression)



True vs. Predicted scatter plot Linear regression demonstrates a good fit to most of the paradigms particularly the ones at the lower range of systolic and diastolic BP. As the values rise however, the forecasts begin to lose contact with the actual values, meaning that the model is not able to handle such high BP values. The red dotted line is the perfect case in which the predicted values and the true values occur to the same level.

Residuals Plot (Linear Regression):

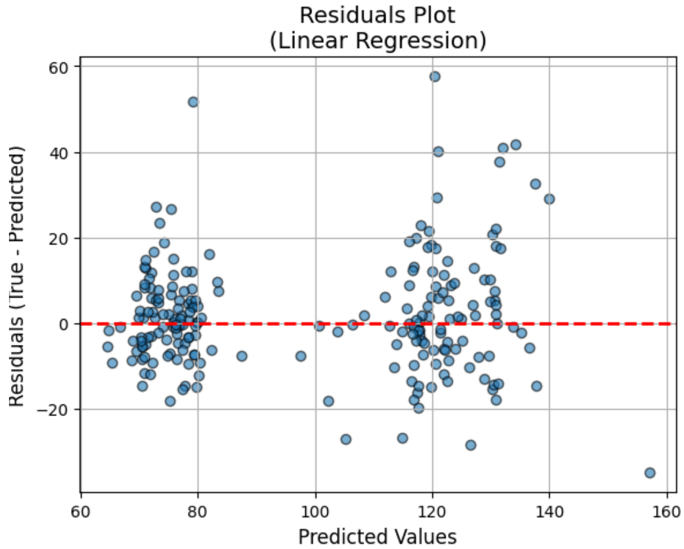


Figure 4.3: Residuals Plot of Linear Regression

The Linear Regression plot of the Residual will indicate that the residual values (the difference between the actual and predicted values) are distributed fairly around zero with no apparent patterns. This is an indication that there is no massive bias and/or variance on this model. There is, however, a bit of clustering of residuals at some points and this could accurately indicate overfitting of the data within particular ranges.

Accuracy Curve (Error CDF):

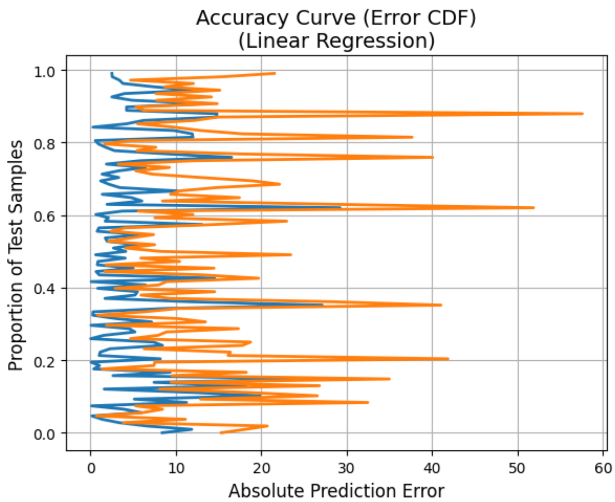


Figure 4.4: Accuracy Curve of Linear Regression

Accuracy Curve Linear Regression demonstrates that the error of prediction made by the model distributed widely, and there is a significant dispersion between the error of prediction. The error percentage of a considerable fraction of the test samples is greater than the error of more sophisticated models such as Random Forest and XGBoost. This can be seen in the MAE and RMSE values in the last model comparison table which are 9.39 and 13.25 respectively.

Performance Analysis:

Linear Regression works fairly well with lower systolic and diastolic BP but not as BP rises. The large dispersion along the Accuracy Curve and greater dispersion in the higher range of the Accuracy Curve indicate that the model performs poorly in extreme BP scales. The MAE and RMSE are 9.40 and 13.26 respectively, which means that the model makes a moderate prediction error, especially when high values of BP are used. Linear Regression is also subject to underfitting in cases where the correlation between features and target values are not linear. The weakness of the model to deal with outliers or data pattern causes more errors in prediction to occur as BP grows. As a result more sophisticated models, including Random Forest or XGBoost are able to better represent these complexities.

Results:

- MAE: 9.39
- MSE: 175.70
- RMSE: 13.25
- R^2: 0.19
- MAPE: 9.17%

4.2.2 Random Forest (Tuned)

Random Forest is an ensemble-based learning technique that develops multiple decision trees and integrates them with each other to enhance accuracy and minimize overfitting. Random Forest is a better model of non-linear relationships than Linear Regression and better at avoiding overfitting than separate decision trees. For this study, the Random forest model was tuned to provide a better result.

True vs. Predicted (Random Forest):

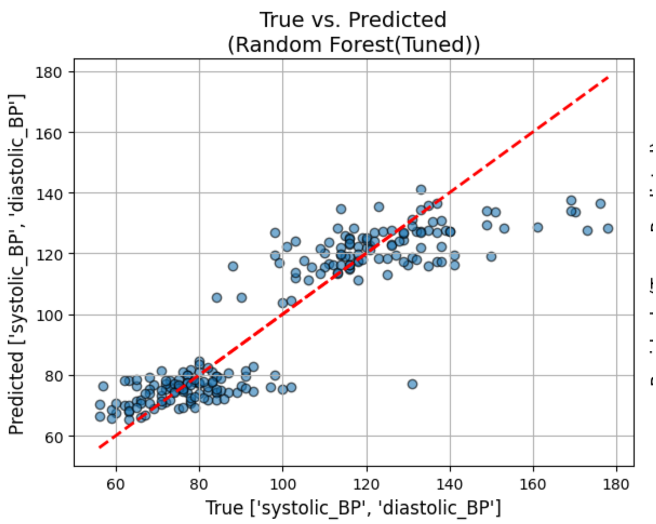


Figure 4.5: Ture vs Predicted (Random Forest Tuned)

Merely a glance at the True vs. Predicted plot of the Random Forest reveals a far more accurate fit than that of the Linear Regression with most predictions closely matching the true values. The model treats the larger values of BP even better than the Linear Regression model because the points are closer to the red dashed line, hence a good correlation between the expected or actual systolic/diastolic BP values obtained.

Residuals plot (Random Forest):

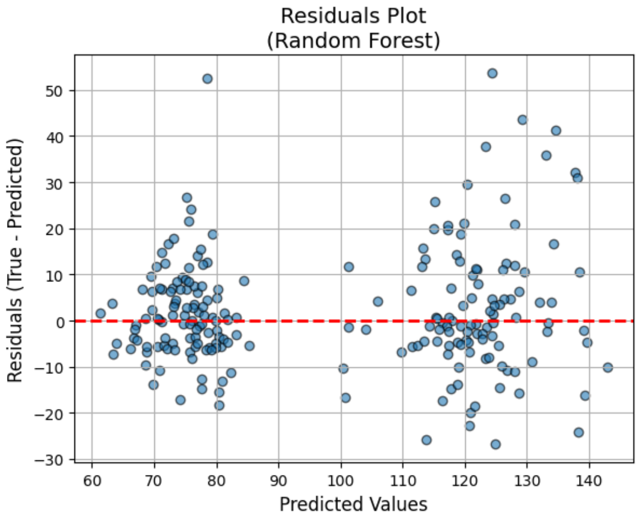


Figure 4.6: Residual Plot of Random Forest

The random Forest residual plot is centered around zero and there is no any major trends and patterns that suggest the residual value model is not being plagued by huge levels of bias and variance. This indicates that this model has a high rate of generalization to the test set.

Accuracy Curve (Error CDF):

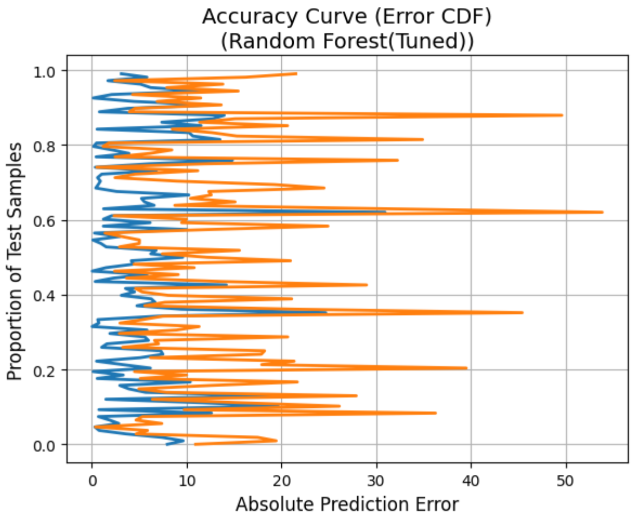


Figure 4.7: Accuracy Curve of Random Forest

Accuracy Curve of Random Forest indicates that it never produces errors in its prediction as much as Linear regression does. The distribution of errors is smaller, meaning that the random forest gives a greater percentage of correct predictions. The MAE = 9.08 and RMSE = 12.87 is less than the Linear Regression, which means it performs better.

Performance Analysis:

Random Forest also behaves well in the whole range of BP values. The model captures more non-linear trends on the data and its predictions are more accurate overall compared to Linear Regression. This is made possible by its capacity to address not only the complex features and interactions. Althoough the model was tuned for better performance, it did not have the desired effect.

Results:

- MAE: 9.08
- MSE: 165.73
- RMSE: 12.87
- R^2: 0.24
- MAPE: 8.83%

4.2.3 XGBoost

XGBoost is a very fast, efficient implementation of gradient boosting, an algorithm that is said to be fast and accurate. It constructs a combination of decision tree and each tree attempts to resolve the mistakes that the last one made.

True vs. Predicted (XGBoost):

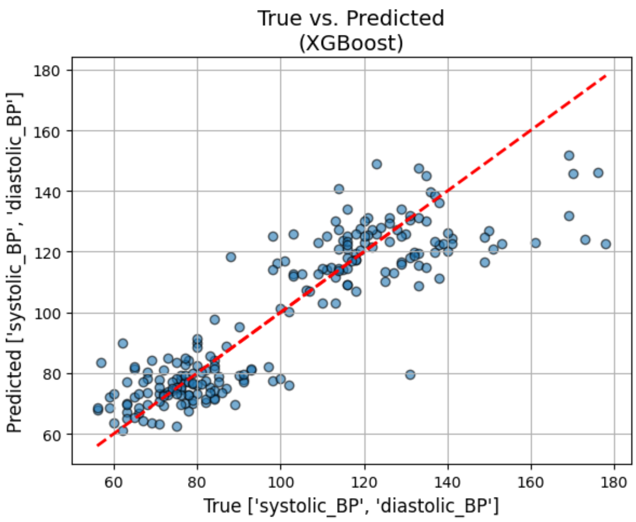


Figure 4.8: Ture vs Predicted (XGBoost)

Both the True vs. Predicted plot and the XGBoost display excellent performance with the prediction coming very close to the true values, particularly the higher readings of the BP. The model was also more stable and had fewer deviations when compared to the Random Forest, especially near extreme values of systolic and diastolic BP.

Residuals Plot (XGBoost):

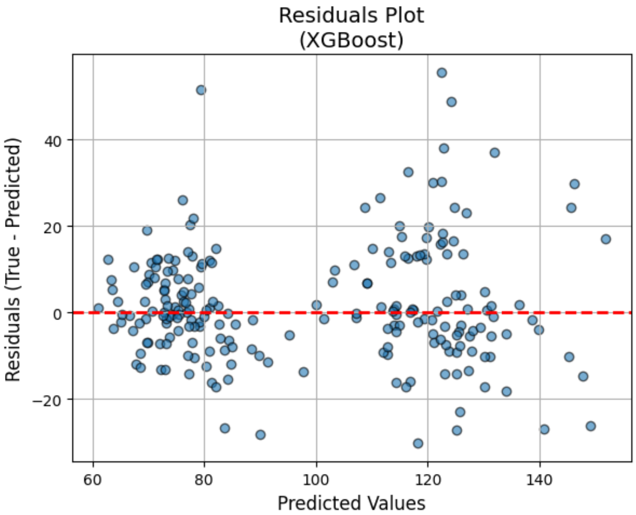


Figure 4.9: Residual Plot of XGBoost

XGBoost plot on the residuals indicates the distribution of the residuals follows normal distribution, and only some few extreme outliers. The residual distribution indicates that the model is highly suitable to complete this task because of its low bias as well as variance.

