ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AUTOMATED DIAGNOSTIC TOOL FOR HYPERTENSION USING DEEP LEARNING MODEL

M.Sc. Thesis by

Tuğba YILMAZ

Department of Computer Engineering

October, 2021

ANKARA

AUTOMATED DIAGNOSTIC TOOL FOR HYPERTENSION USING DEEP LEARNING MODEL

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Department of Computer Engineering

by

Tuğba YILMAZ

October, 2021

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"AUTOMATED DIAGNOSTIC TOOL FOR HYPERTENSION USING DEEP LEARNING MODEL"** completed by **TUĞBA YILMAZ** under the supervision of **ASSOC. PROF. DR. BAHA ŞEN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Baha ŞEN

Supervisor

Assoc. Prof. Dr. Fatih NAR

Jury Member

Prof. Dr. İlyas ÇANKAYA

Jury Member

Prof. Dr. Sadettin ORHAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 01.10.2021 **Signature: ……...**

Name & Surname: Tuğba YILMAZ

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Assoc. Prof. Dr. Baha ŞEN for his tremendous support and motivation during my study. His immense knowledge and precious recommendations constituted the milestones of this study. His guidance assisted me all the time of my research and while writing this thesis.

I would like to express my greatest and deepest gratitude for my dear father Seyhan Yılmaz, my dear mother Reyhan Yılmaz and my dear brother Tolgahan Yılmaz for their support, motivation and understanding during my study.

Finally, I would like to thank my friends for their support and motivation.

2021, 01 October **Tuğba YILMAZ**

AUTOMATED DIAGNOSTIC TOOL FOR HYPERTENSION USING DEEP LEARNING MODEL

ABSTRACT

Hypertension is a systemic disease manifested by persistent high blood pressure, it is an important health problem because it causes serious complications and is common in the society. Untreated hypertension has been shown to increase the rate of heart failure, coronary heart disease, hemorrhagic and thrombotic stroke, renal failure, peripheral artery disease, aortic dissection, and mortality. Complications of hypertension and related mortality increase in direct proportion to high blood pressure. The purpose of the detection and treatment of hypertension is to reduce the risk of heart, brain, vascular and kidney diseases, and the associated mortality rate. Therefore, it requires a continuous, accurate blood pressure measurement system to identify high risk individuals.

In the health sector, an uncomfortable blood pressure holter device is usually attached to the patient for the diagnosis of high blood pressure disease. The aim of the study is to present algorithms for blood pressure prediction that do not occlude continuous and non-invasive vascular access. There are many studies in the literature based on machine learning algorithms for blood pressure prediction, but this study has provided estimation of systolic blood pressure (SBP), diastolic blood pressure (DBP) and mean arterial pressure (MAP) values by focusing on deep learning algorithms. It was aimed to make blood pressure estimates with the features extracted from Electrocardiography (ECG) and Photoplethysmography (PPG) signals.

The feature set consists of ECG-based features such as Womersley number, SDI, QRS and QT_c interval, and PPG-based features such as photoplethysmograph intensity ratio (PIR). The feature set is used to minimize the error in blood pressure estimation and to explore the relationship of these features with blood pressure. The extracted features were trained with deep learning algorithms such as convolutional neural networks and long short-term memory. Evaluation measures such as mean absolute error (MAE), root mean square error (RMSE), mean square error (MSE) and variance score were also calculated.

In the blood pressure estimation of our model, which we obtained by using convolutional neural networks and long-short-term memory algorithms together, it was concluded that the accuracy rate increased. At the end of this thesis, an automated version of blood pressure estimation from electrocardiography and photoplethysmography signals has been implemented, allowing the fastest and most accurate diagnosis of hypertension. The methods used in the noninvasive blood pressure estimation we proposed, by integrating into wearable devices that are very popular in today's technology it can also be used to combat cardiovascular diseases and prevent their risky effects.

Keywords: Hypertension, electrocardiography, photoplethysmography, blood pressure, convolutional neural networks (CNN), long short-term memory (LSTM), deep learning.

DERİN ÖĞRENME MODELİNİ KULLANARAK HİPERTANSİYON İÇİN OTOMATİK TEŞHİS ARACI

ÖZ

Hipertansiyon, sürekli kan basıncı yüksekliği ile kendini gösteren, sistemik bir hastalık olup, ciddi komplikasyonlara neden olması ve toplumda yaygın olarak görülmesi nedeniyle önemli bir sağlık problemidir. Tedavi edilmeyen hipertansiyonun, kalp yetersizliği, koroner kalp hastalığı, hemorajik ve trombotik inme, böbrek yetersizliği, periferik arter hastalığı, aort diseksiyonu ve ölüm oranını artırdığı ortaya konmuştur. Hipertansiyonun komplikasyonları ve buna bağlı ölüm oranı, kan basıncı yüksekliği ile doğru orantılı olarak artmaktadır. Hipertansiyonun belirlenmesinin ve tedavisinin amacı kalp, beyin, damar ve böbrek hastalıkları riskini ve eşlik eden morbidite ile ölüm oranını azaltmaktır. Bu nedenle, yüksek riskli bireyleri belirlemek için sürekli, doğru bir kan basıncı ölçüm sistemi gerekmektedir.

Sağlık sektöründe genellikle yüksek tansiyon hastalığının teşhisi için hastaya konforsuz yapıda olan tansiyon holter cihazı takılmaktadır. Yapılan çalışmanın amacı sürekli ve noninvaziv damar yolunu tıkamayan kan basıncı tahminine yönelik algoritmalar sunmaktır. Literatürde kan basıncı tahminine yönelik makine öğrenmesi algoritmalarına dayalı birçok çalışma bulunmaktadır, ancak bu çalışma derin öğrenme algoritmalarına odaklanarak sistolik kan basıncı, diyastolik kan basıncı ve ortalama arteriyel basınç değerlerinin tahminini sağlamaktadır. Elektrokardiyografi (EKG) ve Fotopletismografi (PPG) sinyallerinden çıkarılan öznitelikler ile kan basıncı tahminlerinin gerçekleştirilmesi hedeflenmiştir.

Öznitelik seti, Womersley numarası, SDI, QRS ve QT_c aralığı gibi EKG tabanlı ve fotopletismograf yoğunluk oranı (PIR) gibi PPG tabanlı özelliklerden oluşmaktadır. Öznitelik seti, kan basıncı tahminindeki hatayı en aza indirmek ve bu özelliklerin kan basıncı ile ilişkisini araştırmak amacıyla kullanılır. Çıkarılan öznitelikler evrişimli sinir ağları ve uzun kısa süreli bellek gibi derin öğrenme algoritmaları ile eğitilmişlerdir. Ortalama mutlak hata (MAE), kök ortalama kare hatası (RMSE), ortalama kare hata (MSE) ve varyans gibi değerlendirme ölçümleri de hesaplanmıştır.

Evrişimli sinir ağları ve uzun kısa süreli bellek algoritmalarını birlikte kullanarak elde ettiğimiz modelimizin kan basıncı tahmininde doğruluk oranının arttığı gözlemlenmiştir. Bu tezin sonunda, elektrokardiyografi ve fotopletismografi sinyallerinden kan basıncı tahmininin otomatik bir versiyonu uygulanarak hipertansiyonun en hızlı ve doğru şekilde teşhis edilmesine olanak sağlanmıştır. Önerdiğimiz noninvaziv kan basıncı tahmininde kullanılan yöntemler günümüz teknolojisinde oldukça popüler olan giyilebilir cihazlara entegre edilerek kardiyovasküler hastalıklarla mücadele etmek ve riskli etkilerini önlemek amacıyla da kullanılabilir.

Anahtar Kelimeler: Hipertansiyon, elektrokardiyografi, fotopletismografi, kan basıncı, evrişimli sinir ağları, uzun kısa süreli bellek, derin öğrenme.

CONTENTS

NOMENCLATURE

Acronyms

AAMI	Association for the Advancement of Medical Instrumentation
ABP	Arterial Blood Pressure
AI	Augmentation Index
ANN	Artificial Neural Network
ASCII	American Standard Code for Information Interchange
BCG	Ballistocardiogram
BHS	British Hypertension Society
BiLSTM	Bidirectional Long Short-Term Memory
BP	Blood Pressure
BRNN	Bidirectional Recurrent Neural Networks
BRP	Bioengineering Research Partnership
BTT	Back-Propagation Through Time
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DBN	Deep Belief Networks
DBP	Diastolic Blood Pressure
DNS	Domain Name System
DT	Diastolic Time
DWT	Discrete Wavelet Transform
ECG	Electrocardiography
ESC	European Society of Cardiology
ESH	European Society of Hypertension
GRU	Gated Recurrent Unit
HR	Heart Rate
HTT	Hilbert - Huang transform
ICU	Intensive Care Unit
IPA	Inflection Point Area Ratio
ISO	International Organization for Standardization
KNN	K-Nearest Neighbors
LASI	Large Artery Stiffness Index

LASSO	Least Absolute Shrinkage and Selection Operator
LED	Light Emitting Diode
LRCN	Long-Term Recursive Convolutional Network
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAP	Mean Arterial Pressure
MIMIC	Multi Parameter Intensive Care Intelligent Monitoring
ML	Machine Learning
MLR	Multiple Linear Regression
MSE	Mean Square Error
NIH	National Institutes of Health
NLP	Natural Language Processing
PAT	Pulse Arrival Time
PCA	Principal Component Analysis
PIR	Photoplethysmography Intensity Ratio
PPG	Photoplethysmography
PTT	Pulse Transit Time
RELU	Rectified Linear Units
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RR	Respiratory Rate
SBP	Systolic Blood Pressure
SQL	Structured Query Language
SUT	Systolic Upstroke Time
SVM	Support Vector Machine
SVR	Support Vector Regression
TD	Temporal Difference
WHO	World Health Organization

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Hypertension, known as blood pressure, is higher than normal values, is the most important preventable cause of death according to the reports of the World Health Organization (WHO) [1]. In national-scale large studies, it has been shown that one out of every three people in adults over the age of 18 and one out of every two people from the age of 50s have hypertension.

Hypertension complications cause 9.4 million deaths worldwide each year. Hypertension is responsible for 45% of deaths due to heart diseases and 51% of deaths due to stroke. Hypertension is a common and important public health problem in Turkey as well as all over the world. According to the data of the Ministry of Health, one of every four deaths in Turkey is caused by hypertension. Therefore, it is very important to recognize and treat hypertension early. So much so that a decrease of 2 mmHg in systolic blood pressure, known as high blood pressure, provides a 7% reduction in deaths due to coronary artery disease and a 10% reduction in deaths due to stroke. However, hypertension is a preventable and treatable disease.

While genetic predisposition and excessive salt consumption are among the causes of hypertension, the cause of high blood pressure is not known in most patients. Hypertension with no identifiable cause is called primary hypertension. This type of hypertension tends to develop gradually over time. In some rarer cases, hypertension is caused by an underlying medical condition. This type of hypertension, called secondary hypertension, tends to occur suddenly, and causes higher blood pressure values than the primary type of hypertension. Medical conditions that cause this condition include parenchymal kidney disease, adrenal gland tumors, congenital defects in blood vessels, renal artery stenosis, obstructive sleep apnea, and thyroid problems, and excessive aldosterone release.

Table 1.1 Reference blood pressure values according to ESH/ESC

Classification	**SBP (mmHg)**	**Operation (and/or)**	**DBP (mmHg)**
Optimal	< 120	and	< 80
Normal	120 to 129	and/or	80 to 84
Normal High	130 to 139	and/or	85 to 89
Hypertension grade 1	140 to 159	and/or	90 to 99
Hypertension grade 2	160 to 179	and/or	100 to 109
Hypertension grade 3	≥ 180	and/or	> 110
Isolated Systolic Hypertension	≥ 140	and	< 90

The purpose of the determination and treatment of hypertension is to reduce the risk of heart, brain, vascular, and kidney diseases and the associated morbidity and mortality rate. Therefore, blood pressure classifications for adults should be made under the leadership of constantly updated guidelines to identify high-risk individuals and to provide follow-up and treatment principles. In adults (> 18 years of age), systolic blood pressure greater than or equal to 140 mmHg and/or diastolic blood pressure greater than or equal to 90 mmHg is defined as hypertension (high BP) according to the standard measurement performed by the physician [2]. BP values below 120/80 mmHg are considered normal. Systolic BP is particularly important and essential in diagnosis in most patients. It is reported [3] that systolic BP is acceptable up to 150 mmHg in those who are greater than or equal to 80 years old.

Classification according to the BP level in the general population is shown in Table 1.1 according to European Society of Hypertension (ESH) and European Society of Cardiology (ESC) guidelines. For the diagnosis of hypertension in adults, an electrocardiogram, echocardiography, 24-hour blood pressure monitoring, and a pulse count not shorter than 30 seconds are performed. In addition, a detailed medical history should be taken, systemic physical examination and necessary laboratory

examinations should be performed to determine the risk factors of the patient and to question the causes of secondary hypertension.

1.1 Purpose of Thesis

The aim of the study is to provide algorithms for blood pressure prediction that do not occlude continuous and non-invasive vascular access. In the study, we aimed to predict systolic blood pressure (SBP), diastolic blood pressure (DBP), and mean arterial pressure (MAP) values with the features extracted obtained from Electrocardiography (ECG) and Photoplethysmography (PPG) signals by focusing on deep learning algorithms.

The feature set is used to minimize the error in blood pressure estimation and to explore the relationship of these features with blood pressure. The extracted features are trained with deep learning algorithms such as convolutional neural networks and long short-term memory. At the same time, the properties are trained by combining the two algorithms. Then, we aimed to compare the performance by calculating evaluation measures, namely mean absolute error (MAE), root mean square error (RMSE), mean square error (MSE) and variance score.

1.2 Motivation

Hypertension, which is defined as high blood pressure, can cause serious diseases, organ damage, and even death, if necessary, precautions are not taken. For this reason, the diagnosis and treatment of hypertension are of great importance in terms of preventing organ damage and accompanying diseases. Accurate measurement of blood pressure (BP) is very important for the diagnosis of hypertension. It requires a continuous and accurate blood pressure measurement system to identify high-risk individuals. In the health sector, a blood pressure holter device, which is uncomfortable, is usually attached to the patient for the diagnosis of high blood pressure disease. In addition, a detailed physical examination, electrocardiogram, echocardiography, 24-hour blood pressure monitoring, and laboratory tests are also performed under expert control.

Considering the need for objective measurement techniques to diagnose such a widespread disease, which is increasing day by day, the development of a computerized diagnostic system will assist in the decision-making processes of clinicians. In this way, an automatic diagnosis system is provided with easy user-friendly software that facilitates the work of specialists, reduces the workload, and speeds up the evaluation and treatment processes. In addition, patients are provided with the opportunity to make continuous and non-invasive blood pressure measurements that do not block vascular access, without the use of a holter device. At the end of this thesis, an automated version of blood pressure estimation from electrocardiography and photoplethysmography signals has been implemented, allowing the fastest and most accurate diagnosis of hypertension.

1.3 Outline

The paper is organized in the following manner: Chapter 1 contains the purpose and the scope of the research. It also has an overview of the literature review. Chapter 2 has information about machine learning and deep learning methods used in the study are explained in detailed information regarding and also describes the dataset. Chapter 3 presents the experimental study carried out in the study, the evaluation procedures used, and the experimental results obtained. Finally, Chapter 4 describes the conclusions accessed from the study and some ideas intended for future work.

1.4 Related Works

In the literature, there are some studies about medical diagnosis using different learning methodologies for blood pressure estimation. This section presents literature reviews separately for blood pressure prediction using machine learning algorithms and deep learning algorithms from electrocardiography and photoplethysmography signals. In the first part of our literature study, we have conducted a literature review considering the studies in blood pressure prediction using machine learning algorithms. Afterward, we examined studies involving blood pressure prediction using deep learning algorithms.

1.4.1. Related Works Using Machine Learning Algorithms

Thambiraj, Gandhi, Mangalanathan, Jose, and Anand have conducted a study on an algorithm based on new time-domain features for continuous blood pressure monitoring which is critical in intensive care units and can be used to estimate cardiovascular diseases [4]. They have aimed to estimated blood pressure continuously that extracts informative features like Womersley number (α), QRS, QTc interval, SDI from ECG, and PPG signals using regression machine learning techniques. They have studied various machine learning models like linear regression, ridge regression, support vector regression, adaboost regression, random forest regression and have considered performance metrics like root mean squared error (RMSE), mean absolute error (MAE), Mean squared error (MSE), bias. Also, they have calculated a 95% Confidence Interval (CI), Intraclass Correlation Coefficient (ICC), Correlation Coefficient (r) to validate the proposed method. It is perceived that the MAE and RMSE of the random forest model are relatively better for different feature combinations. It is also obvious that ECG features along with PTT provide the least MAE and RMSE for the RF model compared to other combinations, which explains better BP evaluation. MAE, bias, and better correlation coefficient are examined for the feature set that contains ECG features. The 95% CI of bias shows the wider interval range for the feature set without ECG compared with the feature set include ECG represents its importance in BP prediction.

As a result, they have used a genetic algorithm (GA) with the random forest model to investigate the relationship of proposed physiological features with blood pressure. Alpha, QRS complex, QT interval, SDI, and heart rate are among significant features in blood pressure prediction. GA has reduced the MAE from 13.20 to 9.54 mmHg, 9.91 to 5.48 mmHg, and 7.71 to 3.37 mmHg for SBP, DBP, and MBP respectively from the best optimal feature set. The following Figure 1.1 shows the comparison results of MAE and RMSE on various ML algorithms they have achieved.

BP	Feature set	Machine Learning Techniques									
		Linear Regression		Ridge Regression		Support Vector Regression		Adaboost Regression		Random Forest Regression	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SBP	ALL	14.05	17.78	14.23	18.04	14.85	18.92	14.20	17.17	10.83	14.96
	PTT + PPG features	15.27	18.90	15.19	18.94	15.14	19.30	15.63	19.17	14.14	18.18
	Physiological features	15.13	18.84	15.10	18.90	14.70	19.04	16.90	20.07	11.89	15.88
	PTT + ECG features + alpha	15.06	18.91	15.08	18.81	14.94	19.07	16.18	19.44	11.16	15.00
DBP	ALL	10.76	13.92	10.94	13.92	10.98	14.41	11.99	14.32	8.43	11.62
	PTT + PPG features	11.37	14.38	11.41	14.31	11.17	14.61	12.52	14.98	10.58	13.73
	Physiological features	11.06	14.05	11.40	14.30	11.08	14.51	11.81	14.36	9.38	12.54
	PTT + ECG features + alpha	11.35	14.33	11.21	14.12	11.05	14.47	12.11	14.45	8.55	11.84
MBP	ALL	10.44	13.28	10.59	13.30	10.70	13.76	10.95	13.07	8.35	11.05
	PTT + PPG features	11.19	13.86	11.14	13.77	10.95	14.01	12.06	14.36	10.22	13.12
	Physiological features	10.86	13.58	11.09	13.76	10.77	13.95	11.27	13.55	9.11	11.91
	PTT + ECG features + alpha	11.08	13.79	10.95	13.60	10.75	13.83	11.96	14.10	8.45	11.33

Figure 1.1 Comparison of performance of various ML methods using different feature set [4]

Figure 1.2 shows the RF model performance with and without ECG features and alpha. MAE, bias, and better correlation coefficient are examined for the feature set that contains ECG features.

BP	Feature set	MAE	RMSE	ICC‡	r‡
SBP	All features	10.83	14.58	0.70 (95% CI: 0.64–0.73)	0.64
	All features without ECG features and alpha	11.53	15.32	0.60 (95% CI: 0.55–0.66)	0.54
DBP	All features	8.40	11.01	0.66 (95% CI: 0.60–0.70)	0.61
	All features without ECG features and alpha	9.22	11.92	0.57 (95% CI: 0.50–0.62)	0.51
MBP	All features	8.25	10.77	0.65 (95% CI: 0.60–0.70)	0.61
	All features without ECG features and alpha	8.86	11.32	0.57 (95% CI: 0.53–0.65)	0.56

‡ICC – Intra class correlation coefficient. ICC values less than 0.5 are indicative of poor reliability, values between 0.5 and 0.75 indicate moderate reliability, values between 0.75 and 0.9 indicate good reliability, and values greater than 0.90 indicate excellent reliability.
‡r – Pearson Correlation Coefficient, CI‡ – Confidence Interval.

Figure 1.2 Performance comparison of feature set with and without ECG features for RF model [4]

Kachuee, Kiani, Mohammadzade, and Shabany have performed a study on cuffless blood pressure estimation algorithms for continuous healthcare monitoring [5]. They have aimed to study based on pulse arrival time (PAT) for continuous and cuffless prediction of systolic blood pressure, diastolic blood pressure (DBP), and mean arterial pressure (MAP) values using regression algorithms. In this study, the Physionet's multiparameter Intelligent Monitoring in Intensive Care (MIMIC) II (version 3, accessed in September 2015) database has used. This dataset includes ECG and PPG signals as well as the arterial blood pressure (ABP) signal. In this study, ECG and PPG signals as the primary inputs of the algorithm have been pre-processed to remove artifacts. It has provided to extract informative properties from pre-processed signals. After these processes, the dimension of the extracted features has been reduced.

Parameter-based features have been extracted from the shape of PPG signals and adapted from various cardiovascular parameters. Examples of these features are PAT, HR, AI, Large Artery Stiffness Index (LASI) and Inflection Point Area Ratio (IPA). Whole-Based Features are representations of the time domain signal in a specific interval. These features have been obtained by cropping the appropriate peaks selected from the ECG and PPG signals and shifting them accordingly. They have studied various learning algorithms for blood pressure prediction using these features. Comparison of the performance using learning algorithms in Figure 1.3.

	Systolic Blood Pressure (mmHg)				Diastolic Blood Pressure (mmHg)			
Feature Set	Parameter-based		Whole-based		Parameter-based		Whole-based	
	MAE	STD	MAE	STD	MAE	STD	MAE	STD
Linear Regression	14.71	10.79	14.14	10.44	6.74	6.11	6.75	6.12
Decision Tree	16.28	16.28	17.15	14.97	7.75	8.54	8.44	9.17
Support Vector Machine	12.26	10.32	12.65	10.33	5.91	5.78	6.19	6.07
AdaBoost	**11.17**	10.09	**11.87**	10.30	**5.35**	6.14	**5.78**	6.61
Random Forest	11.80	**9.87**	12.39	**10.09**	5.83	**5.71**	6.39	**6.06**

Figure 1.3 Comparison of the performance using the two feature sets and various algorithms [5]

As a result of, they have predicted systolic blood pressure, diastolic blood pressure (DBP), and mean arterial pressure (MAP) using calibration-free regression and optional calibration methods. They have recommended calibration methods as optional and complement parts of the system, as it improves the accuracy of cuffless BP estimation. They have evaluated the predictive accuracy of the algorithm using the British Hypertension Society (BHS) Standard and Association for the Advancement of Medical Instrumentation (AAMI) Standard. According to BHS, among the proposed calibration-free methods, results with grade A in DBP estimation and grade B in average BP estimation have been obtained. According to the AAMI, the results for the DBP and MAP have been confirmed by a significant difference. In addition to

the calibration-free method, a calibration treatment is also proposed to increase the prediction accuracy also.

In another study [6], Zhang and Feng have produced a study on a Support Vector Machine (SVM) method for continuous blood pressure estimation from a PPG signal. The study shows that there is not always a linear relationship between the blood pressure obtained from the pulse duration and the photoplethysmography (PPG) signal. Data have been taken from The University of Queensland Vital Signs Dataset for better display of possible pulse and pressure diversity. They have extracted nine different parameters by using heartbeats for analysis and later defined these features as the input vector for training. They have proposed a practical method of measuring human blood pressure based on PPG. In addition, they have stated that the SVM method can be used directly for BP regression analysis. As a result, the comparison between predicted and reference values showed better accuracy than the linear regression method. Also, they have stated that the SVM method compared to the ANN method is of great importance in mobile wearable devices in diastolic blood pressure measurements.

Paviglianiti, Randazzo, Pasero, and Vallan have carried out a study on noninvasive arterial blood pressure estimation using ABPNet. In this study [7], a two-hidden layer neural network called ABPNet was applied to estimate systolic and diastolic ABP values. In addition, the proposed model has been tested in the MIMIC database and validated with gold standards in both invasive and noninvasive approaches. They have explained that the RMSE and Pearson coefficient of ABPNet performance outperformed the standard sphygmomanometer and show higher reliability than ordinary non-invasive pressure sensing methods. They have noted that the proposed noninvasive blood pressure method can be placed in wearable devices such as the VITAL-ECG and used to struggle with cardiovascular disease and prevent its risky effects.

Similar to this study, Tanveer and Hasan have proposed a study that cuffless continuous blood pressure estimation from electrocardiography and photoplethysmography signals using a novel waveform-based hierarchical Artificial Neural Network-Long Short-Term Memory (ANN-LSTM) model [8]. The proposed

model consists of two hierarchy levels. The lower hierarchy level uses ANNs to extract required morphological features from ECG and PPG waveforms. The upper hierarchy level provides LSTM layers to account for the time domain variation of features extracted by the lower hierarchy level. Furthermore, this model can extract the required features without requiring any feature engineering and can learn the changes of the features over time. In this work, the proposed model was applied on 39 subjects using the Association for the Advancement of Medical Instruments (AAMI) standard and the British Society for Hypertension (BHS) standard and satisfied both the standards in the prediction of systolic blood pressure (SBP) and diastolic blood pressure (DBP). For this model, the mean absolute error (MAE) and the root mean square error (RMSE) for SBP prediction are 1.10 and 1.56 mmHg, respectively, and for DBP prediction are 0.58 and 0.85 mmHg, respectively. According to the BHS standard, the blood pressure prediction quality for both SBP and DBP has obtained grade A. They have stated that the performance of the proposed model hierarchical ANN-LSTM model is found to be better than the other traditional feature engineering-based methods. Also, it has been shown that this model is capable of automatically extracting the properties required to reliably predict BP in a non-invasive continuous manner and their time history variations. Finally, they have emphasized that the method is expected to greatly facilitate the mobile healthcare tools available in the continuous BP prediction.

Ghosh et al. [9] have examined the cuff-less and continuous blood pressure prediction from Pulse Transit Time (PTT) using ECG and PPG signals during variations in posture and activity. PTT has been reported to be highly correlated with blood pressure. In this paper, to demonstrate the importance of truth PTT calculation, with and without sparsification blood pressure estimates have been made. Constant or average SBP and DBP values cannot be used since BP values fluctuate from minute to minute. Accuracy has increased when sparsified. For this reason, they have recommended the use of the sparsification method when estimating BP using PTT. The linear regression algorithm estimated the BP from PTT to some degree better during recumbent, seated, and standing positions but, walking and cycling activity making it difficult to accurately identify desired peaks. Finally, they have started the BP prediction based on PTT was reasonably accurate when steady, but not during motion.

In the study titled "Cuff-Free Blood Pressure Estimation Using Pulse Transit Time and Heart Rate," Wang et al. have carried out a study on proposes, justifies, and validates the use of a combination of the PTT, HR, and BP n-J to estimate the current BP value [10]. In this study on the signals extracted from the MIMIC database, they have shown that there are significant correlations between physiological variables and that taking advantage of these correlations can significantly improve the BP estimation.

Kurylyak, Lamonaca, and Grimaldi have performed a study on a neural network-based method for continuous blood pressure estimation from a PPG signal [11]. In this paper, a new method for continuous and non-invasive blood pressure prediction from the PPG signal has been proposed. In the study, a feed-forward ANN algorithm with two hidden layers, 35 neurons in the first layer and 20 neurons in the second layer, has used. They have stated that this type of configuration showed better results in terms of performance compared to other architectures. They have used the MIMIC database for the training of ANN's algorithm. In total, they have analyzed more than 15000 heartbeats and extracted 21 parameters from each of them that define the input vector for the ANN. The results have obtained are 3.80 ± 3.46 mmHg for systolic pressure and 2.21 ± 2.09 mmHg for diastolic pressure. Finally, the study has explained that the comparison between predicted, and reference values showed better accuracy than the linear regression method. In addition, they have demonstrated compliance with the American National Standards of the Association for the Advancement of Medical Instrumentation, where the maximum accepted error is 5 ± 8 mmHg.

In another study [12], Sahoo, Manimegalai, and Thanushkodi have focused on wavelet-based pulse rate and blood pressure estimation systems from ECG and PPG signals. They have aimed to design a noninvasive cuff-less blood pressure and pulse rate prediction system based on pulse transit time (PTT) technique using waveforms. This study has presented a new algorithm and real-time application details to find the pulse rate and PTT using discrete wavelet transform (DWT) in MATLAB and LabVIEW. Wavelet estimation and thresholding methods have been implemented and compared. By discarding all the high-frequency information in the wavelet approximation method, most of the sharpest features of the original signal may be lost, so the wavelet thresholding method has proved to be a better method involving

discarding only the portion of the details that exceed a certain boundary. They have proved that this wavelet approach is an effective method for ECG and PPG signal processing, and they accurately obtained the pulse rate from the PPG signal. They have stated that it is possible to continuously monitor BP with PTT which is a physiological parameter inversely proportional to pulse wave velocity and BP, is inversely proportional to pulse wave velocity and BP.