Accuracy Curve (Error CDF):

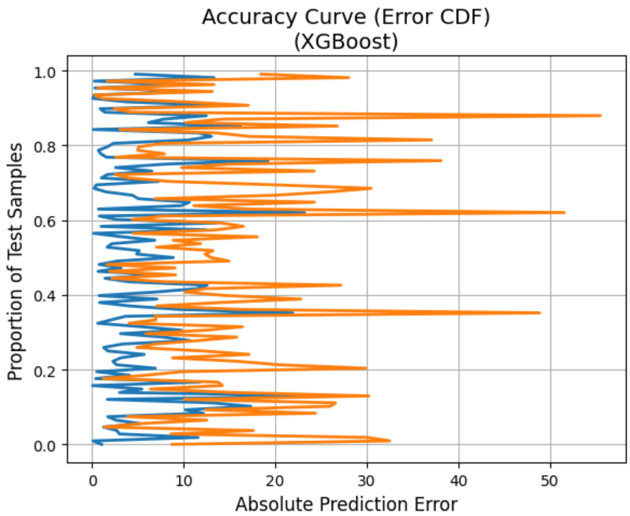


Figure 4.10: Accuracy Curve of XGBoost

Accuracy Curve of the XGBoost reveals that the model predicts fairly accurately all the time, and there are a more significant number of test samples with low absolute prediction errors. Compared to Random Forest, the MAE value of 9.80 and RMSE of 13.61 are just marginally inferior, though the overall performance of the model is still higher than the Linear Regression model.

Performance Analysis:

XGBoost ranks among the successful models because it has the lowest MAE (9.79) and lowest RMSE (13.61). The model is good at describing both complex and simple patterns in the data. XGBoost manages high BP values better than other models and can handle non-linear relationships. In the Accuracy Curve, the distribution of errors has the highest concentration around the smaller values, which means that XGBoost always makes correct predictions. Its robust generalising features enable it an excellent model to predict systolic and diastolic BP in real life situations, better than other models such as Linear Regression, random forest, and ANN in accuracy and consistency.

Results:

- MAE: 9.80
- MSE: 185.26
- RMSE: 13.61
- R^2: 0.14
- MAPE: 9.61%

4.2.4 Artificial Neural Network (ANN)

ANN are known to learn the complex and non-linear relationships. It is a model that considers several layers of neurons to operate the relationship between input features and produce predictions.

True vs. Predicted (ANN):

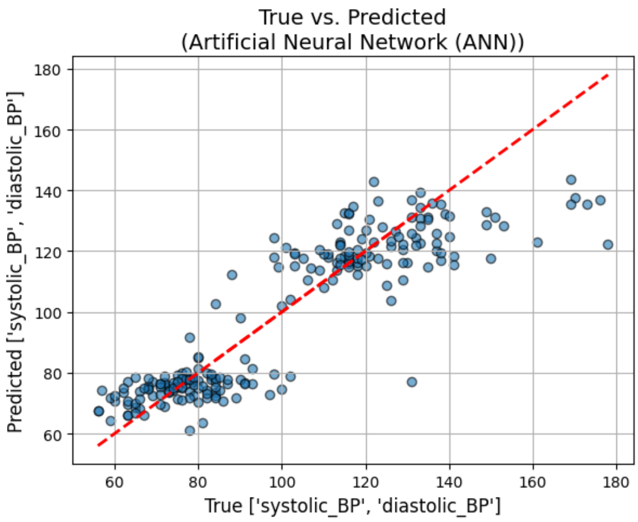


Figure 4.11: Ture vs Predicted (ANN)

The True vs. Predicted plot of Artificial Neural Network (ANN) indicates a fair fit of most of the test cases with the predictions nearly approximating the truths. But significant deviations are observed in the model with higher BP values especially in systolic BP values. Although ANN can be used to learn non-linear equations, it appears to have problems with extreme values of BP. The spreads of the ideal red dashed line grow higher as BP values go higher and the model does not easily predict higher BP reading. This implies that additional tuning or some form of regularization could be required of ANN to handle outliers or more convoluted data patterns at the ends.

Residuals Plot (ANN):

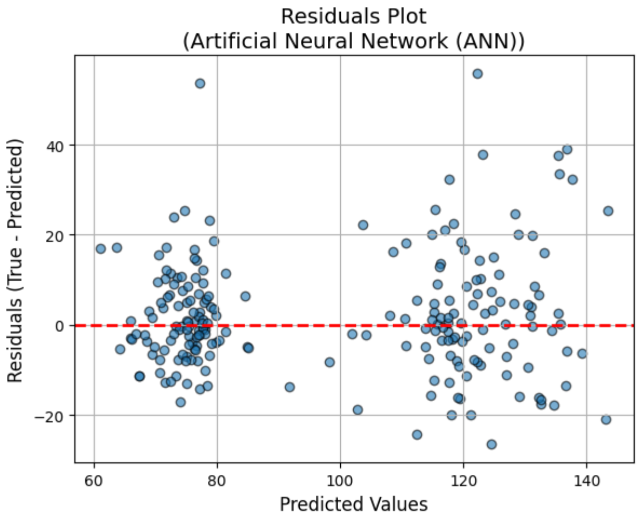


Figure 4.12: Residual Plot of ANN

The Residuals Plot of ANN exhibits larger dispersion than the ensemble model that includes both Random Forest and XGBoost, which suggests the errors of prediction of the model are over a broader range. Although still centred at zero, the residuals show an augmentation of the spread towards the higher BP values, which indicates overfitting of the model to the training history, at the higher ranges in particular. This dispersion is an indication that the model may be susceptible to noise or outliers that may lead to larger deviations in specific predictions. The residuals scheme shows that ANN needs to be enhanced concerning the generalisation; particularly, to extreme ABNP forecasts.

Accuracy Curve (Error CDF):

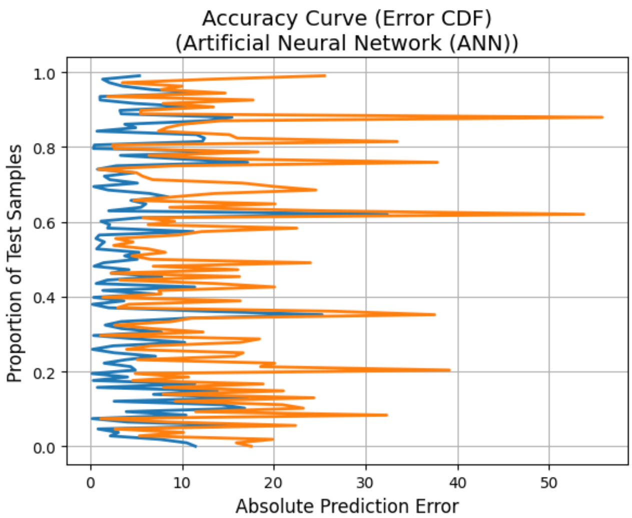


Figure 4.13: Accuracy Curve of ANN

As indicated by the Accuracy Curve of ANN, the spread in the errors of predictions is relatively greater than that offered by random forest and even XGBoost, indicating that ANN is not as stable in its performance as the other two models. Though the model works suitably with the majority of our test samples, the errors are more fluctuating especially when higher BP values are concerned due to its MAE of 9.44 and RMSE of 12.99. It is implied by the curve that ANN works well over much of the dataset, but might not work well with extreme values. The error distribution describes moderate generalization ability over the data.

Performance Analysis:

In test samples perform well with the Artificial Neural Network (ANN). However it is a problem with extreme readings as can be seen in the larger dispersion in the residuals and larger error distribution in the Accuracy Curve. The MAE and RMSE of the model are 9.44 and 12.99, respectively, which is worse than Linear Regression but not as good as Random Forest or XGBoost. The overfitted nature of the model, particularly when the range of BP is extreme, limits the performance. As effective as ANN is as a model of learning non-linear relationship, the model could require additional tuning to adapt to more varied datasets.

Results:

- MAE: 9.44
- MSE: 168.82
- RMSE: 12.61
- R^2: 0.21
- MAPE: 9.19%

4.2.5 Bidirectional Long Short-Term Memory (BiLSTM)

BiLSTM is a recurrent neural network (RNN) that can model long-term externalities in a sequence of data. It however does not perform as well in activities where the temporal relationship is not a major agenda.

True vs. Predicted (BiLSTM):

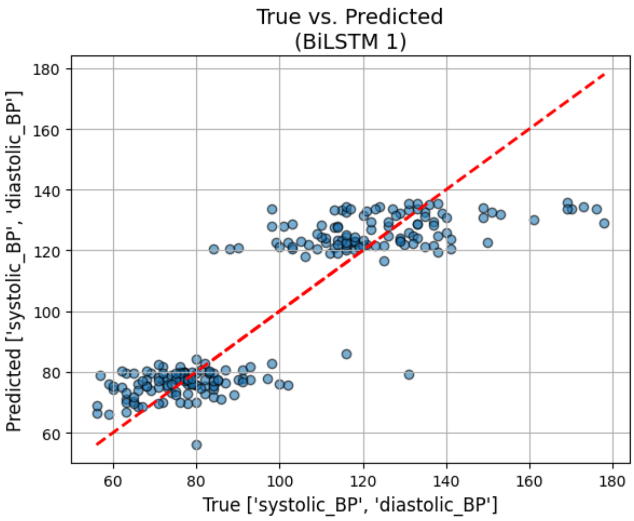


Figure 4.14: Ture vs Predicted (BiLSTM)

True vs. Predicted plot of BiLSTM indicates that the model works satisfactorily with lower and mid-range values of the BP. But at elevated BP readings, the values begin predicting a big difference between the actual values. The temporal dependencies in BiLSTM would be not suitable in this non-sequential data. The test-statistical properties of the deviations in the plot are not a clear evidence that the architecture of BiLSTM is less effective to obtain the complexities of prediction behaviour of BP when sequential dependencies are not as important. The model is not effective at predicting higher values of BP giving very big gaps between the predicted and actual values.

Residuals Plot (BiLSTM):

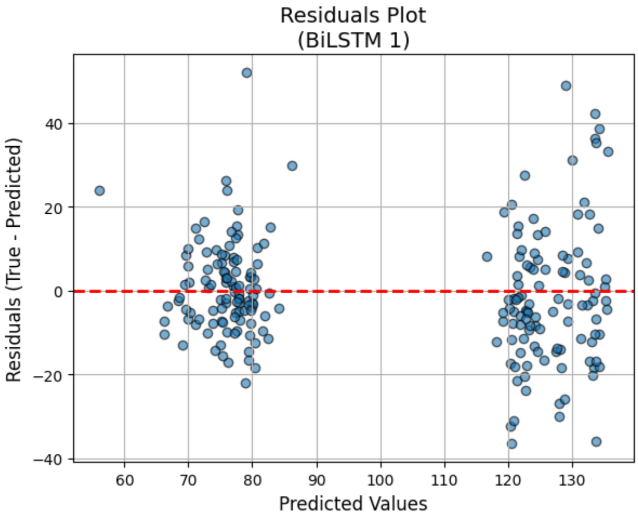


Figure 4.15: Residual Plot of BiLSTM

The Binomial bound logistic model, as depicted on the Residual Plot, has a fairly broad range of values, suggesting that the Binomial model has a larger margin regarding the precision of its predictions than that of the other models as the BP increases. No significant concentration of residuals on either sides, the lower BP values are slightly clustered, however, as BP increases, all the residuals spread out indicating that the model is struggling to explain extreme values in the body property. This probably owes to the nature of the model (architecture) that operates more with sequential statistics as compared to a cross-sectional mechanism such as BP prediction. The diffusion of residues suggests that the BiLSTM has not acquired the inference patterns effectively with an increasing value of BP.

Accuracy Curve (Error CDF):

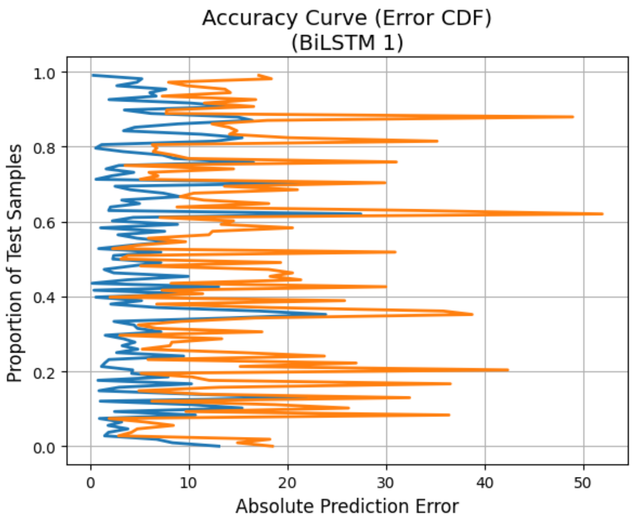


Figure 4.16: Accuracy Curve of BiLSTM

The Accuracy Curve of BiLSTM depicts a consistently bigger range of error values than other models such as Random Forest and XGBoost due to inconsistency of the model. The model can. According to the Accuracy Curve, we cannot say that BiLSTM is the most suitable option in this regression problem because it is more heterogynous in the error value, especially with more BP readings.