Choi, Zhang, and Ko have investigated a potential cuffless non-invasive blood pressure prediction method based on pulse transit time (PTT) between electrocardiogram (ECG) and photoplethysmography (PPG) using Hilbert - Huang transform (HHT) algorithm [13]. The relationship between blood pressure and PTT has been demonstrated, and calibration and recalibration issues have also been discussed in this article. This algorithm has been tested based on continuous data from the MIMIC database and accuracy was measured to validate the proposed method. Both ECG and PPG data have been analyzed with this algorithm before obtaining PTT values with various measurement points such as PTT peak, PTT-foot, and PTT-mid. They have explained that based on the derived PTT values, it performed better in all respects to PTT-peak, error mean, error standard deviation, and correlation coefficient. The results have obtained are SBP: mean = 0.44 mmHg, SD = 3.85 mmHg, r = 0.71, DBP: mean = 0.93 mmHg, SD = 1.84 mmHg, r = 0.69. In addition, they have found that blood pressure and PTTs were inversely proportional except DBP and PTT-foot.

Sharif et al. have proposed a new method based on pulse transit time (PTT) and photoplethysmography intensity ratio (PIR) for the estimation of DBP, MBP, and SBP [14]. The proposed model includes signal preprocessing, parameter extraction, state-space reconstruction, and regression. They have predicted blood pressure by taking advantage of both modeling and prediction approaches and have used the cardiopulmonary system to increase the accuracy of the prediction. Their results showed the dependency of BP values to the final dynamical state that can be presented through PTT and PIR indices. This study includes comparative studies on a large standard data set. The authors argue that the proposed model performed better than the recent and quoted algorithms with improved accuracy.

Yang et al. have proposed a novel method for BP prediction using complexity analysis to extract features from PPG and ECG signals [15]. In total, 129 different datasets have been adopted for preprocessing and these served as inputs for signal complexity analysis resulting in the extraction of 90 features from signals. They have focused on the prediction of DBP and SBP using regression models that were ANN, SVM, and LASSO. This study claimed that the use of signal complexity analysis significantly improved prediction accuracy by extracting features from a combination of both signals, as well as extracting features with relatively high predictive capabilities from PPG or ECG signals.

Simjanoska et al. have performed a study on non-invasive blood pressure estimation from ECG using machine learning techniques [16]. They have developed a method for blood pressure prediction using only electrocardiogram signals. They have created 3129 30-second ECG segments using ECG recordings from 51 different subjects and extracted seven features from the ECG. The raw ECG data was filtered and segmented, followed by a complexity analysis for feature extraction. The regression techniques method was applied to create blood pressure, prediction models. In addition to these techniques, the method has allowed a probability distribution-based calibration to adapt the models to a particular user. The contribution of this study established a design relationship between blood pressure and ECG in the form of machine learning models. Finally, they have stated that believe this proposed solution has promising real-world applications in civil and military settings, as the trend of using ECG sensors indicates a steady increase in demand.

In the study titled "Estimating Blood Pressure from the Photoplethysmogram Signal and Demographic Features Using Machine Learning Techniques," Chowdhury et al. have proposed and implemented a method for predicting systolic and diastolic blood pressure with the help of PPG signal features and machine learning algorithms [17]. They have worked on nineteen different machine learning algorithms for both systolic (SBP) and diastolic (DBP) blood pressure prediction models; from these, they have concluded that the GPR and ensemble trees performed best. Furthermore, they have used various feature selection methods to reduce computational complexity and explained that the combination of ReliefF feature selection and the GPR machine-

learning algorithm produced the best results. The authors argue that the proposed model can be used in developing commercial light computation-based prototypes that can accurately predict the blood pressure like a system can help in continuously monitoring blood pressure and preventing any critical health conditions due to unexpected changes.

Khalid et al. have developed and compared three machine learning algorithms (regression tree, multiple linear regression (MLR), and support vector machine (SVM)) to estimate blood pressure using photoplethysmography (PPG) signals [18]. They have concluded that the regression tree algorithm is the best approach to the ISO standard for device verification, with generally acceptable measurement accuracy. Moreover, the authors suggest that the regression tree algorithm only achieves acceptable measurement accuracy in the normotensive category, suggesting that future algorithm development for blood pressure (BP) prediction should be more specific for different BP categories.

1.4.2. Related Works Using Deep Learning Algorithms

Esmaelpoor, Moradi, and Kadkhodamohammadi have proposed to predict systolic and diastolic blood pressures using PPG signals with a multistage model based on deep neural networks [19]. The model they have proposed consists of two successive stages. In the first step, two convolutional neural networks (CNN) algorithms were used to extract morphological properties from PPG signals, and then a convolutional neural network (CNN) algorithm to predict systolic and diastolic blood pressures. In the second stage, they have studied with a long short-term memory (LSTM) algorithm to capture transient dependencies. Another important feature of the proposed model was that the learning capacity can be easily developed by expanding the architecture of neural networks to meet new requests, especially in the case of more complex data. In this study, the high performance of the model by both AAMI and BHS standards has been verified.

In the study titled "Combined Deep CNN–LSTM Network-based Multitasking Learning Architecture for Noninvasive Continuous Blood Pressure Estimation using Difference in ECG-PPG Features" Jeong and Lim have proposed a non-invasive

continuous algorithm that uses the difference between ECG and PPG as a new feature that can include PTT information [20]. The proposed algorithm is a CNN-LSTM based multitasking machine learning model. The accuracy of the prediction of SBP and DBP using the proposed model is 0.017 ± 1.624 mmHg and 0.164 ± 1.297 mmHg, respectively. This result scored to Grade A by BHS and AAMI standards, which are confirmation standards for blood pressure monitors. This study demonstrated that the difference between the R-peak time of the electrocardiogram (ECG) signal and the systolic peak of the photoplethysmography (PPG) signal, the pulse transit time (PTT), is an indicator that allows non-invasive and continuous blood pressure prediction.

Lee et al. have proposed a blood pressure prediction model with a bidirectional long short-term memory network. In this study [21], the features were extracted from the electrocardiogram, photoplethysmography, and ballistocardiogram (BCG) signals. Additionally, the leave-one-subject-out (LOSO) method is added to generalize the model and was applied. The study results have shown that the performance was improved by BCG signaling and the feature-based network outperformed the raw signal-based network. Moreover, the results have indicated that the LOSO model was better than the personalized model in the multi-day test.

Yang, Zhang, Cho, Correia, and Morgan have proposed a new hybrid deep learning model to estimate BP using raw PPG, ECG signals, and some physical properties [22]. The hybrid models were tested on the gathered dataset and supply superior predictive results compared to traditional machine learning models. Due to their flexible nature, deep learning models can take various combinations of different input types. This is a very useful feature as combining more physiological data that may be related to BP is likely to improve prediction accuracy. They have stated that hybrid models with raw PPG and ECG signals have a high potential for cuffless blood pressure prediction. In this study, different numbers of CNN blocks were used, and three CNN blocks were found to supply the best predictive results. The authors stated that due to the multi-layered and nonlinear structure of deep learning models, the relationship between input and output is not transparent and the predictions are generally not traceable. This situation causes problems in the interpretability of deep learning models and puts them into limited use when causality is of great importance in research.

Finally, Panwar, Gautam, Biswas, and Acharyya have proposed a deep learning model capable of predicting physiological parameters such as diastolic blood pressure (DBP), systolic blood pressure (SBP), and heart rate (HR) using a single channel PPG signal. The proposed model [23] has been designed by leveraging the deep learning framework of the Long-Term Recursive Convolutional Network (LRCN), eliminating the cost-effective steps of feature extraction, making it less complex for deployment in constrained resources. Moreover, the authors claimed that the PP-Net is a lightweight (low complex) model that unobtrusively offers many solutions in one solution using a sensor, providing a cost-effective, safer, and useful health monitoring mode inside/outside the clinical setting. They have said that in the future, they plan to expand this study by including other physiological parameters such as respiratory rate (RR) and SpO2 and targeting mobile platforms for inference mode.

CHAPTER 2

MATERIALS AND METHODS

The methodology proposed in this study consists of two main steps: feature extraction for blood pressure estimation from ECG and PPG signals and model extraction with deep learning algorithms. Accurate determination of blood pressure is of great importance for the success of the methodology, as it is the first step in the diagnosis of hypertension disease.

Considering the long-term blood pressure, electrocardiogram, and echocardiographic monitoring of hypertension disease diagnosis in clinical applications, automatic estimation of blood pressure from ECG and PPG signals would be helpful to accelerate and enhance this process.

The first step is to extract the features to be used in blood pressure estimation from the ECG and PPG signals. These signals consist of various features to be used in different fields. Therefore, it requires deep research to select the appropriate feature for the models to be made in blood pressure estimation. In this study, a total of 43 features were extracted from the ECG, PPG, and combination of this signal using Matlab, and it is tabulated in Table 2.1. The next step is to present deep learning models by processing the appropriate features for blood pressure estimation. The final step is to develop an auto-diagnosis version of hypertension disease by applying to implement a convolutional neural network (CNN), long short-term memory (LSTM) algorithms.

The flowchart of the methods proposed in this study can be seen in Figure 3.2. ECG and PPG signals of patients can be interpreted automatically by predicting blood pressure with preprocessing, signal processing, and computational grading techniques. In this section, the subjects related to the thesis and the methods used in the study are explained in detail.

2.1 Machine Learning

Machine learning (ML) is the process of using mathematical models of data to help a computer learn without direct instruction [24]. Machine learning is considered a subset of artificial intelligence. Machine learning uses algorithms to identify patterns in data, and these patterns are used to create a data model that can make predictions. As the amount of data and experience increases, the results of machine learning become more accurate. The adaptability of machine learning can be used in scenarios where the data, requests, or tasks are constantly changing, or that it is not possible to encode a solution effectively.

The more formal definition of machine learning is *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."* provided by Tom M. Mitchell [25].

Machine learning consists of modeling and algorithms that use mathematical and statistical methods and computer science to analyze existing data and provide meaningful inferences from these data. Machine learning algorithms have two purposes, one is to classify data according to developed models, and the other is to make predictions for future results based on these models. The accuracy rates of the estimates vary depending on the machine learning algorithms used in applications, the type of data analyzed, available resources, and the quality of the data.

Machine Learning basically consists of three main stages. The first stage is to create the model. The second is to enter the data known as the training set. In order to enter data sets, it may be necessary to select and clean the data according to the model, to transform and develop the data if necessary, and to resort to various data mining methods. The third stage is to compare the model with the training data, the learning, and the creation of a new model or prediction. An example pipeline of how machine learning works can be found in Figure 2.1.

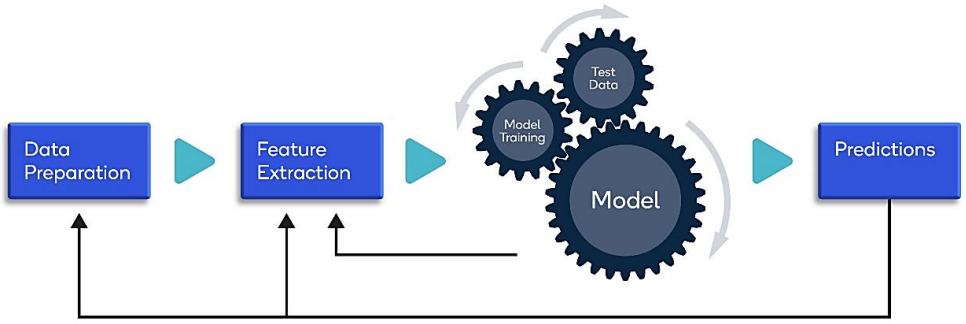


Figure 2.1 An example pipeline of how machine learning works [26]

Machine learning methods are often categorized based on supervised, unsupervised, and reinforcement.

2.1.1 Supervised Learning

Supervised learning is a machine learning approach that learns from labeled data. In the supervised learning method, training data contains label information. In other words, data with known results are used in developing a model for a solution. In this way, it is aimed to predict the results of the data without label information in the data set based on the model created. The model generates an output by using the input value in the mapping function.

$$y = f(x) \quad \textbf{(2.1)}$$

where, y is output, x is input, and $f(\,)$ is the learned mapping function.

The two most common types of supervised learning are classification, and regression. Figure 2.2 shows the supervised learning process.

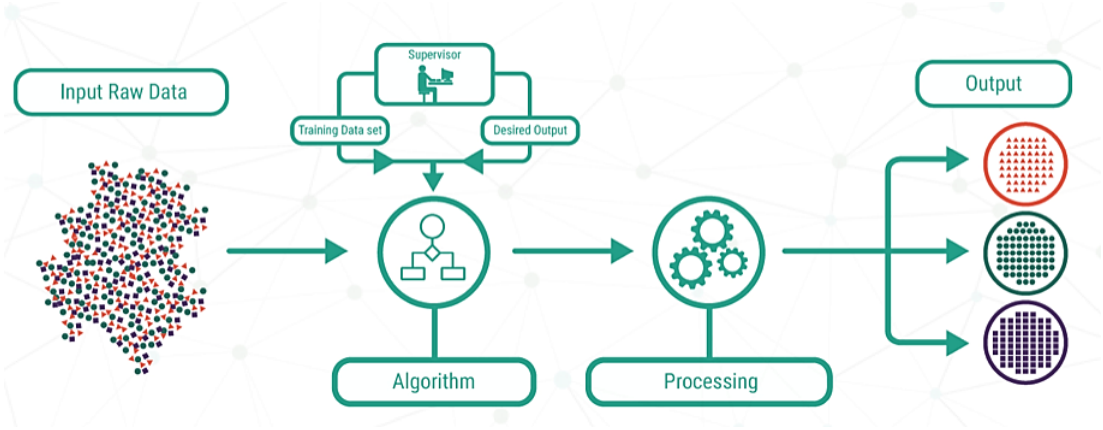


Figure 2.2 Supervised learning process [28]

Classification techniques identify a classifier such that each new data point is assigned to one of the classes. The outputs are discrete labels categorized such as ‘sick’ or ‘healthy’. The common classification algorithms are Support Vector Machine, Naïve Bayes, Nearest Neighbors (kNN), Decision Trees, Logistic Regression, and Artificial Neural Networks (ANN).

Regression techniques identify outputs as a function of the input. The outputs are real valued such as ‘weight’, ‘height. The common regression algorithms are Linear Regression, Nonlinear Regression, Gaussian Process Regression, Support Vector Machine Regression, Decision Trees, and Neural Networks.

2.1.2 Unsupervised Learning

Unsupervised Learning is a machine learning approach that uses a function to predict an unknown structure from unlabeled data. There is no training data in unsupervised learning, and it is unclear to which class the input data belongs. The algorithm has a collection of variables and tries to group data and to find out similarities between them and classify them into clusters. Figure 2.3 shows the unsupervised learning process.

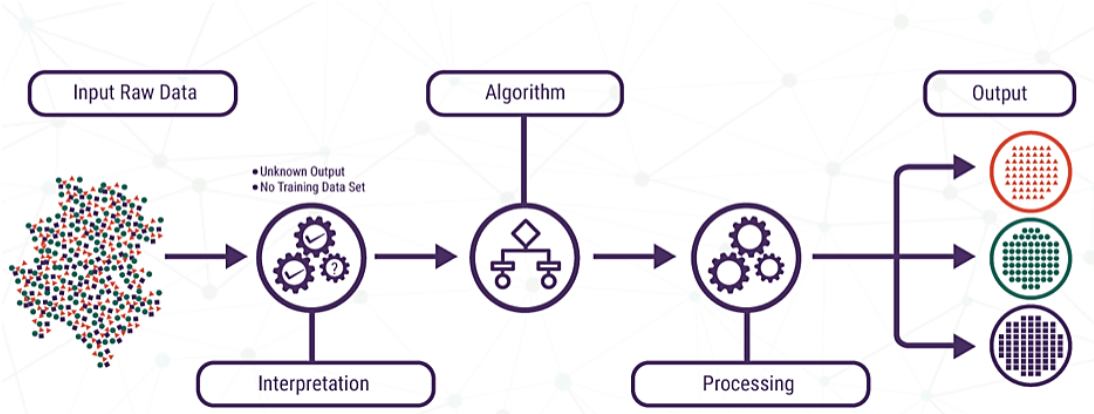


Figure 2.3 Unsupervised learning process [28]

The most common unsupervised learning methods are K-Means Clustering, Hidden Markov Model, Principal Component Analysis (PCA), and Apriori algorithm. Clustering is the grouping of data with similar characteristics in a data set. This method explore data and find hidden patterns or groupings in data.

2.1.3 Reinforcement Learning

Reinforcement learning is a machine learning approach inspired by behaviorism that deals with what actions intelligent agents must take to achieve the highest amount of cumulative reward in an environment. Reinforcement learning process can be found in Figure 2.4.

Figure 2.4 Reinforcement learning process [28]

Reinforcement learning differs from supervised learning in that it does not need to present tagged input/output pairs and does not need sub-optimal actions to be corrected. Instead, its focus is on finding a balance between exploration (of the undiscovered region) and exploitation (of available knowledge).

In the process of reinforcement learning, the subject interacts with the environment by taking an action. By observing the environment, it evaluates the new situation and the rewards it has gained, and then actions and observations follow one another in a cycle.

The most common reinforcement learning methods are Q-Learning, Temporal Difference (TD), and Deep Adversarial Networks. Example applications of the reinforcement learning algorithms are computer played board games (Chess, Go), robotic hands, and self-driving cars.

2.2 Artificial Neural Networks

Artificial neural networks (ANNs) are computer systems developed to simulate the learning path of the human brain, to generate new data from the data collected by the brain through learning, recall, generalization, and to automatically perform basic functions such as creating and discovering new information without any assistance.

Cybernetics is to examine the behavior of living things, modeling them mathematically, and producing similar artificial models. Artificial neural networks have emerged because of mathematical modeling of the learning process by taking the human brain as an example. ANNs mimic the structure of biological neural networks in the brain, their ability to learn, remember, and generalize. Artificial neural networks are mainly used in areas such as diagnosis, classification, prediction, control, data association, data filtering, and interpretation.

In Figure 2.5, the structure of a human nerve cell is as follows:

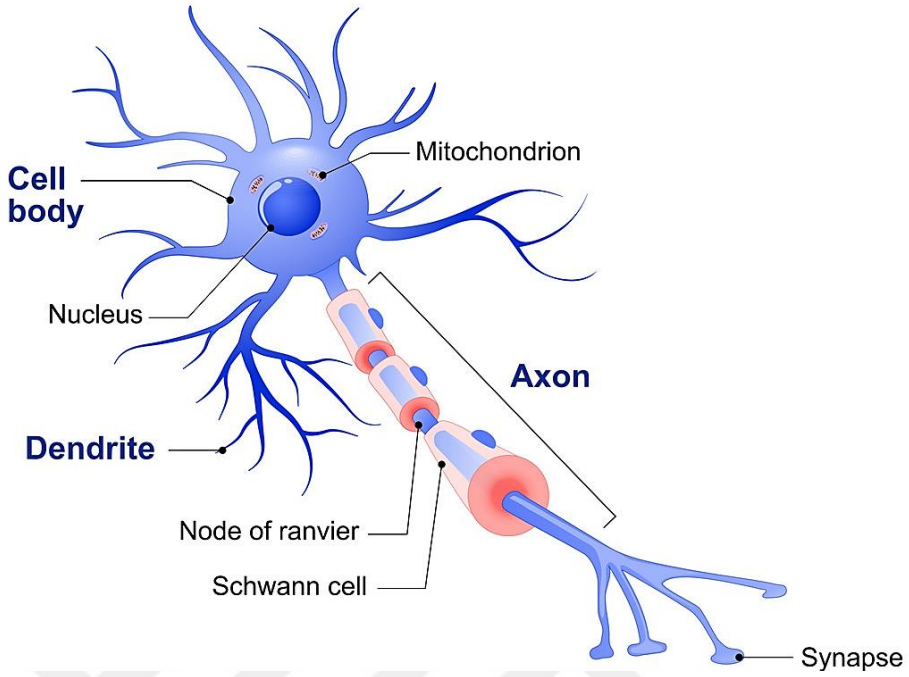


Figure 2.5 The structure of a human nerve cell [30]

Axon is the electrically active body where the output pulses are generated and the transmission on the body is unidirectional that is the system output. Dendrites are electrically passive arms that collect signals from other cells that are the system input. Synapse provides the connection of axons of cells with other dendrites. Myelin sheath is an insulation material that affects the spreading rate. The nucleus provides periodic reproduction of signs along the axon.

The nucleus is transmitted along an axon. Here, the sensor data obtained from the dendrite ends at the exit terminals are weighed in the nucleus, transmitted along the axon, and connected to another nerve cell. In this way, communication between nerves is provided.

In Figure 2.6, the mathematical model of a human nerve cell can be shown as follows:

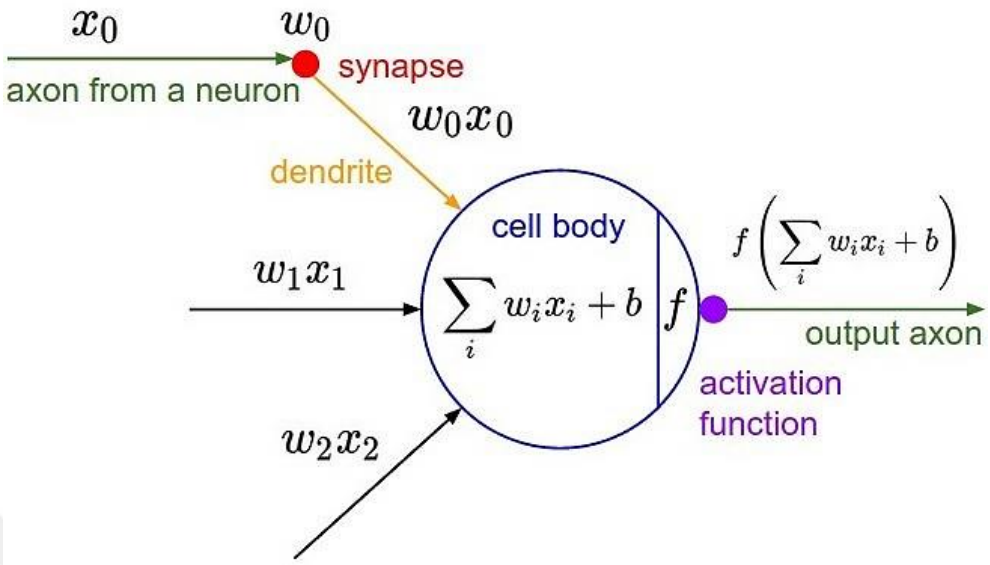


Figure 2.6 Mathematical model of a human nerve cell [31]

An artificial nerve cell consists of five parts:

1. Inputs: Inputs are data coming to neurons. Data from these inputs are sent to the neuron nucleus to be collected as in biological nerve cells.
2. Weights: The information coming to the artificial nerve cell is multiplied by the weight of the connections before reaching the nucleus through the inputs and transmitted to the nucleus. In this way, the effect of the inputs on the output to be produced can be adjusted.
3. Summation Function (Evaluation): The addition function is a function that calculates the net input of an artificial nerve cell by adding up the inputs received by multiplying them by weights.
4. Activation function: A function that takes the weighted sum of all inputs in the previous layer and then generates an output value (typically nonlinear) and passes it to the next layer. An example of commonly used activation functions illustrated in Figure 2.7.
5. Outputs: The value that comes out of the activation function is the output value of the cell. Although each cell has more than one input, it only has one output. This output can be linked to any number of cells.

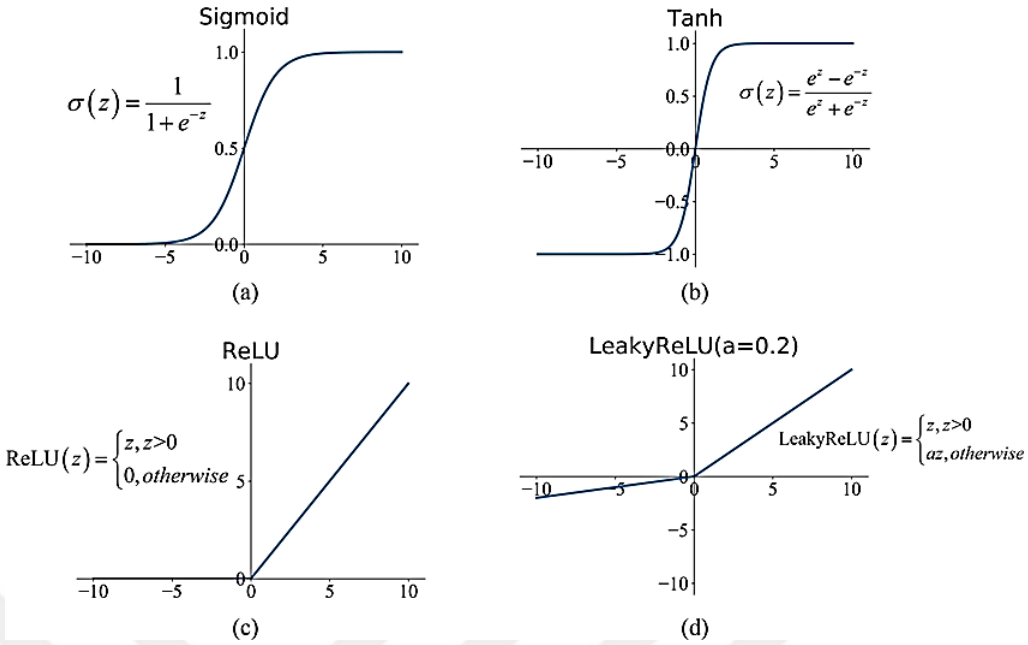


Figure 2.7 An example of commonly used activation functions: (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LReLU. [32]

Artificial neural networks consist of neurons (nerve cells) and are usually arranged in layers. Neurons have information processing properties. Neurons connect with each other to form functions. Artificial neural networks are examined in three main layers: input layer, hidden layers, and output layer. Figure 2.8 shows structure of artificial neural network.

1. Input Layer: The data that comes to the system as an entrance is included in this layer. In this layer, it transfers the input data to the hidden layer, without making any changes to it.
2. Hidden Layer: It is the layer on which data is transferred. Learning happens in this layer.
3. Output Layer: The output values that are required to be learned by the system according to the input data are located here. This is where the system output is taken.

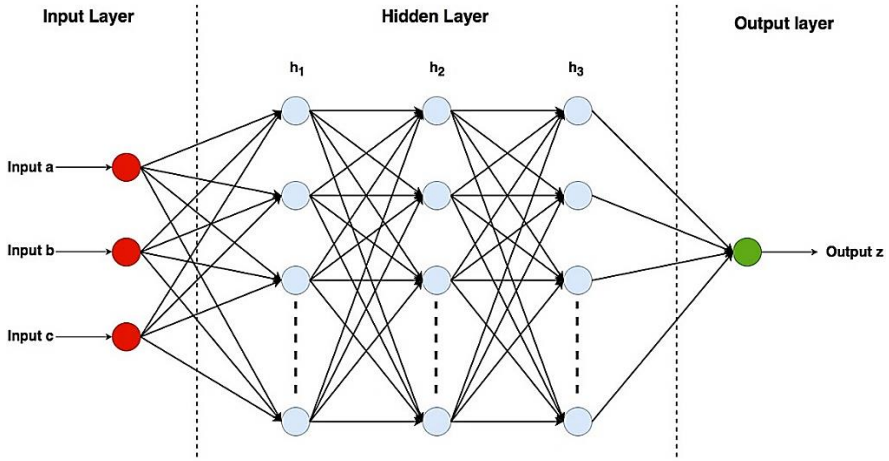


Figure 2.8 Structure of Artificial Neural Network [33]

The first computational model of a neuron was proposed by Warren McCulloch and Walter Pitts in 1943 [34]. In Figure 2.9, Perceptron was created by Rosenblatt in 1958 [35].

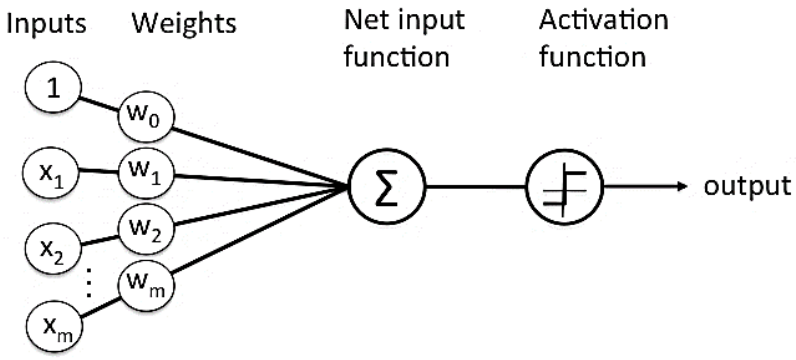


Figure 2.9 Schematic of Rosenblatt's perceptron [36]

Perceptron can be used in problems that can be divided into two parts by a linear function. Examples of these problems include AND, OR, NOT cases in Figure 2.10. However, perceptron fails to solve problems that cannot be classified linearly, such as the XOR problem in other words, it is not possible to divide the outputs into two or more classes by drawing a line or lines between them.