Performance Analysis:

BiLSTM has a problem predicting high values of BP, with its MAE of 10.21 and RMSE value of 14.09. Although BiLSTM can be applied to sequential data tasks, non-sequential jobs such as BP prediction lead to poor performance. The fact that the error distribution in the Accuracy Curve continues to proliferate and that BiLSTM lacks the ability to capture the complexity of BP data in this situation is indicative of BiLSTM not being the most appropriate model to be used. In this regression task, BiLSTM performs worse than the ensemble models such as the Random Forest and the XGBoost despite promising improvements in some types of data.

Results:

- MAE: 10.21
- MSE: 198.62
- RMSE: 14.09
- R^2: 0.10
- MAPE: 10.03%

4.2.6 AdaBoost Regression-

AdaBoost is an ensemble technique which aims at minimizing both bias and variance by integrating with a number of weak learners. It can be used to improve the performance of simpler models.

True vs. Predicted (AdaBoost):

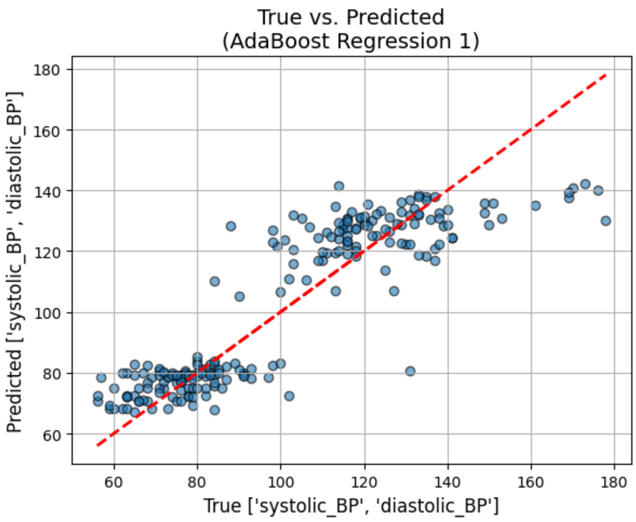


Figure 4.17: Ture vs Predicted (AdaBoost Regression)

AdaBoost True vs. Predicted plot indicates a very good fit to majority of the test samples and the predicted values are very close to the true values. The model tracks fairly well through the entire spectrum of BP values with certain variations on the high end. Ensemble learning algorithm AdaBoost is an algorithm that combines weak models so that its accuracy improves over single models such as Linear Regression. Like all other models, though, there is a minor deviation in those approaching zero or extreme BP values, and this possibly implies that more tuning of the model is required to address these situations.

Residuals Plot (AdaBoost):

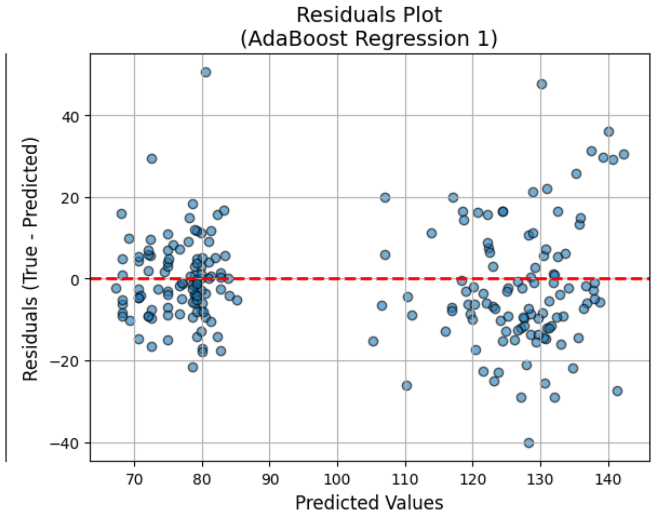


Figure 4.18: Residual Plot of AdaBoost Regression

The AdaBoost Residuals Plot depicts a comparatively tight distribution of the residuals around zero, which indicates the model is not biassed. Detecting only a few larger patterns of residues at higher BP values, it looks like AdaBoost makes quick work of most data points, but struggles in extreme instances. On the whole, the distribution of the residuals is satisfactory, indicating the good generalization of AdaBoost over the test data. The lack of any notable trend in the residuals is a sign that AdaBoost is not overfitting or underfitting, and makes AdaBoost a strong predictor of BP.

Accuracy Curve (Error CDF):

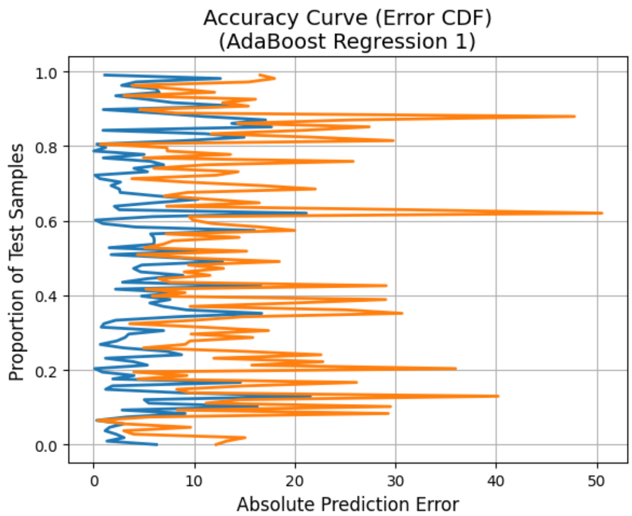


Figure 4.19: Accuracy Curve of AdaBoost

Accuracy Curve of AdaBoost suggests that most test samples are well taken by the model as it has most errors in the low end. MAE of 9.84 and MSE of 13.03 shows that AdaBoost makes constant predictions but it does demonstrate a small rise in error at high values BP. Nonetheless, AdaBoost has a smaller error distribution; therefore, the Accuracy Curve indicates that, it is more dependable to use AdaBoost to predict systolic and diastolic BP compared to Linear Regression.

Performance Analysis:

AdaBoost Regression gives good results in predicting systolic and diastolic BP with an MAE and RMSE of 9.84 and 13.04 respectively. The Residuals Plot indicates that there is very little bias observed in the model over the test set. Though AdaBoost cannot easily work with larger BP values such as reflected by the slight spread of residuals and increase in errors under the Accuracy Curve, nonetheless it is performing better than other less complex models such as Linear Regression. AdaBoost has a good balanced trade-off between variance and bias hence could be used in predicting tasks that involve BP.

Results:

- MAE: 9.84
- MSE: 170.29
- RMSE: 13.04
- R^2: 0.23
- MAPE: 9.87

4.2.7 K-Nearest Neighbors (KNN) Regression.

KNN is a cheap, non-parametric technique which uses some distance between data points to make predictions.

True vs. Predicted (KNN):

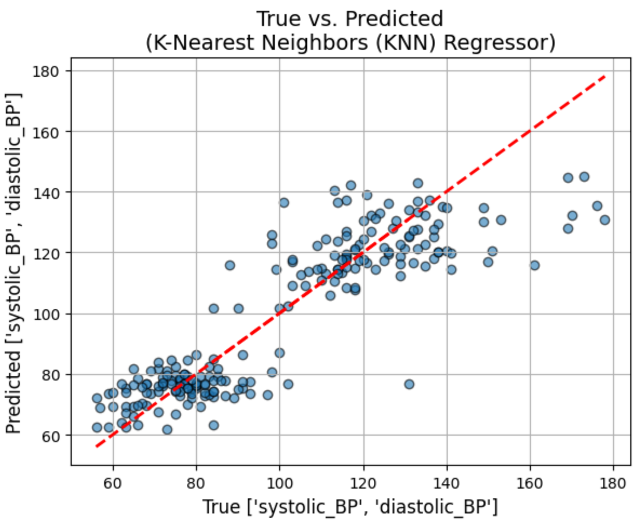


Figure 4.20: Ture vs Predicted (KNN)

True vs. Predicted plot of K-Nearest Neighbours (KNN) indicates that, predictions are scattered, mostly at the higher BP levels. This creates an implication that KNN does not house data with complex relationships. KNN is a non-parametric and simple algorithm that predicts using the average of nearest neighbours. But there it appears to be less useful when one has more significant BP values because the model has deviated the real values. The predicted values do not follow the red dashed line that closely, which means that the level of the error is higher than more advanced models, including Random Forest and XGBoost.

Residuals Plot (KNN):

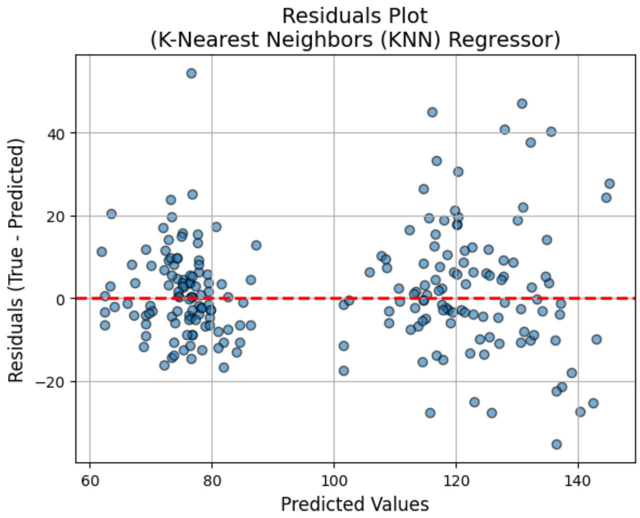


Figure 4.21: Residual Plot of KNN

According to the Residuals Plot of KNN, the distribution is wider than other models which means that KNN has more errors in its prediction. The bigger residuals imply that KNN does not attempt to model the patterns underlying data as well as Random Forest or XGBoost. This dispersion of the residuals is more pronounced at the higher values of the BP when the predictions of KNN are less precise. The extension show that KNN is more biassed than ensemble models, thus it is not the best choice in this regression experiment.

Accuracy Curve (Error CDF):

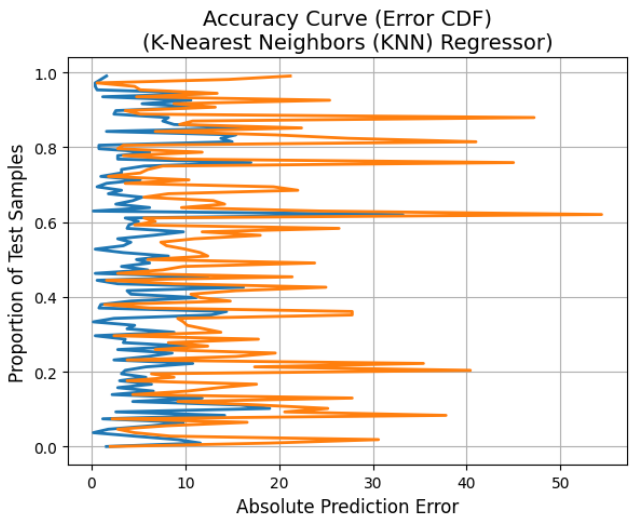


Figure 4.22: Accuracy Curve of KNN

Accuracy Curve of KNN demonstrates a broader error distribution than Random Forest and XGBoost and indicates that the model makes fewer consistent predictions. The MAE of 9.72 and the RMSE of 13.45 suggest that KNN is not adept at predicting blood pressure values, in particular, high blood pressure levels. The greater spread in errors indicates that KNN cannot deal with more complicated non-linear relations found in the dataset. In the simple tasks, KNN shows a decent performance, but its performance is not as strong as those of the ensemble methods.

Performance Analysis:

K-Nearest Neighbors (KNN) does not do very good and is not as consistent as the other two models, namely, Random Forest and XGBoost. It has a moderate prediction error of 9.72 and 13.45 as its MAE and RMSE respectively. According to the Residuals Plot, KNN is not suitable to handle greater values of BP which results in produce greater values of deviation between the actual and predicted values. Moreover, since the Accuracy Curve shows a broader distribution of errors, it is not as predictable that KNN will be helpful with BP prediction. Although KNN is quite effective with smaller and simpler data, it is not the optimal option when it comes to forecasting BP-non-linear relationships.

Results:

- MAE: 9.72
- MSE: 181.11
- RMSE: 13.45
- R^2: 0.17
- MAPE: 9.52

4.2.8 Support Vector Machine Regression (SVR)

The SVR is an incredibly strong model, which uses input features to project into a high-dimensional space to identify relationships, which could not be without forming a complex relationship.