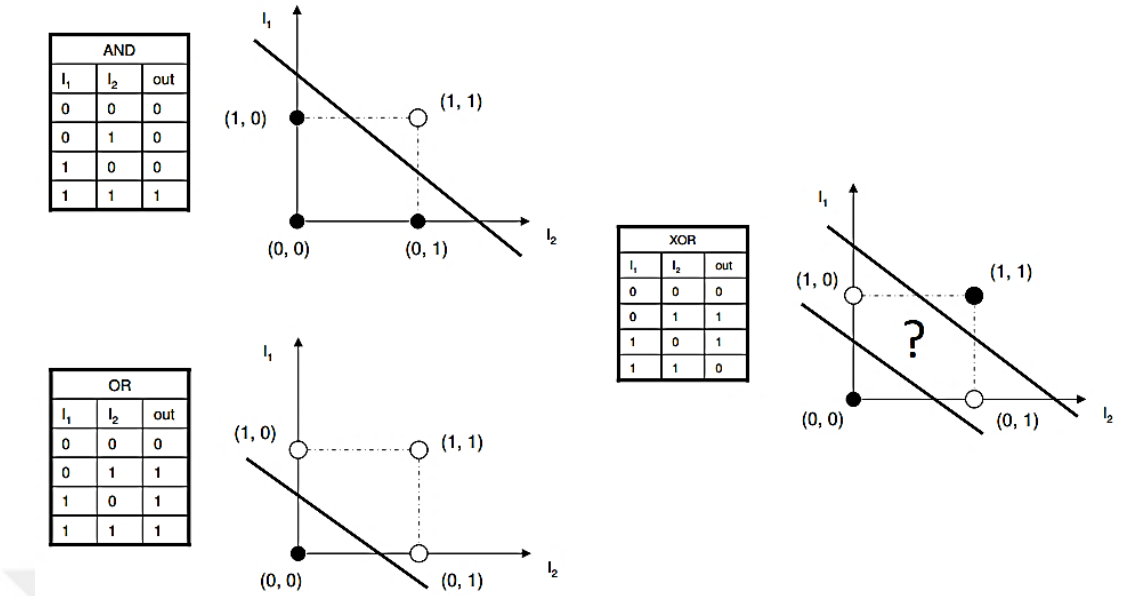


Figure 2.10 AND, OR and XOR gates on coordinate system [37]

As a result of the studies carried out to solve the XOR problem, a multilayer artificial neural network model has been developed. This error propagation model developed by Rumelhart et al. is called the backpropagation model.

2.2.1. Backpropagation

Backpropagation is a supervised learning algorithm for multi-layer neural networks training. The backpropagation algorithm looks for the minimum value of the error function in the weight domain using a technique called the delta rule or gradient descent. This algorithm is thought that weights that minimize the error function are a solution to the learning problem. Figure 2.11 shows structure of artificial neural network. The aim of the backpropagation algorithm is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs.

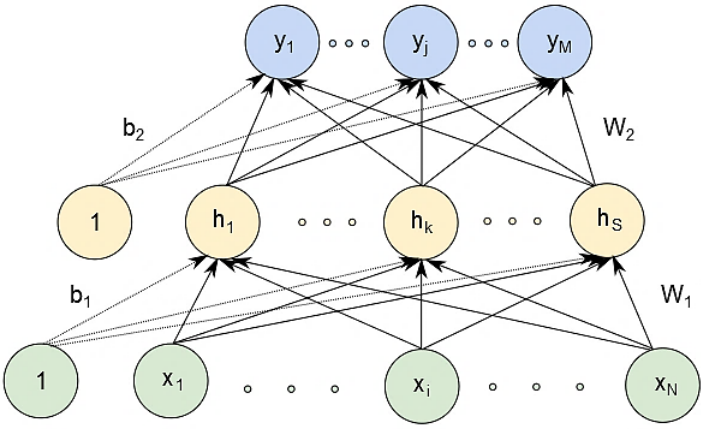


Figure 2.11 An illustration of backpropagation [38]

By calculating the total net input of each hidden layer neuron and using an activation function, the total net input is calculated, then the process is repeated with the output layer neurons. These can be expressed as follows:

$$net_{h_i} = \sum(w_j * i_k) + b \quad \textbf{(2.2)}$$

$$out_{h_i} = \frac{1}{1+e^{-net_{h_i}}} \quad \textbf{(2.3)}$$

Where w_j is each weight connected to h_i, i_k is the input of each w_j

$$\text{net}_{\text{o}_\text{i}} = \sum(w_j * out_{h_k}) + b \quad \textbf{(2.4)}$$

$$out_{o_i} = \frac{1}{1+e^{-net_{o_i}}} \quad \textbf{(2.5)}$$

Where w_j is each weight connected to o_i, h_k is the hidden neuron of each w_j

$$E_{total} = \sum \frac{1}{2}(target - output)^2 \quad \textbf{(2.6)}$$

$$E_{total} = \sum E_{o_i} \quad \textbf{(2.7)}$$

With backpropagation, the goal is to update each of the weights in the network so that the actual output is closer to the target output, thus minimizing error for each output neuron and the network as a whole.

$$\Delta w_i \leftarrow \Delta w_i + \eta(target - output) * i \quad \textbf{(2.8)}$$

where η is the learning rate.

$$\frac{\partial E_{total}}{\partial w_i} \quad \textbf{(2.9)}$$

Above this formula is read as the partial derivative of E_{total} with respect to w_i, also called the gradient with respect to w_i.

By applying the chain rule

$$\frac{\partial E_{total}}{\partial w_i} = \frac{\partial E_{total}}{\partial out_{o_i}} * \frac{\partial out_{o_i}}{\partial net_{o_i}} * \frac{\partial net_{o_i}}{\partial w_i} \quad \textbf{(2.10)}$$

Some sources extract the negative sign from δ so it would be written as,

$$\frac{\partial E_{total}}{\partial w_i} = -\delta_{o_i}\ out_{h_i} \quad \textbf{(2.11)}$$

To reduce the error, this value is subtracted from the current weight and multiplied by an optionally adjusted learning rate.

$$w^+{}_i = w_i - \eta * \frac{\partial E_{total}}{\partial w_i} \quad \textbf{(2.12)}$$

2.2.1.1. Vanishing Gradient

The vanishing gradient problem arises in machine learning when training artificial neural networks using gradient-based learning techniques and backpropagation. Each iteration of training, each of the neural network's weights receives an update proportionate to the partial derivative of the error function with respect to the current weight. The issue is that the gradient may be vanishingly small in some circumstances, thus preventing the weight from changing its value. In the worst-case scenario, the neural network will be unable to train further. Traditional activation functions, such as

the hyperbolic tangent function, have gradients in the range (0, 1), while backpropagation computes gradients using the chain rule. This has the effect of multiplying n of these small values to compute gradients of the first layers in an n-layer network, implying that the gradient (error signal) drops exponentially with n as the first layer's train slowly. Vanishing Gradience can be overcome with Relu activation function, LSTM and GRU.

2.2.1.2. Exploding Gradient

During backpropagation, an exploding gradient happens when the derivatives or slope get greater and larger with each layer. This situation is the exact opposite of vanishing gradients. This issue arises as a result of weights, not the activation function. Because of the high weight values, the derivatives will likewise be high, resulting in a large difference between the new and old weights, and the gradient will never converge. As a result, it is possible that it will oscillate about minimum and never reach the global minima point. Exploding Gradience can be overcome with truncated BTT instead of starting backpropagation at the last timestamp, clip gradience to the threshold, and RMSprop to adjust the learning rate.

2.3 Deep Learning

Deep learning is an artificial intelligence method that uses multi-layered artificial neural networks in areas such as object recognition, natural language processing, and is one of the types of machine learning [39]. Deep learning, unlike traditional machine learning methods, instead of learning with coded rules; can learn automatically from the symbols of the data of pictures, videos, sounds, and texts. Since they are flexible, they can learn from raw image or text data and their prediction accuracy can increase according to the size of the data.

Deep learning is a specialized subset of machine learning. Deep learning uses many layers of nonlinear processing units for feature extraction and conversion. Each successive layer takes the output from the previous layer as input [40]. Algorithms can be supervised (like classification) or unsupervised (like pattern analysis).

In deep learning, there is a structure based on learning more than one feature level or representation of the data. Top-level features form a hierarchical representation derived from lower-level features. This representation learns multiple levels of representation corresponding to different levels of abstraction [41]. Deep learning is basically based on learning from the representation of data. When a representation is called for an image; a vector of density values per pixel or features such as clusters of edges, custom shapes can be considered. Some of these features represent the data better. At this stage, again, as an advantage, deep learning methods use efficient algorithms for the extraction of hierarchical features that best represent the data, instead of handcrafted features [42].

While traditional machine learning algorithms have a simple structure like linear regression or decision tree, deep learning algorithms are based on an artificial neural network. However, deep learning algorithms require much less human intervention. In deep learning methods, features are extracted automatically, and the algorithm learns from its errors. Due to the complex multi-layer structure, deep learning algorithms need much more data than a traditional machine learning algorithm to work properly. Comparison of traditional machine learning and deep learning illustrated in Figure 2.12.

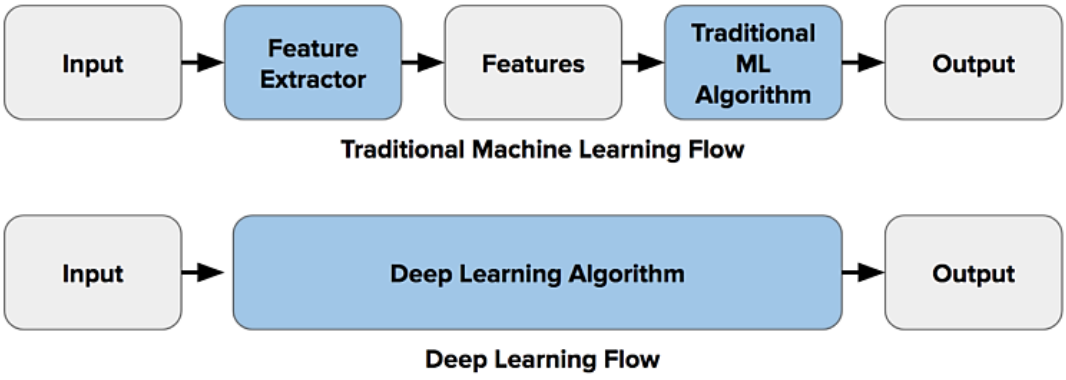


Figure 2.12 Comparison of traditional machine learning and deep learning [43]

Convolutional Neural Networks, Recurrent Neural Networks (RNN), and Deep Belief Networks (DBN) are common deep learning algorithms based on artificial neural networks. Taxonomy of deep learning methods can be found in Figure 2.13.

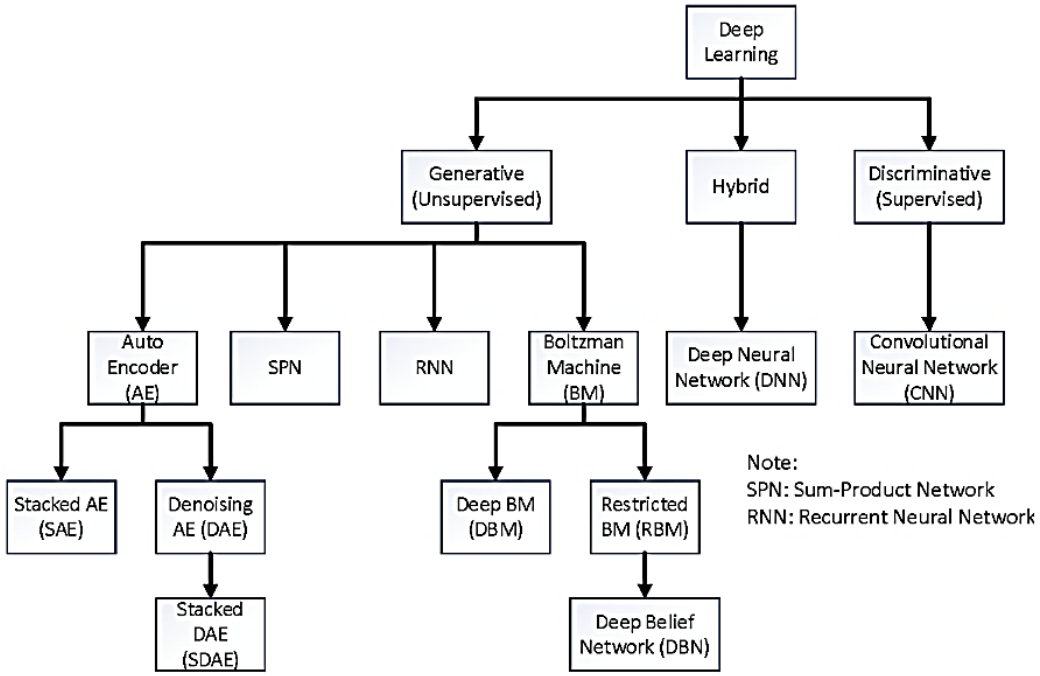


Figure 2.13 Taxonomy of deep learning methods [44]

Deep learning provides the opportunity to work with multidimensional data in multi-layered architectures using artificial neural network algorithms. Deep learning methods have been applied to a wide variety of areas such as natural language processing (NLP), image processing, visual object detection, medical diagnosis, military, and other fields.

2.3.1. Convolutional Neural Network

A convolutional neural network (ConvNet or CNN) is a deep learning algorithm that is used to analyze visual information and takes images as input. This algorithm, which captures the features in images given as input with different operations and classifies them, consists of different layers.

Basically, there are five types of layers exist in convolutional neural network architectures are convolutional layer, non-linearity layer, pooling layer, flattening layer, and fully connected layer. A convolutional neural network uses a standard neural network to solve classification problems, but prior to that it uses other layers to prepare data and detect certain features.

The architecture of a convolutional neural network is similar to the connection model of neurons in the human brain and is inspired by the structure of the visual cortex in Figure 2.14. The visual cortex is the part of the cerebral cortex of the brain that processes visual information. The visual cortex contains a complex arrangement of cells called receptor fields, which are sensitive to small sub-regions of the visual field. These cells act as local filters on the input space and are well suited to take advantage of the strong spatially local correlation found in natural images.

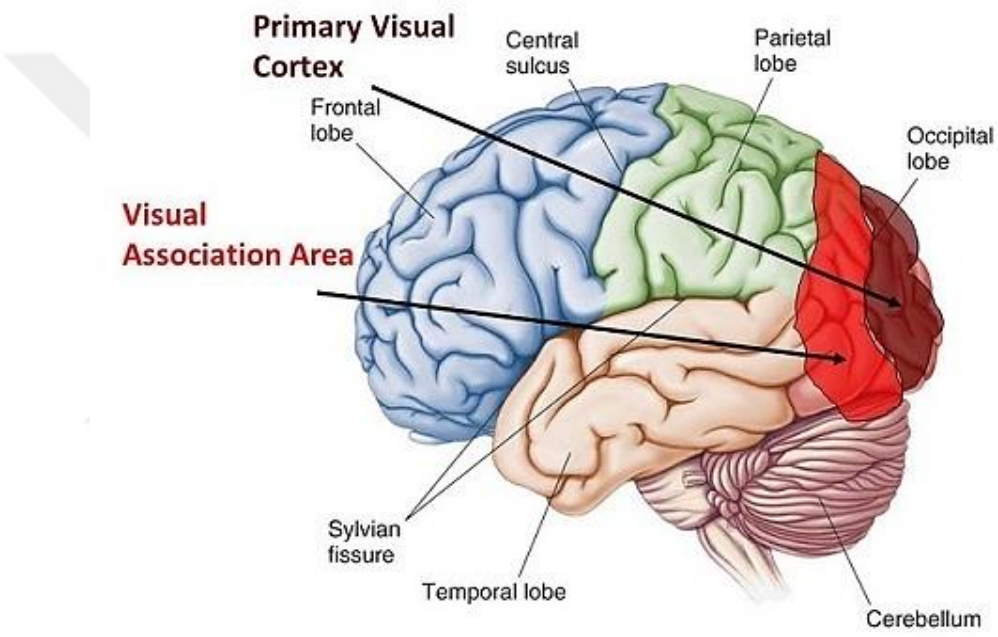


Figure 2.14 Visual Cortex [45]

CNN algorithms are applied in many different fields such as especially in the field of image and video recognition, image classification, natural language processing (NLP), and biomedical. An example CNN that is shown in the Figure 2.15.

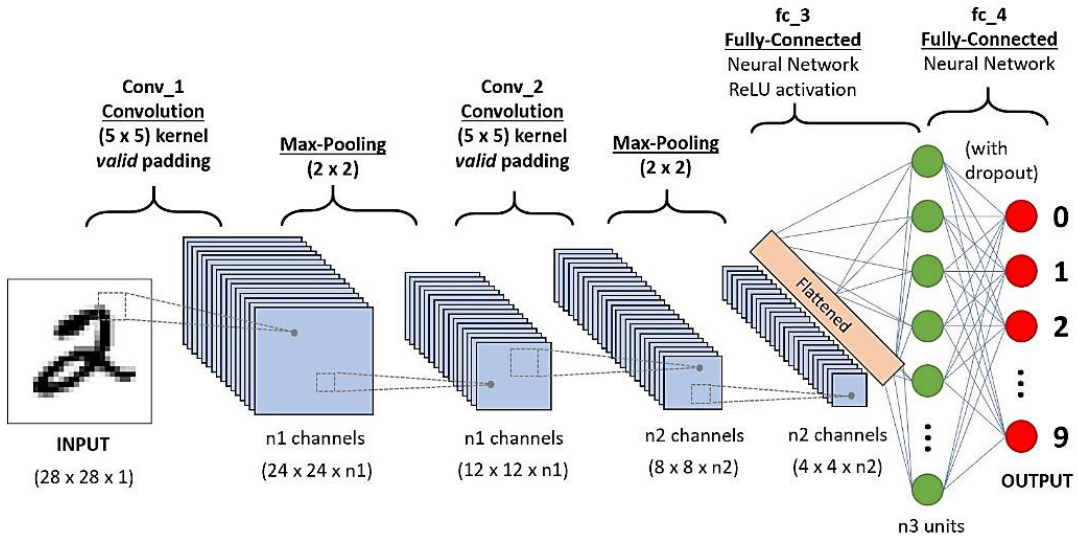


Figure 2.15 Convolutional Neural Network [46]

2.3.1.1. Convolutional Layer

The convolutional layer is the first layer that can take in an input image in CNN algorithms. The convolutional layer, which is used to determine features, consists of a series of learnable filters. In this layer, which is responsible for perceiving the properties of the input, some filters are applied to the input to remove the low and high-level features in the image. For example, these filters can detect edges. Filters are usually 3-dimensional, including width, height, and depth, and contain pixel values.

Inputs are matrices consisting of pixels with certain values. In the convolution layer, a filter smaller than the original image size hovers over the image and tries to capture certain properties from these images. Basically, the image that comes as input in this layer is passed through a filter. The values resulting from filtering allow us to create the feature map. An example pipeline of convolution operation can be found in Figure 2.16.

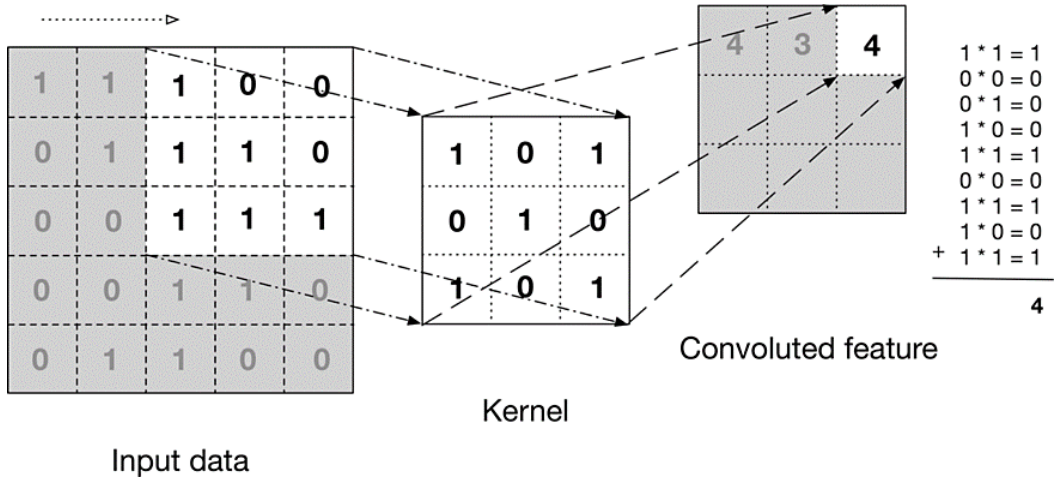


Figure 2.16 An example of convolution operation [47]

A filter in convolutional neural networks is obtained by multiplying part of the input image with the weight matrix to produce output. For example, in an image with a size of 28 x 28, to create an output with a 3 x 3 filter, the image must be multiplied by different 3 x 3 parts. The size of the specified filters is usually smaller than the size of the original image. By multiplying each 3 x 3 matrix-sized part of the image with the 3 x 3 filter shown below in Figure 2.17, the feature map is created by applying it to the whole image.

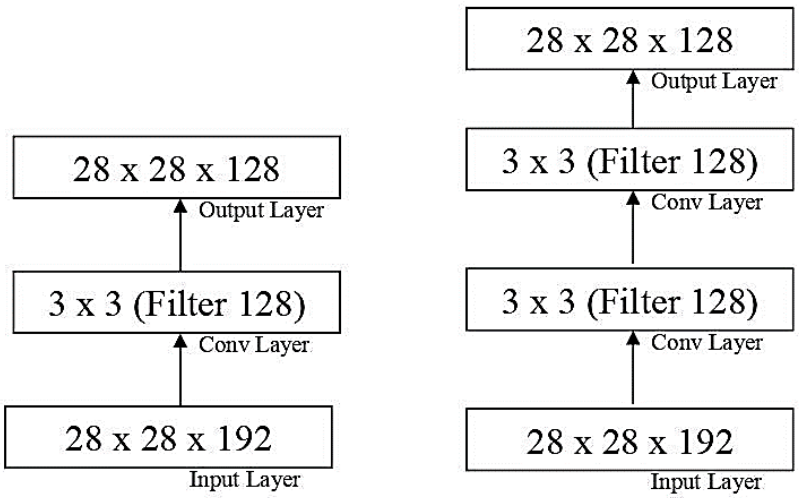


Figure 2.17 The left image displays standard 3x3 convolution, resulting in (3x3x192x128) multiplications, whereas the right image utilizes a 1x1 layer before convoluting it with a 3x3 window, resulting in ((1x1x192x64) + (3x3x64x128)) multiplications [48].

From a filtered input, a feature map can be created and only detect one feature type. It can then create a second feature map that detects another type of feature using a second filter. It can also make filters more complex when some complex features in the image need to be extracted.

The parameters learned in CNN algorithms are the values in these filters. The model constantly updates these values and begins to detect features even better. Apart from that, visuals can be sharpened with some known filters, blur can be added, and edge detection can be done. The filters used for these are as follows.

2.3.1.2. Stride

The stride value is a variable that can be changed as a parameter in CNN models, controlling how the filter evolves around the input image. The Stride value is used to determine how many pixels the filter will move over the main image. For example, in convolution operations with stride value one, the filter completes this process by simply skipping one pixel. If the stride value is 2, the number of pixels it moves increases and it also affects the size of the output of the feature map to be obtained.

2.3.1.3. Padding

In the CNN stages, while applying the filters, it is necessary to preserve as much information as possible for the other convolutional layers. The filtered image is smaller than its original size. Padding is applied to prevent this. In the padding process, zeros are added from all four sides as if there will be a frame per image. These zero-added layers can be increased according to the size of the filter. An example of padding operation is illustrated in Figure 2.18.

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	0	1	1	1	0	0
0	0	0	0	1	1	1	0
0	0	0	0	1	1	0	0
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0

Figure 2.18 An example of padding operation [49]

Convolution layer has 3 advantages:

1. Minimize the number of parameters with the weight sharing mechanism of the feature map in the image.
2. It provides the communication and correlation between the pixels in the image.
3. Allows objects in the image to change position.

2.3.1.4. Non-linearity Layer

The Non-Linearity layer usually comes after all the convolutional layers. It is aimed to introduce nonlinearity to the system. Since all layers in the CNN structure can be a linear function, the neural network behaves like a single perception, that is, the result can be calculated as a linear combination of outputs. In this case, it causes a linearity problem in the image. In order to avoid this problem, activation functions are used in this layer. In past applications, nonlinear activation functions such as sigmoid and tahn were generally used, but now this function has started to be used because the Rectifier (ReLu) function gives the best result in the speed of neural network training. An example of ReLU operation is illustrated in Figure 2.19.

ReLU (Rectified Linear Unit) is a nonlinear function that works as

$$f(x) = \max(0, x) \quad \textbf{(2.13)}$$

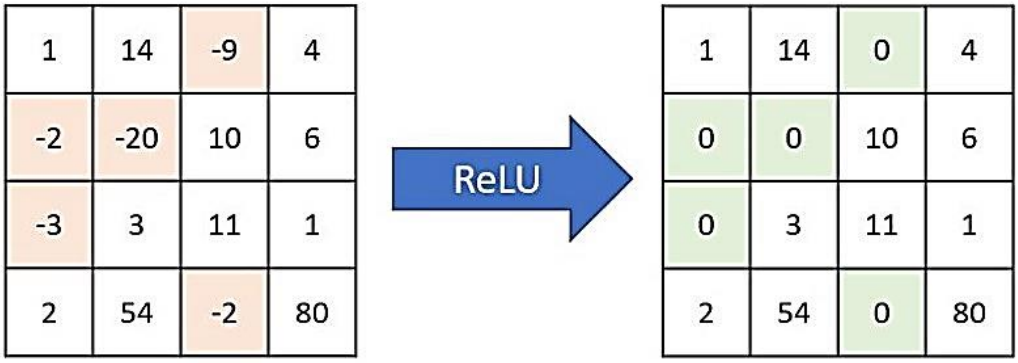


Figure 2.19 An example of ReLU operation [50]

The main purpose of the ReLU function is to get rid of negative values. For example, a ReLU function that takes the value -25 outputs 0, and a function that takes the value 25 gives 25. In order to prevent our model from learning negative values or not being able to grasp some features due to these negative values, the ReLU function is usually used. Black values in Feature Map are negative. After the Relu function is applied, the black values are removed and 0 is put in its place.

2.3.1.5. Pooling Layer

The pooling layer is a layer that is often added between successive convolutional layers in CovNet. It is used to reduce the number of weights and check their suitability. The task of this layer is to reduce the sheer size of the representation and the number of parameters and calculations within the network. In this way, both the required processing power is reduced, and the captured unnecessary features are ignored, and more important features are focused on.

There are many pooling operations, but the most popular is max pooling. There are also average pooling and L2-norm pooling algorithms, which work on the same principle as max pooling. Figure 2.20 shows an example of max pooling and average pooling.

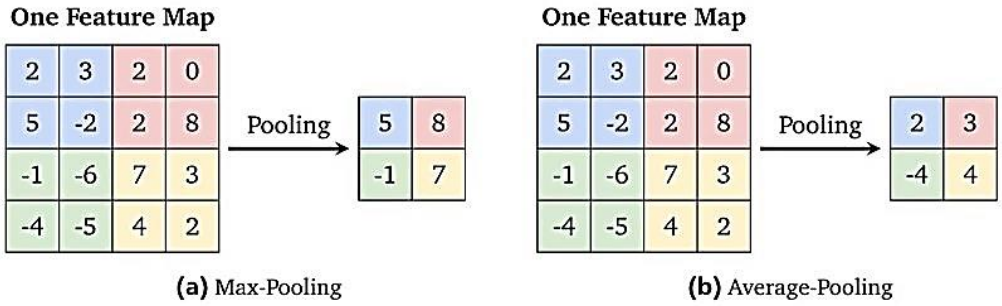


Figure 2.20 An example of Max-Pooling and Average-Pooling [51]

In the pooling layer, which has a kernel (filter) as in the convolutional layer, this kernel again hovers over the image. However, instead of a convolutional operation, he applies the determined pooling technique. That is, if max pooling is applied, the largest value in the area covered by the filter if average pooling is applied, the average of the values in the filter is taken. In this way, the size is reduced, and its important properties can be determined. In Figure 2.21 shows an example of max pooling taking a pooling stride of 2 x 2 dimension.