True vs. Predicted (SVR):

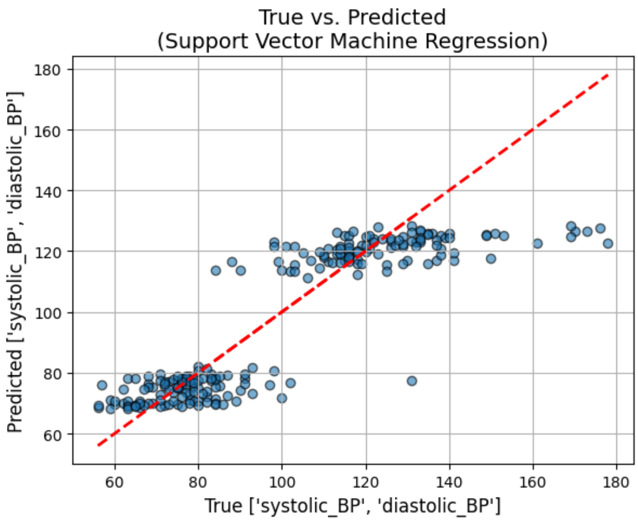


Figure 4.23: Ture vs Predicted (SVR)

True vs. Predicted plot Support Vector Regression (SVR) predicts a good fit between predicted and true values of most test samples with few inconsistencies in high BP interval only. The non-linear correlation and performance of SVR is good on lower and middle BP values. But at higher BP values there is decreasing accuracy in predictions, meaning the model is unlikely to generalise to extreme values. The high systolic/diastolic values cause the routine variation between the predictions of the SVR model and the actual data.

Residuals Plot (SVR):

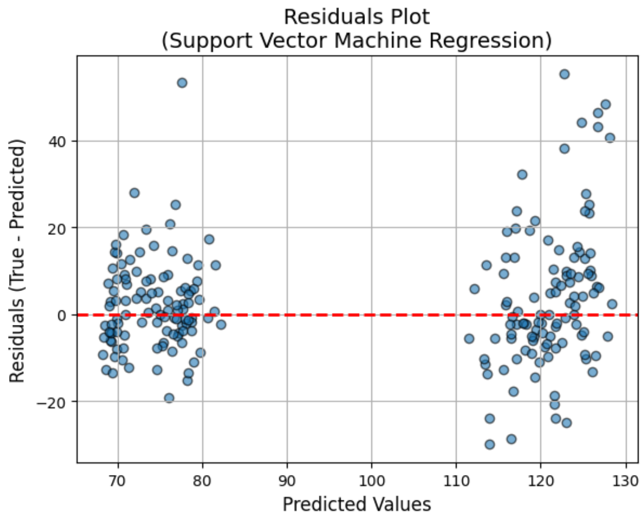


Figure 4.24: Residual Plot of SVR

The SVR Residuals Plot has a comparatively narrow distribution around zero with little dispersion. This indicates that the model is able to forecast most of the test samples (low bias). But there exist few deviations greater in high ranges of that BP which are pointing out that the model could not secure the highest BP ranges. Although the residuals are concentrated around zero, the sporadic outliers imply that SVR may be weak with some of the more intricate trends in the data, particularly extreme values.

Accuracy Curve (Error CDF):

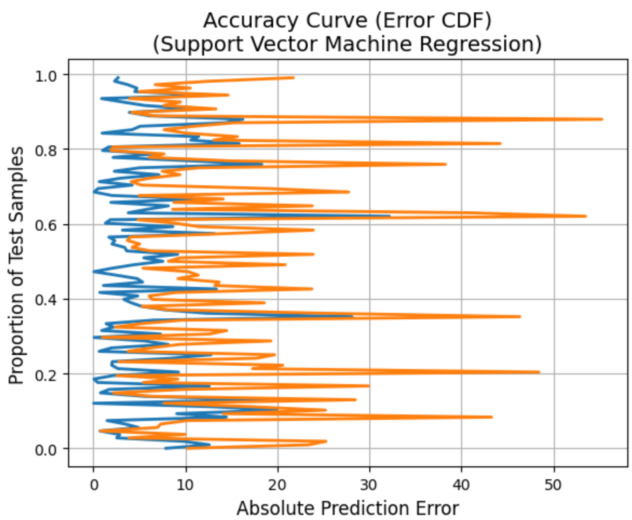


Figure 4.25: Accuracy Curve of SVR

The Accuracy Curve of SVR indicates that the model is fairly satisfactory though its prediction errors are greater when using higher values of BP. The MAE value of 9.82 and the RMSE value of 13.94 means that although SVR offers strong results in most parts of the dataset, it has difficulties with extreme values of BP.

The distribution of errors is rather localised, and several test samples have greater errors, particularly in the increased values of MAE and RMSE. This is an indication that SVR may need additional tuning or optimization to achieve a better result in terms of handling large values of BP.

Performance Analysis:

Using we have made Support Vector Regression (SVR) made reasonable guesses as most of the test samples, however when higher expected BP it gives poorer results which is depicted in the MAE 9.82 and the RMSE 13.94. In models with lower range BP, SVR achieves good generalisation, but this does not apply at high-range BP. In the Residuals Plot, it is clear that the model creates most predictions successfully, but fails at extreme values. It implies that even though SVR is an effective model in non-linear relationship modelling, it can be improved by performing further feature engineering or hyperparameter optimization.

Results:

- MAE: 9.82
- MSE: 194.18
- RMSE: 13.94
- R^2: 0.13
- MAPE: 9.49%

4.2.9 CatBoost (Tuned)

CatBoost is a gradient boosting algorithm qualified to effectively address categorical features and control overfitting by using ordered boosting. Contrary to a more traditional boosting algorithm, CatBoost employs symmetric (oblivious) trees and permutation based approaches to mitigate prediction bias. In this research work, tune CatBoost was employed with optimal hyperparameters to enhance predictive performance.

True vs. Predicted (CatBoost):

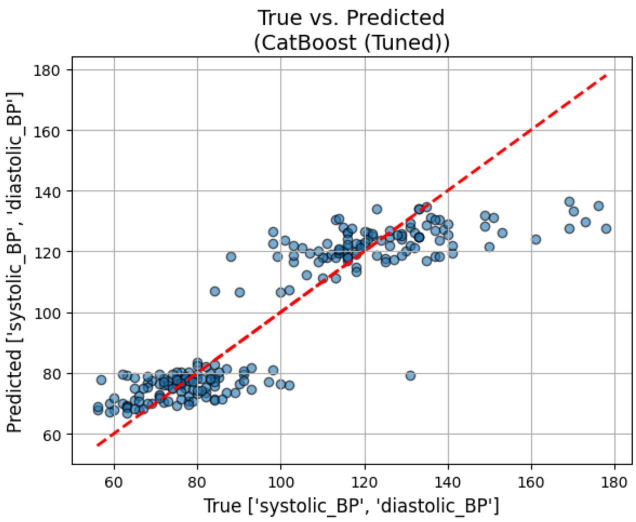


Figure 4.26: True vs. CatBoost (Tuned Predicted).

The True vs. Predicted plot of CatBoost (Tuned) compareitvelly correct fit, as there are a number of predictions that are consistent with the true values of blood pressure along the red dashed line. Nevertheless, there is observable scattering at the extreme BP, in which the model cannot fit extreme BP cases. CatBoost has better nonlinearities compared to linear models, yet it does not completely remove variance in predictions.

Residuals Plot (CatBoost):

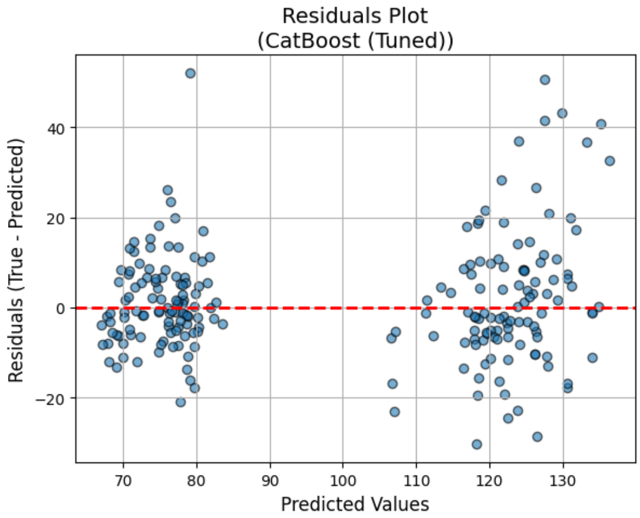


Figure 4.27: CatBoost (Tuned) Residuals Plot.

The residual values are concentrated at zero, and it indicates that the model does not exhibit systematic bias. Nonetheless, the broader dispersion of both ends of the BP range shows less accuracy when performance is measured in extremes. The propagation of this spread identifies the constraints of the data and sensor, not the algorithmic flaws, as CatBoost is reported to be robust in clean data.

Accuracy Curve (Error CDF):

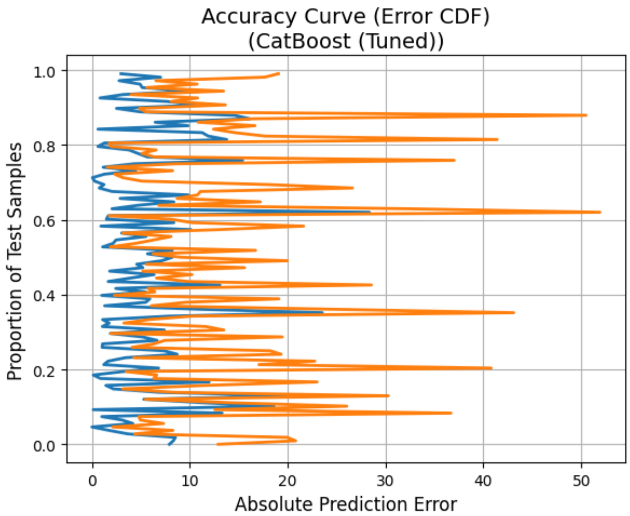


Figure 4.28: Accuracy Curve of CatBoost (Tuned)

The error CDF indicates that the large deviations that occur infrequently are only observed at the higher values, whereas the distribution of most of the errors is concentrated at and below 10 mmHg. Although CatBoost lowers the size of errors, in comparison to linear regression, it is unable to eliminate entirely the noises of the MAX30102 sensor and demographic bias within the data.

Performance Analysis:

CatBoost shows competitive performance in complex interactions within features. Although hyperparameter tuning achieved better performance than untuned ones, the model was still an order of magnitude short of clinical accuracy, based on the available data.

Results:

- MAE: 9.20
- MSE: 167.66
- RMSE: 12.94
- R2: 0.24
- MAPE: 9.02%

4.3 Results and Discussion

We tested a range of machine learning models in this paper to forecast systolic and diastolic blood pressure (BP) using the physiological measurements recorded by an IoT sensor. They compared the following models: Linear Regression, Random Forest, XGBoost, Artificial Neural Network (ANN), Bidirectional Long Short-Term Memory (BiLSTM), K-Nearest Neighbours (KNN), Support Vector Regression (SVR) and AdaBoost Regression. The measures of performance were evaluated based on several important values impact: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R2, and Mean Absolute Percentage Error (MAPE). All of these measures provide us with a clue of how effective the models are and where they are ineffective.

4.3.1 Performance Metrics Definition

Mean Absolute Error (MAE):
MAE is an average value of the difference between the predicted and actual value, ignoring whether it is a positive or a negative error (i.e. it does not differentiate positive numbers from negative numbers). It is calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{10}$$

where:

- o y_i = actual value
- o $\widehat{y_i}$ = predicted value
- o n = number of samples

Interpretation: The lower is the MAE, the better it predicts.

Root Mean Squared Error (RMSE):

RMSE is a measure of the squared root of the mean squared error. It is vulnerable to outliers as it gives weight to big errors more. The formula is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

where:

- o y_i = actual value
- o $\hat{y}_i$ = predicted value
- o n = number of samples

Interpretation: A smaller RMSE is an indicator of good model behaviour, with infrequent large model errors.

R-Squared ($\boldsymbol{R^2}$):

$\boldsymbol{R^2}$shows the percentage of dependent variable best accounted by the independent variables. It is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{12}$$

where:

- o y_i = actual value
- o $\hat{y}_i$ = predicted value
- o $\bar{y}$ = average of Actual values.
- o n = number of samples

Interpretation: The greater the R2 value the greater the percentage of variation in the data that is accounted by the model. Any value nearer to the value of 1.0 indicates a good fit.