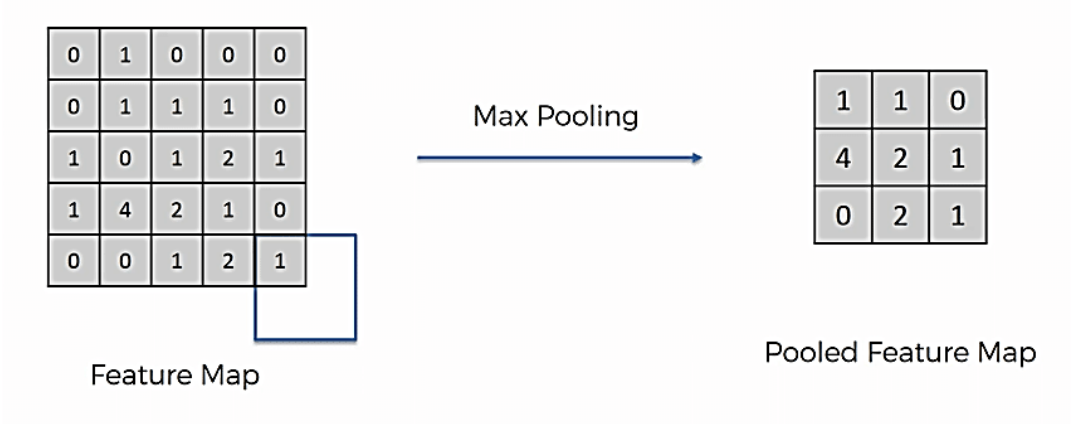


Figure 2.21 An example of Max-Pooling taking a pooling stride of 2 x 2 dimension [52]

2.3.1.6. Flattening Layer

The flatting layer is the layer that enables to prepare the data for the classical neural network. The matrix obtained by passing through the other layers of CNN must be passed through the flatting process in order to be used in the Fully Connected layer. After this step, inputs of Fully Connected Layer are prepared. This structure can also

be considered as a simplified version of the fully connected layer structure. Figure 2.22 shows an example of flatting operation.

The task of this layer is simply to prepare the data at the input of the last and most important layer, the Fully Connected Layer. Generally, neural networks receive input data from a one-dimensional array. The data in this neural network are the matrixes coming from the convolutional and pooling layers are converted into a one-dimensional array.

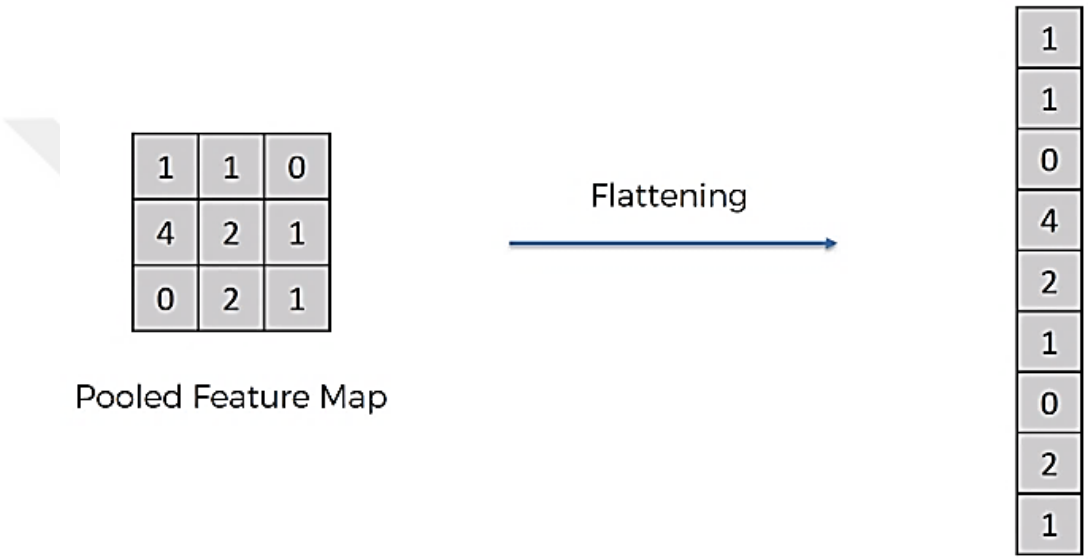


Figure 2.22 An example of flatting operation [53]

2.3.1.7. Fully Connected Layer

Fully Connected layer is the standard neural network used in classification and is the last and most important layer of ConvNet as illustrated in Figure 2.23. It takes the data from the flattening process and performs the learning process through the neural network. The matrix image that passes through the convolutional layer and the pooling layer several times in the Fully Connected layer is transformed into a flat vector.

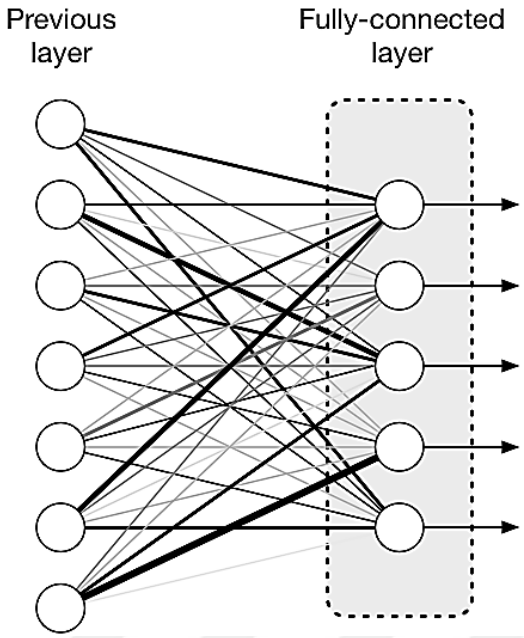


Figure 2.23 Fully connected layer [54]

2.3.1.8. Dropout

Dropout (also called Dilution) is a regularization technique that prevents complicated co-adaptations on training data, therefore decreasing overfitting in artificial neural networks. The term dropout, which is an efficient way to perform model averaging with neural networks, refers to the thinning of weights. Dropout refers to the practice of randomly "dropping out," or omitting, units (both hidden and visible) during the neural network training process. Both thinning weights and dropping out units cause the same sort of regularization, and the word dropout is frequently used to refer to weight dilution.

Dropout is often divided into two types: mild dilution and high dilution. Weak dropout refers to the process in which the finite percentage of deleted connections is small, and strong dropout refers to the process in which this fraction is high. There is no obvious distinction between strong and weak dilution, and the term is sometimes useless while having consequences for how to solve for exact answers. Dilution is sometimes used to provide dampening noise to the inputs. In that situation, weak dilution refers to the addition of a small quantity of damping noise, whereas strong dilution refers to the

addition of a larger amount of damping noise. Both may be rewritten as weight dilution versions.

These approaches are sometimes known as random pruning of weights, although this is generally a one-way process that does not occur on a regular basis. The network is trimmed and then retained if it improves on the prior model. Dropout and dilution both relate to an iterative process. Weight pruning generally does not indicate that the network continues to learn, whereas dilution/dropout implies that the network continues to learn after the approach is performed. An example pipeline of a neural network without dropout and a neural network after dropout can be found in Figure 2.24.

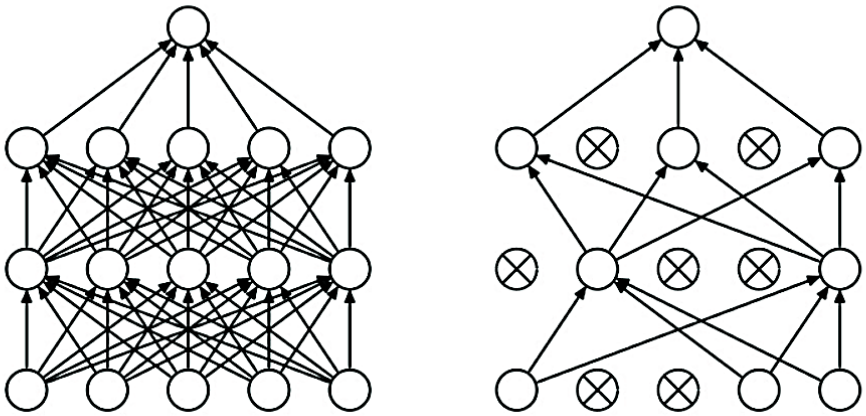


Figure 2.24 A neural network without dropout and a neural network after dropout [55]

2.3.1.9. Softmax

The softmax function, also called as the softargmax normalized exponential function, is a multidimensional extension of the logistic function. It is commonly employed as the final activation function of a neural network to normalize the output of a network to a probability distribution across anticipated output classes in multinomial logistic regression. The Softmax activation function calculates the relative probabilities is in Figure 2.25.

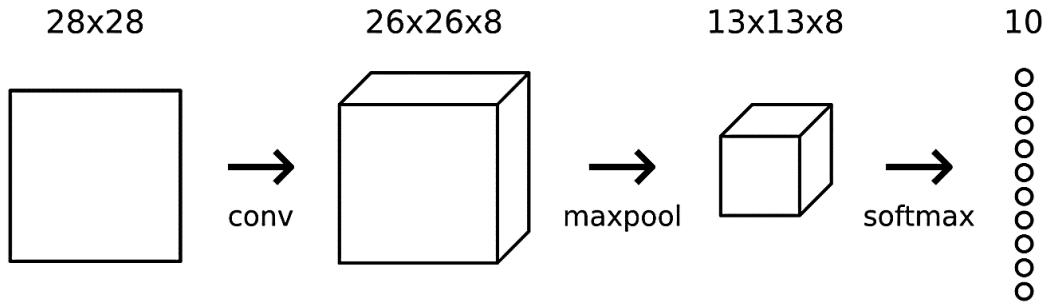


Figure 2.25 An example of softmax operation [56]

In neural network models that predict a multinomial probability distribution, the softmax function is used as the activation function in the output layer. That is, softmax is utilized as the activation function in multi-class classification situations when class membership on more than two class labels is needed.

The softmax function converts a vector of K real values to a vector of K real values that total to one. The softmax converts input values that are positive, negative, zero, or more than one into values between 0 and 1, allowing them to be understood as probabilities. Softmax converts it to a small probability if one of the inputs is small or negative, and a large probability if an input is large, but will always remain between 0 and 1.

Many multilayer neural networks end up in the penultimate layer, which yields real-valued scores that are not properly scaled and can be difficult to work with. Softmax is very useful here because it converts the scores into a normalized probability distribution that can be displayed to a user or used as input to other systems. Therefore, it is usual to add a softmax function as the last layer of the neural network.

Input of function is Z_j, where $j = 1, \dots, K$

The softmax formula is as follows:

$$\sigma(Z_j) = \frac{e^{Z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \textbf{(2.14)}$$

where all Z_j values are input vector elements and can accept any real value The normalization factor at the bottom of the calculation guarantees that all of the function's output values total to 1, resulting in a legitimate probability distribution.

2.3.1.10. Architectures of Convolutional Neural Networks

The way to set up a simple CNN architecture is to put several convolutional layers in a row and add a ReLU layer after each. And after that the Pooling layer (s) and Flattening layer must be added. Then Fully Connected layer is added as much as the ReLu layer. CNN's last layer must be the Fully Connected layer.

There are several builds that use the CNN network structure, the most common of which are:

LeNet: LeNet, one of the first successful applications of Convolutional Networks, was developed in the 1990s by Yann LeCun [57].

AlexNet: The first popular study on Convolutional Networks in computer vision was AlexNet [58], developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. The structure of the network is similar to LeNet. But compared to LeNet, it contains convolution layers that are deeper, larger, and stacked on top of each other [59].

ZF Net: It is the improved version of AlexNet. Minor adjustments were made on the structural hyper parameters of AlexNet, the dimensions of the middle convolution layers were expanded and the filter sizes in the first layer were reduced [58].

GoogLeNet: The main contribution of this network is the development of the inception module, which drastically reduces the number of parameters in the network [57]. There are several more sequel versions of GoogLeNet, and the most recent one is Inception-v4 [58].

VGGNet: VGGNet is the network developed by Karen Simonyan and Andrew Zisserman. The main contribution of VGGNet is that it demonstrates that the depth of the network is an important part of good performance. Pre-trained models are accessible for plug-and-play use. The disadvantage of VGGNet is that it is very

expensive and uses a lot of memory and parameters (140M). Most of these parameters are in the first fully linked layer [57].

ResNet: It is a network structure developed by Kaiming He et al. The feature of ResNet is that it heavily uses stack normalization and skips connections. This structure also loses the fully connected layers at the end of the network. ResNets are currently the most advanced network structure in ESA models and frequently used in studies [57].

2.3.2. Recurrent Neural Network

Recurrent neural networks (RNNs) are a class of artificial neural networks in which connections between nodes form a directed loop. In iterative neural networks, the output depends not only on the available inputs but also on the state of the neuron of the previous step. Since RNNs can form a loop, they can make sense of sequentially developing activities with each other. An example of basic recurrent neural network architecture is illustrated in Figure 2.26.

The main purpose of recurrent neural networks is to use sequential information. In classical artificial neural networks, there is no association between previous states or inputs, while in RNNs associations made with previous inputs or states. It has been widely used in recent years due to its ability to classify and interpret the activities in the flow. Example applications of RNNs can be given as speech recognition, language modeling, language translation, image captioning.

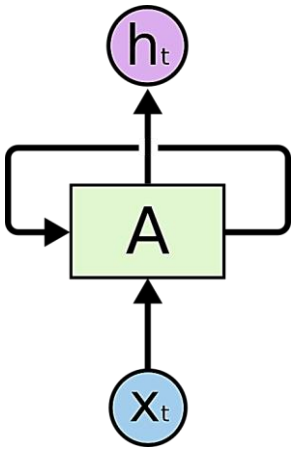


Figure 2.26 Basic recurrent neural network architecture [60]

In traditional artificial neural networks, the results from the cells do not come back to them as inputs. If the RNN opened, an architecture like the following in Figure 2.27 emerges. In this period, the same cell repeats itself more than once. Thus, meaning established between the activities.

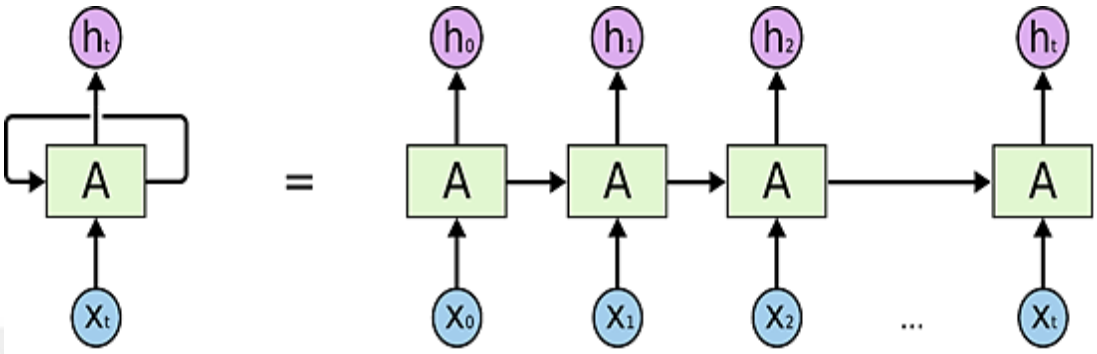


Figure 2.27 An unrolled recurrent neural network [60]

A simple recurrent neural network is shown in the Figure 2.26 above. The rectangle called 'A' is a cell in an artificial neural network. The input value of the network is X. The output value of the neural network is h. A value that comes out of the cell comes back to itself, forming a loop. Since the information of the previous time can also be used with this cycle, new information can be interpreted using old information and thus classification can be made.