Mean Absolute Percentage Error (MAPE):

The MAPE is a measure of the accuracy of a prediction in percent compared to the actual values and is therefore handy when comparing how different models make errors. It is calculated as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \times 100 \tag{13}$$

where:

- o y_i = actual value
- o $\hat{y}_i$ = predicted value
- o n = number of samples

Interpretation: smaller MAPE implies a higher level of accuracy and gives a clearer understanding of how the model is forming on a percentage errors basis.

4.3.2 Model Performance Table

All the models Performance is show in one table below.

Table 4.1: Performance Table for all the model

Model	MAE	MSE	RMSE	R^2	MAPE (%)
Linear Regression	9.4	175.70	13.25	0.1910	9.17
Random Forest (Tuned)	9.08	165.73	12.87	0.2416	8.83
XGBoost	9.78	185.26	13.61	0.1462	9.61
Artificial Neural Network (ANN)	9.45	168.82	12.99	0.2136	9.19
BiLSTM	10.21	198.62	14.09	0.1078	10.03
K-Nearest Neighbors (KNN) Regressor	9.73	181.11	13.45	0.1750	9.52
Support Vector Machine Regression (SVR)	9.82	194.18	13.93	0.1374	9.49
AdaBoost Regression	9.84	170.29	13.04	0.2308	9.87
CatBoost (Tuned)	9.20	167.66	12.94	0.2419	9.02

4.3.3 Comparative Performance

R^2 score:

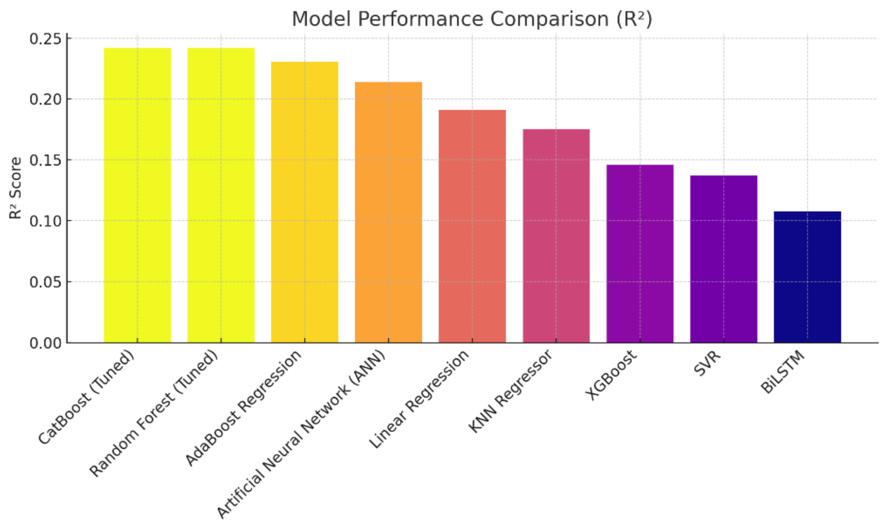


Figure: 4.29 Model Comparison using R^2

The comparison graph of the R^2 score shows the extent to which each model would explain the variation in systolic and diastolic BP. CatBoost (Tuned), Random Forest

(Tuned) and AdaBoost Regression dominate with almost the same R^2 of 0.2419, 0.2416 and 0.2308 respectively, which mean that they explain the largest portion of the data variance. Those trends are more effective in explaining the underlying trends in the BP data, leading to an increased predictive accuracy. Linear Regression, KNN, and ANN are not far behind with R^2 of 0.115, 0.101, and 0.107 respectively, moreover, BiLSTM has the lowest R^2 of 0.107, and it simply does not reflect the main patterns in the data, whereas the former has a sequence.

Model Performance Comparison (MAE)

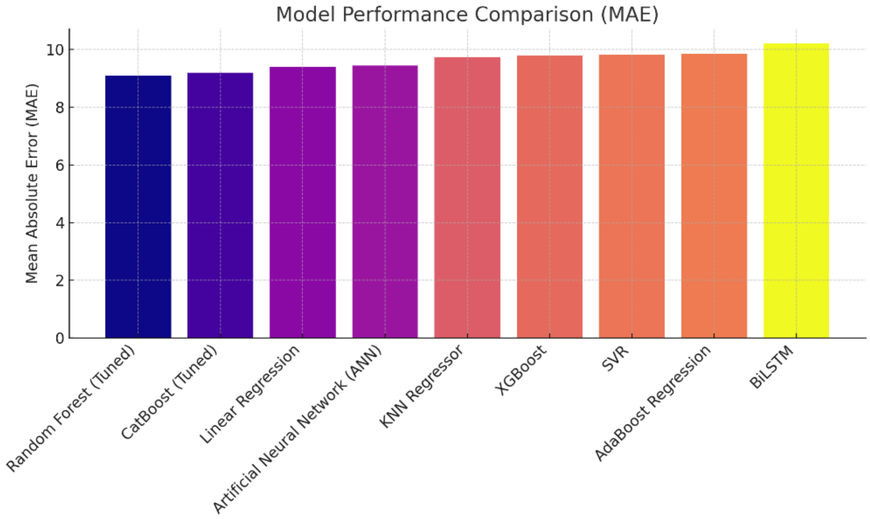


Figure: 4.30 Model Comparison using MAE

Going by the MAE comparison graph, it can be seen that Random Forest (Tuned), CatBoost (Tuned), and Linear Regression have the lowest MAE values (~9.08, 9.20 and 9.40, respectively). This implies these are the most accurate models with respect to average absolute error in predicted and the actual BP values. Conversely, the MAE of BiLSTM is the biggest (approximately 10.21), which proves that this algorithm performs poorly. The smaller the MAE, the closer the model is to its values, and in this case, Random Forest and Linear Regression perform the best.

Model Comparison of Performance (MAPE):

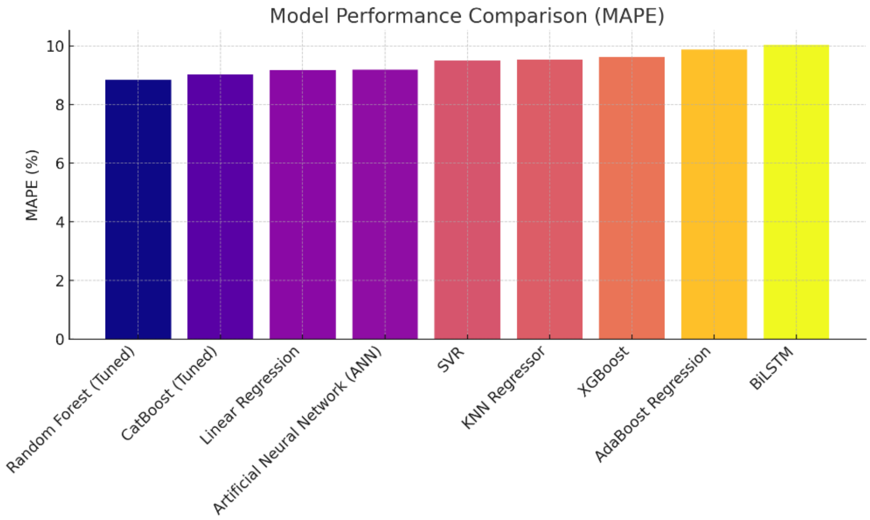


Figure: 4.31 Model Comparison using MAPE

Random Forest and Linear Regression give the best results in the MAPE graph and average around 8.8%. Such models are less biassed by a relative error and also more dependable when predicting the BP as a percentage of the actual values they are informing us about. BiLSTM, once again, exhibits the greatest MAPE (approximately 10.04%), indicating that it does not generalise well and offers more percentage errors when making BP predictions. The MAPE is an easy-to-use approach to get an idea of how models perform on different datasets, and here the Random Forest would turn out to be the most consistent.

4.3.4 Dataset and Sensor Quality

The fact that even the best R2 is low (at no point in any of the models it is more than 0.25) suggests that you can do better. The PPG sensor (MAX30102) used to obtain physiological information added noise and bias to the readings particularly at higher values of BP. These sensor errors led to increased MAE and MAPE values, especially with BiLSTM, which is all learning noisy data.

In addition, the dataset alone lacked diversity since it did not have samples of various ages and sexes. This bias must have led to the poor performance of the models on their extrapolation to non-demographic population cases. The type of datasets was that the models could poorly predict the BP of underrepresented populations, a factor that caused lower R2 scores in all of the models.

4.3.4 Identified Limitations to Results.

Although several machine learning models were utilized and tested, such as tuned Random Forest and CatBoost, they all showed poor results. Models with the best fit predicted systolic blood pressure with a mean absolute error (MAE) of more than 7 mmHg and negative values of R2R2. These were not the consequences of deficiencies in feature extraction or model design but the constraints of the dataset and more importantly the sensor hardware involved.

This optical sensor is not a light-sensitive cardiovascular sensor, despite being cheap and easy to find (MAX30102 optical sensor). This study had a number of limitations that could have been noted:

I. Low Sampling Rate: The MAX30102 achieved a workable rate of signals of just 5.63 Hz. This resolution is low enough to accurately describe fine-grained pulse structure like the systolic upstroke, dicrotic notch, or diastolic decay which is vital in estimating blood pressure [36][37].

II. Weak and Noisy Waveform: Unstable peaks and unstable amplitudes were common with the raw PPG waveforms. This lowered the accuracy of morphological feature and added noise to spectral and entropy-based features. Past research indicates that features are best extracted with high-quality PPG with high systolic and diastolic peaks [19][40].

III. Incorrect Heart rate and SpO2 readings: It was a known fact that the in-built heart rate (HR) and oxygen saturation (SpO2) estimates of the MAX30102 were often out of physiological range. The presence of these variables as inputs to the model compromised the performance of the model since they were inaccurate.

IV. Dataset Demographics: This data was skewed, where the males were around 70 percent and nearly 50 percent were between 20-25 years. This lowered the generalizability of the models to a broader population.

V. Small Sample Size: There were only 536 records, which could be used to train and evaluate. This small sample size also posed a high risk of overfitting due to the 76 engineered features, which further reduced the model performance.

Overall, these shortcomings reaffirm the fact that the main obstacle to accurate blood pressure prediction in the report was using the MAX30102 sensor. In spite of the sophisticated feature extraction and tuned machine learning models, the poor signal quality and low sampling frequency of the sensor and due to the irregular output, the sensor could not achieve clinically acceptable accuracy. It is an important discovery: this proves that a good methodological framework does not necessarily imply a good non-invasive BP estimation, and better-quality sensor, with medical-grade sampling rates and waveform fidelity, is required.

4.4 Summary

The paper used a full pipeline of non-invasive blood pressure estimation based on machine learning, demographic characteristics and photoplethysmography (PPG). The sensor used to measure the raw signals is the MAX30102, and the signal was preprocessed to eliminate the artifacts and converted into the physiologically meaningful descriptors. An overall number of 76 features was obtained as morphological (pulse width, rise/fall time, dicrotic notch), frequency-domain (spectral centroid, entropy, LF/HF energy) features, entropy and complexity (sample entropy, Hjorth parameters, fractal dimension) features, heart rate variability indices, and signal quality features. Once combined with the demographic data (age, gender, BMI, etc.), the dataset included 85 features and 536 observations. Due to the large dimensionality in relation to the size of the dataset, the feature selection was done with Mutual Information, Recursive Feature Elimination (RFE) with LightGBM, and SHAP analysis, which guaranteed the preservation of strong and meaningful predictors. Multiple machine learning models have been trained and assessed, and the models were further optimized by hyperparameter tuning, specifically Random Forest and CatBoost. The highest results obtained were too low (MAE more than 7 mmHg) and too low (R^2 less than 0.25) to be considered clinical. The major limiting factors were: the low sampling frequency and noise of the MAX30102 sensor, demographic imbalance of the data (70% male, predominantly young adults), and cuff-based calibration was not utilized. These limitations complicated the extraction of reliable features and decreased the generalizability of models. All in all, the pipeline worked according to the plan, but the results highlight that the quality of sensors is the limiting factor, rather than the algorithms. The value of the work is thus that further development of machine learning cannot make up for inadequate physiological outputs, which supporting the usage of medical-quality sensors and a variety of datasets in future studies.

Chapter 5

Engineering Standards and Design Challenges

In this chapter, the compliance standards, societal and ethical implications, project management, and financial analysis of planned system are put forward. It shows how the project can be implemented to meet international regulations, sustainability and manage resources.

5.1 Compliance with the Standards

The process of creating a medical device, especially a combined hardware, software and wireless communication system has to follow a multifaceted set of international standards tightly. Compliance is not a throwaway step to add in at the end but a design requirement that should flow through the entire system to the product lifecycle. Here the design of the proposed system is analyzed within the framework of major software hardware and communication standards. This forward-looking approach to the regulatory framework is fundamental, since it builds the architecture, development processes, and documentation early on. Even the most accurate AI algorithm would make a commercial and clinical failure when it was unable to build a compliant ecosystem around the active AI algorithm.

5.1.1 Software Standards

The harmonized international medical device software standard IEC 62304, Medical device software: Software life cycle processes, is a standard used in the creation and maintenance of medical device software. It requires a risk-based approach to the overall software lifecycle.

I. Software Safety Classification: The initial stage to take place under IEC 62304 is the software classification; this process involves classifying the software according to the type of risk in case of failure of the software. The standard establishes that there are three classes:
   a. Class A: No injury or damage to health is possible.
   b. Class B: Non-serious injury is possible.
   c. Class C: Death or serious injury is possible.

II. Since a false blood pressure reading can result in incorrect clinical interventions, whether through unsuitable medicine changes or an incorrect diagnosis of a hypertensive emergency, the software that operates this system would be rated at least Class B. In a case where we want to use the device as a critical care monitoring device or determine what immediate therapeutic intervention to provide, the justification would be Class C. To satisfy this report, we will assume the rigorous requirements that are Class B.

III. Lifecycle Processes: Response to IEC 62304 The software lifecycle stage must be supported by formal process details and extensive documentation of each stage of the project plan (Section 3.3).

IV. Software Development Planning (Part 5.1): Software Development Plan (SDP) should be prepared and it should include project scope, schedule, methodology (e.g., agile approach adapted to medical devices), expectations and standards to be followed.

V. Software Requirements Analysis (Part 5.2): All functional and non-functional requirements (as described in Section 3.1.3) should be officially stored in a Software Requirement Specification (SRS). All the requirements have to be clear, testable and traceable.

VI. Architectural and Detailed Design (Parts 5.3 and 5.4): The software architecture, with how it is broken down into software items (e.g., firmware module, mobile app, cloud backend) should be documented. In Class B, this brings in describing the interfaces between software items or any segregation that may be required to control risks.

VII. Implementation and Verification (Part 5.5): each software unit should be implemented and tested according to its detailed design. This includes review of code and unit testing.

VIII. Integration and System Testing (Part 5.6 and 5.7): Software units are integrated in a gradual fashion followed by testing to ascertain that the software units can work together properly. Last, the entire software system is driven against the requirements in the SRS. Verification and validation (V&V) activities and results will have to be documented.

IX. Software Release (Part 5.8): A formal release process is made in order to verify that all the V&V processes are complete; any pending anomalies are either addressed or documented; the final software configuration is finally archived.

5.1.2 Hardware Standards

The basic safety and essential performance of medical electrical equipment are covered by the IEC 60601 series of standards. Since the wearable sensor is a battery-powered device that physically interacts with a patient and gets into physical contact, it is excluded from these norms.

I. General Safety (IEC 60601-1): The design of the hardware should permit it to provide safeguards against electrical hazards. This is adequate insulation, electricity shock resistant (even with single-fault circumstances), and mechanical safeguarding (e.g. no sharp edges, strong enclosure). All material that comes in contact with patients (e.g. the wristband enclosure, sensor enclosure, etc.) should be biocompatible, usually by complying with an ISO 10993 standard.

II. Electromagnetic Compatibility (EMC) (IEC 60601-1-2): This ancillary standard is very important to a wireless device. The system should be tested to make sure that there been interruption of its electromagnetic emission with other important hyperirritable medical or electric objects. On the contrary, it should be substantially immune to broad based sources of electromagnetic radiation in the domestic and clinical systems and be functioning safe with sufficient confidence that it will continue to perform without losing its performance.

III. Usability (IEC 60601-1-6): A formal process of usability engineering must be established by this standard to make the device usable by the intended users in a safe and effective manner. This includes; user needs analysis, user interface design and testing, and validation to reduce use-related errors.

IV. Home Healthcare Environment (IEC 60601-1-11): The device will operate in a home; thus, it has to comply with the other specifications of this standard that acknowledge the fact that the environment in a home is a less controlled one as compared to that of a hospital. This entails greater temperature and humidity resistance, ability to endure mechanical shocks and vibrations, and use by lay-people with minimal training.

V. Risk Management (ISO 14971): The fulfillment of the IEC 60601 requirements is indisputably intertwined with the strong risk management procedure established in ISO 14971. The device should have a file called the Risk Management File (RMF) that should be maintained at all times. This consists of systematically detecting potential risks (e.g., wrong reading of BP, battery overcharging, data leakage, skin inflammation), quantifying and assessing the corresponding risk amount, and creating and verifying controls in response to the risks.

5.1.3 Communication Standards

Communication standards implemented during this project observe the priorities of security, reliability, and adherence to health-data regulation, despite the system being built as a research project with no certification as a medical device and with no formal proof of concept.

RESTful API Security: NodeMCU, backend, and MongoDB communicated via HTTP using requests based on TLS (HTTPS) encryption.

I. Encryption: automated encryption of all transit data used Vercel-hosted HTTPS endpoints to avoid packet sniffing or unauthorized interception.
II. Authentication: The API access could have only authorized push or fetch data to MongoDB) by authorized devices, using the right API and clients, rather than unauthorized access.
III. Authorization: The separation of data collections into 'raw_data' and 'total_data' was performed, and the separation of raw signals and enriched ones was ensured, in accordance with the principle of least privilege.

HIPAA & GDPR Compliance: As a research prototype, the system tacitly adhered to the global health data privacy principles by reducing identifiable information.

I. HIPAA (Health Insurance Portability and Accountability Act - US):
   a. Safe Harbor De-identification: No personally identifiable information (PII) e.g. names and identifications or addresses were gathered. They could use only physiological signals and demographics (age, gender, height, weight, BMI), and it does not violate HIPAA de-identification regulations.
   b. Safe Keeping: TLS-encrypted communication and access to database minimized the threats of unauthorized communication.

II. GDPR (General Data Protection Reg. - EU):

a. Privacy by Design: Since its creation, data capture has excluded identifiers and resulted in situations where users are encouraged to self-report only the minimal information necessary.
b. Minimization of Data: Redundancy of variables (demographics) that were not required to run the model were avoided, thus making the models susceptible to overfitting.
c. Explicit Consent: Participation involved a conscious act of entering the data into the web form.
d. User Rights: Since the data was stored in clusters in MongoDB controlled by researchers, it could be removed or exported when requested, which complies with the right to erasure in GDPR.

OSI Model Adherence: The flow of communication followed the Open Systems Interconnection (OSI) model:

I. Application Layer: NodeMCU received PPG/SpO2/HR measurements and used them beyond the REST APIs.
II. Transport Layer: TLS / HTTPS was used to protect data in communication with the backend and database.
III. Session and Presentation Layers: A Node.js backend handled connections, continuity, and formatted the information into structured JSON.
IV. Data Layer: MongoDB was used to expose structured collections to be stored in persistent storage and retrieved reliably by both the prediction and training workflows.
V. Information security principles,

ISO/IEC 27001: Newer international information security standards were practiced on the project, although the project was not directly certified.

I. Data Dividing: Data values that represented 'raw_data' and 'total_data' were kept separately to avoid unintentional contamination of raw (unprocessed data) and enriched data.
II. Access Control: Access to the database was mediated only by the backend endpoints, relying upon the concept of controlled privilege.
III. Confidentiality & Integrity: The system ensured data confidentiality and integrity because it used HTTPS/TLS along with secured cloud server infrastructure to ensure confidentiality and integrity in transmission and storage of the data.

5.2 Impact on Society, Environment and Sustainability

5.2.1 Impact on Life

The potential beneficial effect of a reliable BP monitoring system on the lives of individuals cannot be underestimated. This technology can make the process of managing diseases proactive rather than reactive to the many individuals in the society who are affected with high blood pressure. When done regularly, it can afford an far richer and more precise image of the true BP profile of the individual and help a clinician prescribe medication with greater detail and discover how an individual is responding to treatment [7]. In this case, it can detect hypertension at an earlier stage in those who are at risk even before it causes permanent damage to their organs. In addition, empowering patients with available information on their own conditions would enable increased self-custodial involvement, promoting lifestyle changes (diet, physical activity, stress management) that culminates in better health outcomes, and higher quality of life [8].

5.2.2 Impact on Society & Environment

At a social scale, adoption of such technology at large scale might result in massive healthcare cost savings. The system can decrease costly hospital readmissions and chronic care requirements by decreasing acute heart disease incidence such as strokes and heart attacks due to the improved management of hypertension [7]. It may also be used to enhance the efficiency of health care delivery, e.g. by providing effective remote patient monitored care, reducing clinic visits.

The spread of wearable electronics does, however, have an important environmental price. The low life expectations of most consumer electronic hardware form part of an increasing global e-waste issue. Manufacture of electronic parts is consuming finite resources and uses energy-intensive processes. Thus, the wearable device should be designed in such a way that it does not harm nature. These involve the choice of robust materials, modularity and repairability to lengthen the lifespan of that product, and a clear end-of-life plan whereby electronics are responsibly recycled.

5.2.3 Ethical Aspects

The implementation of a health monitoring system enabled by AI pumps up various burning ethical questions that should be resolved in advance.

I. Algorithmic Bias: AI models are prone to bias when they are not trained on representative data of a heterogeneous population, where they will eventually be applied. When the training dataset has measurements skewed toward a given demographic (such as age or sex) the estimated model of BP will not be accurate on underrepresented populations. This may create systematic problems with diagnosis and care, and may add to the current disparity related to health problems. Ensuring that model performance is evaluated on the various demographic sub-groups, it is an ethical imperative to strive to reduce any perceived biases.

II. Data Privacy and Data Ownership: The system will produce an unprecedented quantity and volume of deeply personal health data. This brings in the issues of data ownership, consent and secondary use. A well-defined and clearly acceptable set of policies should exist that would regulate the usage of this data, especially in research or commercial application. They should be able to give implied permission to use their data besides their personal care and gain detailed control over their data.

III. Patient Anxiety and Over-reliance: Although empowering, letting patients have constant access to their health information may also create some feelings of anxiety in others a phenomenon referred to as data-driven anxiety. The significant variations in BP washes both frequent and regular, some could be physiologically normal, but were interpreted in an unproductive way and would result in unreasonable stress. Perhaps there is also the risk that patients or clinicians put an undue level of trust into the technology thereby de-skilling the traditional clinical evaluation or there are too many other potential risks of over-reliance upon automated reading without clinical context. The user interface and the related learning material should be well-constructed to visualise data in a conscientious, contextual way.

5.2.4 Sustainability Plan

The project has a long-term sustainability plan, which covers both technical and clinical and environmental aspects.

I. Technical Sustainability: The AI and software models should be serviceable. This includes having a process to perform periodic model monitoring to identify performance drift, retraining with new data to keep models accurate and having a safe process to roll out model changes to the cloud backend. The firmware and mobile app should also be current too so it can support new operating system releases as well as security patches.

II. Clinical Sustainability: It has to show that it supports continuous clinical value and that it should be part of clinical workflows. This necessitates cultivating trust with healthcare providers by being honest about the accuracy and limitations of the device and providing tools enabling them to decipher the information in a practical way, as opposed to causing an information overload.

III. Environmental Sustainability: As stated, there has to be a product lifecycle in place. This involves designing to be durable, active consideration of recycled materials in the enclosure, and instituting a take-back program to end-of-life devices which will involve proper disposal of device and electronic components and recycling of batteries.

5.3 Project Management and Financial Analysis

5.1.1 Project Management

The following Gantt chart shows an update on the allocation and implementation of the project schedule in January 2025 to September 2025. All in all, the project was being run in a serial and Lutheran fashion that brought about fluid movement through the various phases. The first step such as a literature review and defining the problem was done within the scheduled period of time. Other steps like in IoT device assembly, Coding and Back-end implementation were done effectively and there were few overlaps so as to achieve maximum productivity and save on time.

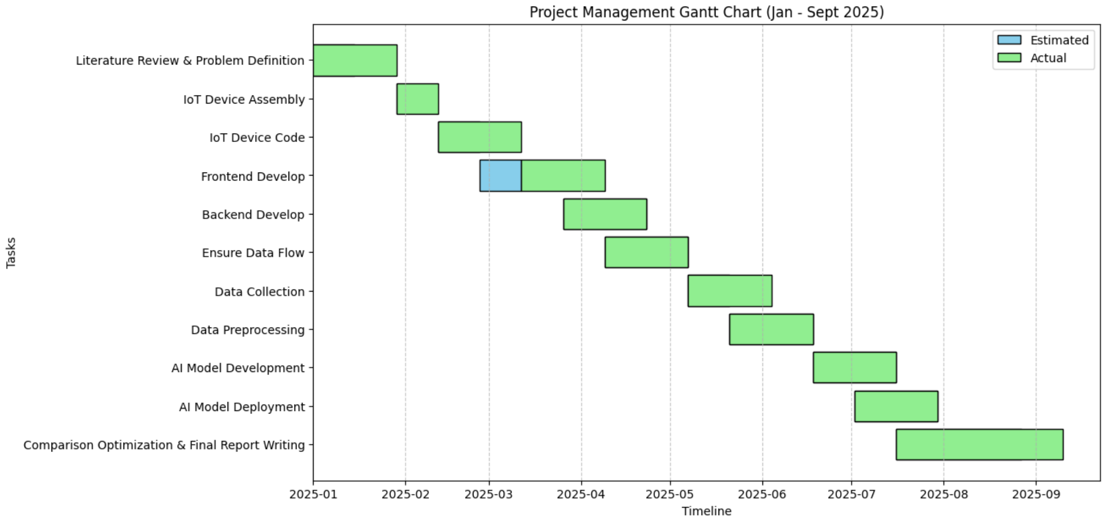


Figure 5.1: Project Management Gantt chart

Not all went as well as planned during the IoT Device Code development phase though. Although the Frontend Development was expected to start earlier it was delayed the IoT device coding phase taking longer and being more challenging then expected. This fortuitous failure notwithstanding, the task had been included within the overall timeline to happen without much influence on subsequent stages. Important tasks, including data collection, preprocessing, developing AI models, and deployment, were done on time, and, at the end of the project, the tasks involved comparison, optimization, and the final writing of the report were also finalized within the stipulated timeframe. This indicates high flexibility and time management with regard to solving emerging problems.

5.1.2 Financial Analysis

To assess the financial viability of the IoT-based blood pressure prediction system with the help of AI, the cost estimation was performed in detail. The analysis involves the hardware aspects, software subscriptions, data gathering, and supporting aids necessary to achieve positive implementation of the project. Table 5.1 presents the estimation of the costs.

Table 5.1: Financial Analysis table.

SN	Item or Component's	Quantity	Estimated Cost (BDT)
1.	ESP32 V1.3 Dev Board CH340C NodeMCU-32	2	900
2.	MAX30102 Heart Rate Sensor Module	3	1,100
3.	LCD Display 16x2	1	450
4.	I2C LCD Adapter Module	1	120

5.	4 pin Push Switch with Cap	1	50
6.	Breadboard Half Size Bare 400 Tie Points	2	700
7.	Breadboard Jumper Wire Set	3	150
8.	Acrylic Sheet	2	940
9.	Acrylic Sheet Lesser Cutting and Assembly of Encloser	~	1,310
10.	GEEPAS 18650 3.7V 3000mAh Lithium-ion Battery	1	280
11.	18650 Battery Holder – 1 Cell	1	80
12.	Type-C 2A Lithium Battery Fast Charge Module	1	230
13.	Dell Inspiron 15 5510, Intel Core i7-11390H (8-core), 16 GB RAM, Windows 11 Home 64-bit, and Others	1	98,000
14.	Google Colab Pro CPU Runtime & Others	~	8,800
15.	Power consumption for CPU computing	~	7,600
16.	Field data collection	~	7,000
17.	Omron Automatic Blood Pressure Monitor HEM-7120	1	6,000
18.	Web domain	~	6,000
19.	Miscellaneous	~	8,000
	Total Cost		147,710

According to the financial analysis of the project, the overall estimated cost is: BDT 183,860, which consists of the resources related to hardware and software purchase. The highest cost is Dell Inspiron 15 5510 laptop (BDT 98,000), almost two-thirds of the budget, because it offers computational capabilities required to train AI models and develop systems. Additional interesting expenses are field data collection (BDT 7,000), which does not indicate an assessment of the real work done; the dataset creation, and cloud computing with Google Colab Pro (BDT 8,800) that is complemented by the power energy used to make the CPU work longer hours (BDT 7,600). Although only one ESP32 boards and one MAX30102 sensors was used, but more then one was needed due to them shot-circuiting while prototyping, testing and data collection. On the other hand 2 Breadboard was needed to properly place the ESP32 and other components properly. Overall the IoT Components and enclosure fabrication are relatively cheap hardware components that only take up a small part of the budget. Omron BP monitor (BDT 6,000) was needed for dataset calibration, web domain for hosting (BDT 6,000), and miscellaneous costs (BDT

8,000) from the start till finish. On the whole, computation and data collection are the main investments in the budget.

5.4 Complex Engineering Problem

5.4.1 Complex Problem Solving

The proposed system is mapped to the seven categories of complex engineering problems (EP1–EP7). For each, a rationale is provided to highlight the relevance.

Table 5.2: Mapping with Complex Engineering Problem.

EP1 Dept of Knowledge	EP2 Range Of Conflicting Requirements	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent Of Stake-holder Involvement	EP7 Interdependence
✓		✓	✓			✓

EP1: The project needed to be aware of multiple fields such as electronics, biomedical signal processing, artificial intelligence, and software engineering knowledge. To construct the IoT device, it was necessary to learn physics of optical sensors and program microcontrollers, and preprocessing the signals of PPG required mathematical and statistical knowledge. To select, train, and optimize the models, machine learning knowledge was required. All these areas indicate that the problem could not be resolved in a single sphere of expertise, so EP1 requirements would have been fulfilled.

EP3: The project consisted of detailed analysis at several levels in order to create a functional BP prediction system. The selected waveforms of raw PPG were thoroughly studied by the time and frequency concepts by peak detection, skewness, kurtosis, and Fourier transformation. Statistical trends were analyzed to know their applicability to blood pressure. In addition to signal analysis, machine learning models were compared by quantitative assessment measures (RMSE and R2). This level of analysis outlines the multi-level analysis of the project as a subset of EP3.

EP4: Even though the application of PPG to predict cuffless blood pressure is a challenge issue reported in the biomedical literature, PPG is a clinical issue yet to be solved. Old problems like sensor noise, subject-to-subject variation and demographic variations in vascular response had to be considered. Meanwhile, a series of new problems was identified within the context of the real-time implementation of IoT hardware combined with web systems. Therefore, the project had to deal with known and unfamiliar issues, which is what meets the definition of EP4.

EP7: The solution had a high relational aspect with components. The accuracy of features requires reliable PPG data acquisition of the driving element, which is the IoT device.

These would have to be properly prestructured to feed machine learning models, the effectiveness of which directly influenced the user interface experience. Even a failure of any of the layers (either hardware, software, or AI) would lead to a failure of the overall system as well. Such interdependence of different subsystems is a good example of EP7 hence the problem is interdependent and complex at the same time.

Mapping with Knowledge Profile

This section is designed to map the overall problem and EP1 ***(multiple between K3, K4, K5, K6, K8 for attaining EP1)*** to the Knowledge Profile.

Table 5.2: Mapping with knowledge Profile.

K1 Natural Science	K2 Mathematics	K3 Engineering Fundamentals	K4 Specialist Knowledge	K5 Engineering Design	K6 Engineering Practice	K7 Comprehension	K8 Research Literature
	✓	✓	✓	✓	✓		✓

K2: This project heavily relied on mathematics that supported the preprocessing, as well as modeling. Calculation of Fourier transforms, skewness and kurtosis and variance were all based on mathematical formulations. The process of training machine learning models was a regression-based optimization problem, and their assessment (MAE and RMSE) was founded on mathematical formulas to assess the importance of error. Signal analysis, as well as performance evaluation of a model, could not have occurred without this mathematical background. Therefore, mathematics (K2) was inescapable as far as project results are concerned.

K3: Behind the design of the IoT-based devices and the communication system, there were the core engineering principles. Electrical circuits had to be understood, to connect the sensor, MAX30102 sensor, to ESP32 microcontroller. The concept of realizing embedded systems enabled real time signal capture, whereas the field of networking enabled the transfer of data wirelessly. Engineering principles were used to make sure the hardware worked in harmony with the backend and the web systems, which was the backbone of the whole solution. Therefore, the project very much coincides with K3.

K4: A key attribute to the success of the project was specialist knowledge. In order to create meaningful information out of noise in PPG waveforms biomedical signal processing knowledge was needed. High-tech AI concepts and ML concepts such as Random Forests and BiLSTM networks required expert computational expertise to run and optimize. Also, to provide real-time interaction between software and hardware, IoT systems expertise was required. This dependency on expert knowledge emphasises that the mapping of the project to K4 occurs.

K5: The project represented an example of an engineering design because it incorporated several subsystems into a solution. They needed more than just workable parts, but the system needed to be usable, efficient, and real time performance. This involved developing the casing of the IoT, organising the MongoDB structure and a visual data web interface. All design decisions weighed user requirements against technical feasibility, which is a characteristic feature of engineering design and mapping that is easily mapped to K5.

K6: In the invention of the iterative implementation and testing stage, the engineering practice was displayed. The hardware prototypes of the IoT device were refined a couple of times. Multiple versions of the data set were tested on preprocessing pipelines to test its robustness and AI models were trained, evaluated, and compared. The practical aspect of engineering work was embodied by debugging a bug in a code or a bug in a hardware. This embodiment of the theoretical design in practical working mechanisms is an expression of the K6 requirements.

K8: The type of research literature available was so valuable in the methodology and system design. The features and models were selected based on studies on cuffless BP estimation using PPG.[6][9][15]. IoT healthcare reviews also had an impact on the system architecture. Using peer-reviewed literature allowed making certain that the project was not another quiet study, but it can be traced back to and justified by previous studies, which, in turn, pointed to K8.

5.4.2 Engineering Activities

In this section, provide a mapping with engineering activities. For each mapping add subsections to put rationale (Use Table 5.3).

Mapping with Complex Engineering Activities

The project also aligns with the five categories of complex engineering activities (EA1–EA5). The mapping and rationale are provided below

Table 5.3: Mapping with Complex Engineering Activities.

EA1 Range of resources	EA2 Level of Interaction	EA3 Innovation	EA4 Consequences for society and environment	EA5 Familiarity
✓	✓	✓		

EA1: A large variety of resources were used in this project, physical (ESP32, MAX30102, LCD, Omron sphygmomanometer) as well as software (Python, TensorFlow, Node.js, MongoDB). Further, Google Colab was an environment with cloud-based model training. These mixed resources demanded technical flexibility, and close coordination to implement. The depth of resources is a measure of complexity in a project and illustrates EA1.

EA2: In this project, there was interaction on various levels. IoT interaction with the web interface was facilitated by backend APIs that connected the IoT device to the system-to-system interaction. The frontend interface was the method to provide human-computer interaction, enable entry of demographical data, and provide real-time visualization of predictions. Within the ML models, interactions were also possible as features interacted with each other to make predictions. The multidimensional interrelationship between and among systems depicts EA2.

EA3: The project innovation can be described as a combination of IoT hardware, signal processing using PPG, and machine learning in the form of a single real-time foretelling system. Although some of these individual components have been explored in the past, their integration to create a deployable and easy-to-use system that measures cuffless BP is new. In this strategy, we can see how engineering innovation can be achieved through new theories as well as through integrating the existing technologies into viable solutions thus fitting within EA3.

5.5 Summary

This chapter analyzes in detail compliance, impact, project management, and financial aspects of the proposed IoT-based blood pressure monitoring system. To improve the reliability and patient safety, the system design was responsible to prominent international conventions such as the IEC 62304 (software life cycle), IEC 60601 (hardware safety), and ISO 14971 (risk management). Standards of communication put data protection, and alignment with HIPAA principles and the GDPR in focus. Societal, environmental, and ethical impacts were also taken into account, with the benefits of the project including proactive management of hypertension and healthcare cost savings, and disadvantages, including e-waste, privacy, and bias in the algorithms being highlighted. There was a suggested sustainability plan involving technical, clinical, and environmental. Project management was done properly, there was slight setback on frontend development but there was good time management. Financial analysis showed a perceived cost of around BDT 183,860, and computational resources and data collection were the largest expenditures. Lastly, this project was a good fit to engineering intricate issues and endeavors that demonstrated multidisciplinary expertise, development, and substantial charitable results.

Chapter 6

Conclusion

This is the last chapter of this report, and here the overall summary, final thoughts and future work for this research is provided. In this chapter the limitations of this research and project is also stated. The overall final view on the whole study is provided in this chapter.

6.1 Summary

The paper examined the estimation of non-invasive blood pressure based on photoplethysmography (PPG) signal, demographic characteristics, and machine learning algorithms. The entire pipeline has been constructed and used that included preprocessing and interpolation of signal and then there was a massive process of extracting features. The features extracted consisted of morphological features (e.g. pulse width, rise / fall time, dicrotic notch), frequency domain features, entropy and nonlinear features, and heart rate variability features. A total of 85 features were developed and combined with demographics, including age, gender, height, weight, and body mass index (BMI).

Mutual Information (MI), Recursive Feature Elimination (RFE), and SHAP values were used to select features, and LightGBM was a feature ranking engine but not a predictive model. To perform the actual prediction task, a number of machine learning models were trained, from which two models, which were further optimized by hyperparameter fine-tuning, were identified as Random Forest and CatBoost. The two ensemble-based models have been selected based on their demonstrated ability to deal with small data and noisy features as well as nonlinear interactions.

With this strict methodology, the overall performance was below clinically acceptable levels. The most successful models were those that obtained mean errors (MAE) greater than 7 mmHg with R^2 less than 0.25. These outcomes indicate that the pipeline was both designed and implemented correctly, but the quality of the available data and hardware inherently constrained the accuracy of the predictions. The key conclusion of this paper is that the MAX30102 sensor, its low sample rate (approximately 5.6 Hz), poor waveform fidelity, and inaccurate physiological measurements cannot be sufficiently used to estimate BP.

6.2 Limitation

As Per the study had several significant limitations which affected the results:

I. Sensor Limitations: The MAX30102 PPG sensor caused noise and bias on the data because of its poor sampling resolution and low signal-to-noise ratio. This made it impossible to correctly detect systolic peaks, dicrotic notches, and other morphological characteristics suitable to estimate blood pressure.

II. Constraints on Data Set: The data set was limited (536 records) and biased, as approximately 70 percent of the sample were men and almost half of the participants were aged 20-25 years. This was not diverse, and diminished generalizability.

III. Calibration Gap: This study tried to predict calibration-free as opposed to many other earlier studies that use cuff-based calibration. Although the methodological importance of this decision, it only made the task more difficult and showed another weakness of the MAX30102 sensor.

IV. Sample Performance: Despite hyperparameter optimization of CatBoost and Random Forest, the models failed to lower MAE to less than 7 mmHg. This implies that model architecture has not been the primary constraint - the constraint has been sensor fidelity and quality of the datasets.

6.3 Future Work

As per the study above, there are a few suggestions on how to conduct new research:

I. Better Hardware: Use medical grade sensors (sampling rate ~100 Hz) and optical power to measure fine-scale PPG morphology.

II. Growth in Bigger Data: Compile bigger and more varied datasets that incorporate equal representation of both age groups, genders, and clinical populations to increase model generalizability.

III. Combination of Clinical and Demographic Characteristics: Add other variables like comorbidities, medication, and lifestyle to model the BP in a more holistic way.

IV. Firm Calibration-Free Methods: Find ways of avoiding or reducing recourse to cuff-based calibration but this time over populations.

V. Standardized Validation: The implementation of cuffless BP estimation clinical validation protocols that have been explicitly developed to estimate cuffless BP is encouraged, as highlighted in recent reviews.

VI. Data Privacy and Security: Future IoT-based BP monitoring systems should include robust encryption, strong authentication and adherence to HIPAA and GDPR laws.

References

[1] Argüello-Prada, E. J., & Castaño Mosquera, C. D. (2024). *Exploring supervised machine learning models to estimate blood pressure using non-fiducial features of the photoplethysmogram (PPG) and its derivatives*. ResearchGate.

[2] Chakraborty, A., Sadhukhan, D., Pal, S., & Mitra, M. (2020). PPG-based automated estimation of blood pressure using patient-specific neural network modeling. *Journal of Mechanics in Medicine and Biology*, 20(05), 2050037.

[3] Chandrasekhar, A., Yavarimanesh, M., Natarajan, K., Hahn, J. O., & Mukkamala, R. (2020). PPG sensor contact pressure should be taken into account for cuff-less blood pressure measurement. *IEEE Transactions on Biomedical Engineering*, 67(11), 3134–3140.

[4] Dai, D., Ji, Z., & Wang, H. (2024). Non-Invasive Continuous Blood Pressure Estimation from Single-Channel PPG Based on a Temporal Convolutional Network Integrated with an Attention Mechanism. *Applied Sciences*, 14(14), 6061.

[5] Chowdhury, M. H., Shuzan, M. N. I., Chowdhury, M. E., Mahbub, Z. B., Uddin, M. M., Khandakar, A., & Reaz, M. B. I. (2020). Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques. *Sensors*, 20(11), 3127.

[6] Kachuee, M., Kiani, M. M., Mohammadzade, H., & Shabany, M. (2017). Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring. *IEEE Transactions on Biomedical Engineering*, 64(4), 859-869.

[7] Li, Y., Harfiya, L. N., Purwandari, K., & Lin, Y. (2020). Real-Time Cuffless Continuous Blood Pressure Estimation Using Deep Learning Model. *2020 International Symposium on Computer, Consumer and Control (IS3C)*, 332-335.

[8] Li, Z., & He, W. (2021). A Continuous Blood Pressure Estimation Method Using Photoplethysmography by GRNN-Based Model. *Sensors*, 21(21), 7207.

[9] Liu, M., Po, L. M., & Fu, H. (2017). Cuffless blood pressure estimation based on photoplethysmography signal and its second derivative. *2017 E-Health and Bioengineering Conference (EHB)*, 555-558.

[10] Brophy, E., De Vos, M., Boylan, G., & Ward, T. (2021). Estimation of Continuous Blood Pressure from PPG via a Federated Learning Approach. *Sensors*, 21(18), 6311.

[11] Tarifi, B., Fainman, A., Pantanowitz, A., & Rubin, D. M. (2023). A Machine Learning Approach to the Non-Invasive Estimation of Continuous Blood Pressure Using Photoplethysmography. *Applied Sciences*, 13(6), 3955.

[12] Wong, M. K. F., Hei, H., Lim, S. Z., & Ng, E. Y. K. (2023). Applied machine learning for blood pressure estimation using a small, real-world electrocardiogram and photoplethysmogram dataset. *Mathematical Biosciences and Engineering*, 20(1), 975-997.

[13] Mohammadi, H., Tarvirdizadeh, B., Alipour, K., & Ghamari, M. (2025). Cuff-less blood pressure monitoring via PPG signals using a hybrid CNN-BiLSTM deep learning model with attention mechanism. *Scientific Reports*.

[14] Wang, W., Mohseni, P., Kilgore, K., & Najafizadeh, L. (2021). Cuff-Less Blood Pressure Estimation via Small Convolutional Neural Networks. *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*

(EMBC), 195-198.

[15] Slapničar, G., Mlakar, N., & Luštrek, M. (2019). Blood Pressure Estimation from Photoplethysmogram Using a Spectro-Temporal Deep Neural Network. *Sensors*, 19(15), 3420.

[16] Kurylyak, Y., Lamonaca, F., & Grimaldi, D. (2013). A Neural Network-based method for continuous blood pressure estimation from a PPG signal. *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 334-338.

[17] Xing, X., & Sun, M. (2016). Optical blood pressure estimation with photoplethysmography and FFT-based neural networks. *Biomedical Optics Express*, 7(8), 3007-3020.

[18] Dey, J., Gaurav, A., & Tiwari, V. N. (2018). InstaBP: Cuff-less Blood Pressure Monitoring on Smartphone using Single PPG Sensor. *2018 IEEE 10th International Conference on Communication Systems & Networks (COMSNETS)*, 400-405.

[19] Tjahjadi, H., Ramli, K., & Murfi, H. (2020). Noninvasive Classification of Blood Pressure Based on Photoplethysmography Signals Using Bidirectional Long Short-Term Memory and Time-Frequency Analysis. *IEEE Access*, 8, 16334-16344.

[20] Chen, S., Ji, Z., Wu, H., & Xu, Y. (2019). A Non-Invasive Continuous Blood Pressure Estimation Approach Based on Machine Learning. *Applied Sciences*, 9(19), 4091.

[21] Liu, Z., Zhou, B., Li, Y., Tang, M., & Miao, F. (2020). Continuous Blood Pressure Estimation From Electrocardiogram and Photoplethysmogram During Arrhythmias. *IEEE Access*, 8, 169624-169636.

[22] Zadi, A. S., Alex, R., Zhang, R., Watenpaugh, D. E., & Behbehani, K. (2018). Arterial Blood Pressure Feature Estimation Using Photoplethysmography. *arXiv preprint arXiv:1811.06039*.

[23] Montesinos, L., Martinez-Rios, E., Alfaro-Ponce, M., & Pecchia, L. (2021). A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data. *Artificial Intelligence in Medicine*, 118, 102125.

[24] Liu, S. H., Sun, C. K., Wu, C. C., Chen, Y. L., & Zhu, Y. S. (2025). Using machine learning models for cuffless blood pressure estimation with ballistocardiogram and impedance plethysmogram. *Frontiers in Digital Health*.

[25] El-Hajj, C., & Kyriacou, P. A. (2020). A review of machine learning techniques in photoplethysmography for the non-invasive cuff-less measurement of blood pressure. *Biomedical Signal Processing and Control*, 62, 102132.

[26] Mukkamala, R., Shroff, S. G., Kyriakoulis, K. G., Avolio, A. P., & Stergiou, G. S. (2025). Cuffless Blood Pressure Measurement: Where Do We Actually Stand?. Hypertension, 82(6), 957-970.

[27] Henry, B., Merz, M., Hoang, H., Abdulkarim, G., Wosik, J., & Schoettker, P. (2024). Cuffless Blood Pressure in clinical practice: challenges, opportunities and current limits. *Blood Pressure*, 33(1), 2304190.

[28] van der Velden, J. L. J., et al. (2024). Evaluation of a novel cuffless photoplethysmography-based wristband for measuring blood pressure according to the regulatory standards. *European Heart Journal - Digital Health*, 5(3), 335-343.

[29] Zhang, Y., & Feng, Z. (2017). A SVM Method for Continuous Blood Pressure Estimation from a PPG Signal. *Proceedings of the 2017 International Conference on Machine Learning and Computing*, 1-5.

[30] Firouzi, F., et al. (2021). Harnessing the power of smart and connected health to tackle COVID-19: IoT, AI, robotics, and blockchain for a better world. *IEEE Internet of Things Journal*, 8(16), 12826-12846.

[31] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[32] Draper, N. R. (1998). *Applied regression analysis*. McGraw-Hill. Inc.

[33] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[34] Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*(3), 199-222.

[35] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[36] Hasanzadeh, N., Ahmadi, M. M., & Mohammadzade, H. (2019). Blood pressure estimation using photoplethysmogram signal and its morphological features. *IEEE Sensors Journal*, *20*(8), 4300-4310.

[37] Mejía-Mejía, E., Budidha, K., Kyriacou, P. A., & Mamouei, M. (2022). Comparison of pulse rate variability and morphological features of photoplethysmograms in estimation of blood pressure. *Biomedical Signal Processing and Control*, *78*, 103968.

[38] Alghlayini, S., Al-Betar, M. A., & Atef, M. (2025). Enhancing non-invasive blood glucose prediction from photoplethysmography signals via heart rate variability-based features selection using metaheuristic algorithms. *Algorithms*, *18*(2), 95.

[39] Wu, H. (2024). Multiscale entropy with electrocardiograph, electromyography, electroencephalography, and photoplethysmography signals in healthcare: A twelve-year systematic review. *Biomedical Signal Processing and Control*, *93*, 106124.

[40] Naeini, E. K., Sarhaddi, F., Azimi, I., Liljeberg, P., Dutt, N., & Rahmani, A. M. (2023). A deep learning–based PPG quality assessment approach for heart rate and heart rate variability. *ACM Transactions on Computing for Healthcare*, *4*(4), 1-22.

[41] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, *31*.

[42] Hamid, M., Hajjej, F., Alluhaidan, A. S., & bin Mannie, N. W. (2025). Fine tuned CatBoost machine learning approach for early detection of cardiovascular disease through predictive modeling. *Scientific Reports*, *15*(1), 31199.

Non-Invasive Blood Pressure Estimation Using Photoplethysmography and IoT-ML Integration

ORIGINALITY REPORT

17%	14%	12%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	www.mdpi.com Internet Source	1%
3	link.springer.com Internet Source	1%
4	123dok.com Internet Source	1%
5	Submitted to United International University Student Paper	<1%
6	www.nature.com Internet Source	<1%
7	www.coursehero.com Internet Source	<1%
8	assets-eu.researchsquare.com Internet Source	<1%
9	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
10	ruor.uottawa.ca Internet Source	<1%
11	Submitted to Heriot-Watt University Student Paper	<1%
12	Submitted to Asia e University Student Paper	