RNNs give successful results in time-based problems due to their ability to connect with the past and make sense. However, in RNNs, it is not known which activities will be remembered and for how long. All information is stored in the model. While some information is important for activities, some information is unnecessary. Therefore, in the classification of some activities, it is not necessary to hide the whole history. A memorable recurrent neural network is shown in the Figure 2.28.

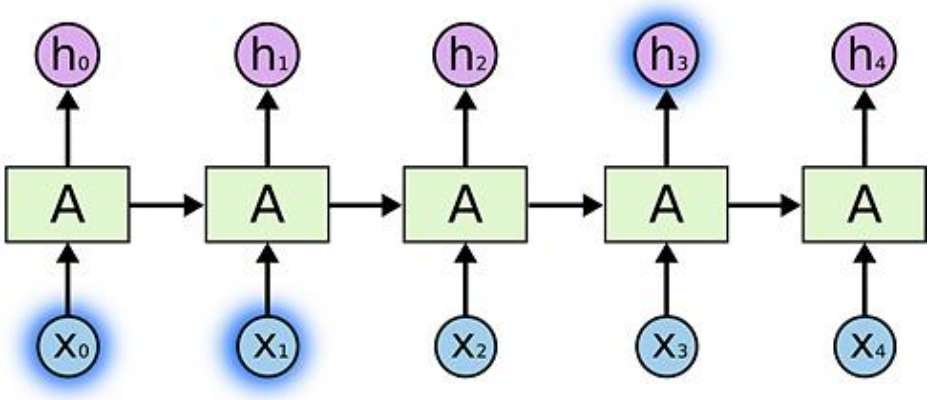


Figure 2.28 A memorable RNN [60]

In activity classification, if the necessary information has already been formed, this information may not be available. In some cases, the event in the previous frame and the event in the current frame are linked. An unremembered RNN shows in Figure 2.29.

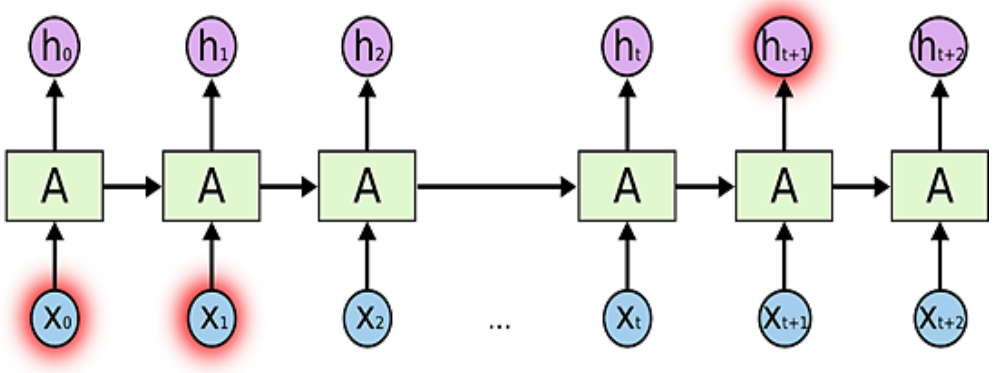


Figure 2.29 An unremembered RNN [60]

RNNs need a different architecture in order to predict events long beforehand. Long Short-Term Memory (LSTM) networks, which are a variant of better working RNNs, are used in such problems.

In theory, RNNs have the capacity to remember activities from the long past, as they repeat themselves in their internal architecture. Since such a parameter selection is not possible in practice, RNNs cannot learn from the long history.

LSTM networks do not have such a problem. Therefore, LSTMs are mostly preferred for classification of many activities. LSTMs have been used in the vast majority of recent successful studies with RNNs.

2.3.2.1. LSTM Networks

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997) and were refined and popularized by many people in following work. LSTMs work well on a large variety of problems and are widely used. Long Short-Term Memory networks are explicitly designed to avoid the long-term dependency problem. Figure 2.30 for an example of single layer repetitive in standard RNN.

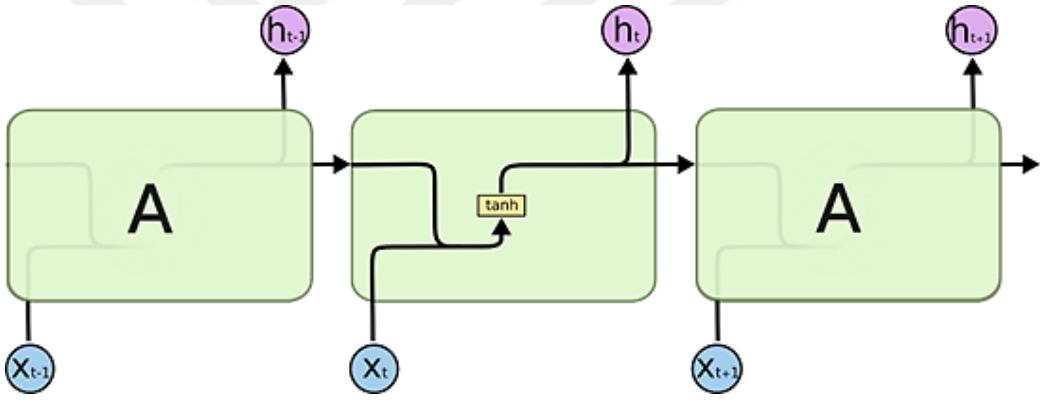


Figure 2.30 Single layer repetitive in standard RNN [60]

In traditional RNNs, there is only one tangent function (layer) in the repetitive part. The figure shows this single tangent function (layer). Instead of a single nerve layer, LSTMs have four different layers in the form of a repeating chain. The figure below shows the four different layers available in LSTM. There are many symbols in the images in which the LSTM architecture is visualized. These symbols are shown below in Figure 2.31.

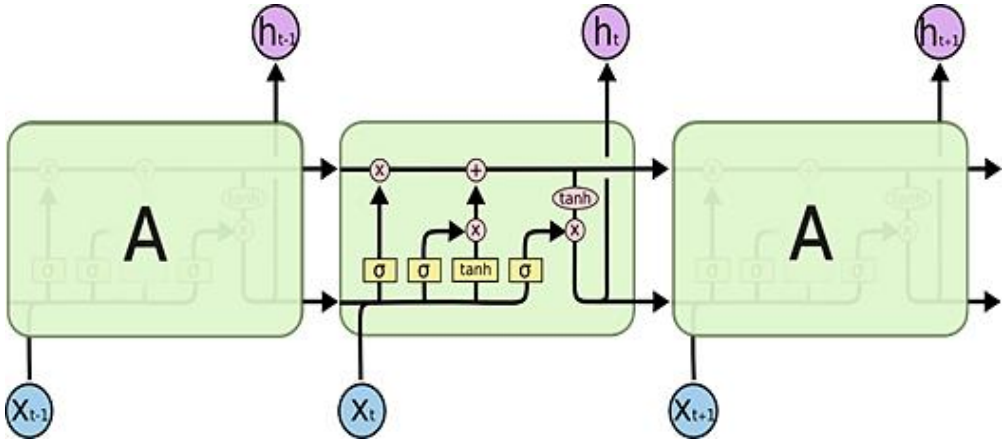


Figure 2.31 Four interactive layers in LSTM [60]

2.3.2.2. Bidirectional RNNs

Bidirectional RNNs are based on the idea that the output at t can depend not only on previous elements in the array but also on future elements. For example, bidirectional RNNs are very suitable if you want to look at both left and the right content to predict the missing word in a row. Bidirectional RNNs have only two RNNs stacked on top of each other. Then the output is calculated based on the latest state of both RNNs.

The Bidirectional RNN idea is to divide the neurons of a conventional RNN into two directions, one for positive time direction (forward states) and one for negative time direction (backward states). The outputs of the two states are not linked to the inputs of the opposite direction states. The right diagram depicts the overall structure of RNN and BRNN. In contrast to traditional RNN, which requires delays to include future information, by employing two-time directions, input information from the past and future of the present time frame may be incorporated.

BRNNs can be trained using similar algorithms to RNNs because the two-directional neurons do not have any interactions. When back-propagation across time is used, however, extra processes are required since updating input and output layers cannot be done at the same time. The following are the general training procedures: Forward and backward states are transferred first in forwarding pass, followed by output neurons. In the case of a backward pass, output neurons are passed first, followed by

forwarding states, and finally backward states. The weights are adjusted once forward and backward passes are completed.

The data before and after the current time step define the hidden state for each time step in bidirectional RNNs. Bidirectional RNNs are mostly useful for sequence encoding and estimating observations in a bidirectional environment, but they are very costly to train to owe to the lengthy gradient chains. An illustration of bidirectional RNNs can be found in Figure 2.32.

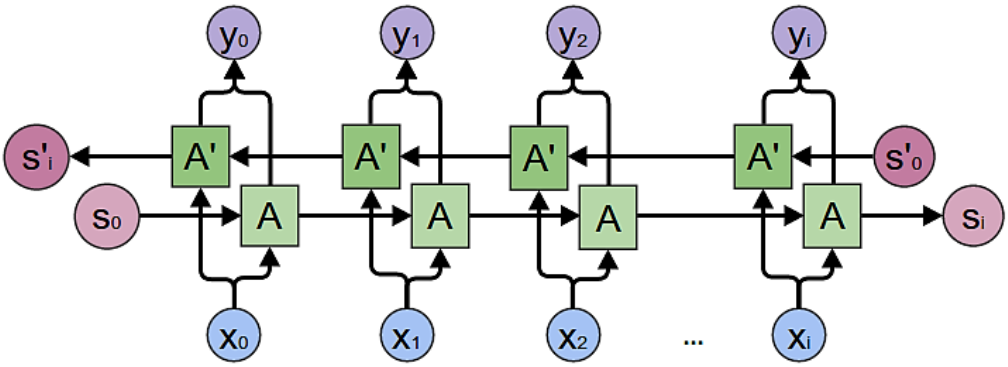


Figure 2.32 Architecture of Bidirectional RNNs [61]

2.3.2.3. Deep RNNs

Deep RNNs are similar to Bidirectional RNNs only in that we have multiple layers in the time step. In practice, it provides a higher learning capacity but also needs a lot of training data.

In deep RNNs, the concealed state information is transmitted to the current layer's next time step and the following layer's current time step. Deep RNNs come in a variety of types, including LSTMs, GRUs, and vanilla RNNs. Conveniently, all of these models are available as part of deep learning frameworks' high-level APIs. Model initialization necessitates caution. Deep RNNs, in general, need a significant amount of effort (such as learning rate and clipping) to achieve proper convergence. An illustration of deep RNNs can be found in Figure 2.33.

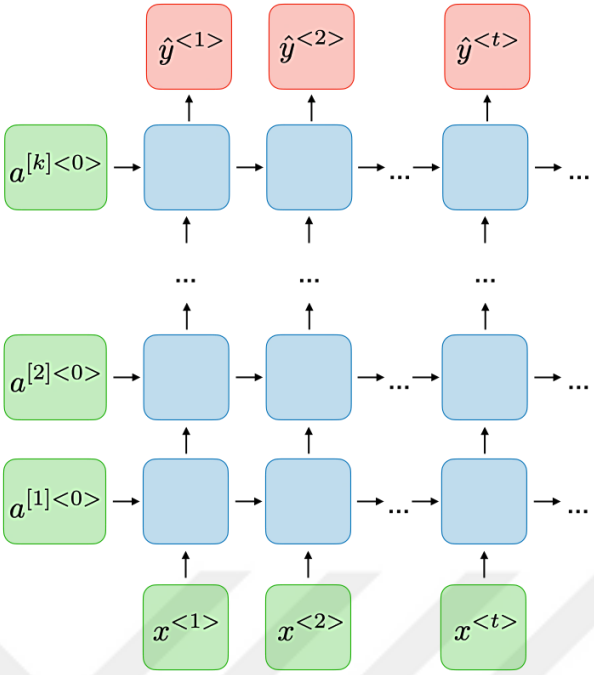


Figure 2.33 Architecture of Deep RNNs [62]

2.3.2.4. Different types of RNN's

One-to-one

One-to-one neural networks, also called Plain/Vaniall Neural networks, deal with a fixed size of the input to a fixed size of output, when independent of previous information/output, and are often used in image classification fields.

One-to-Many

One-to-Many neural networks deal with a fixed size of information as input, giving a sequence of data as output. For example, Image Caption takes the image as input and outputs a phrase.

Many-to-One

Many-to-One neural networks take a sequence of information as input and output a fixed size of the output. An example of many to one neural network is sentiment analysis, in which a particular sentence is classified as expressing positive or negative emotions.

Many-to-Many

Many-to-Many neural networks take a sequence of information as input and repeatedly process it as a sequence of data output. An example is a machine translation.

Bidirectional Many-to-Many

Bidirectional Many-to-Many neural networks provide synchronized sequence input and output. Since iterative transformation is fixed, it can be applied many times. An example is a video classification.

The Vanishing gradient problem occurs during backpropagation in deep neural networks owing to the sigmoid and tan activation functions, while the Exploding gradient problem occurs because to high weights. An illustration of Different types of RNNs can be found in Figure 2.34.

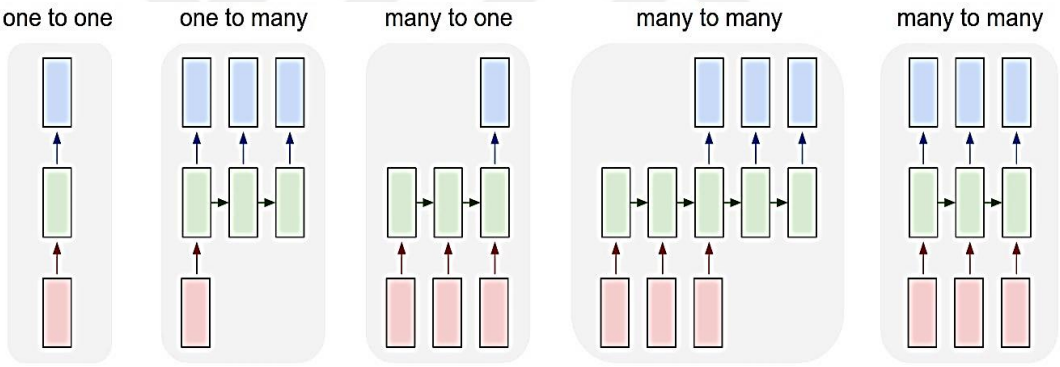


Figure 2.34 Different types of RNN's [63]

2.4 Dataset

The Research Resource for Complex Physiologic Signals, or PhysioNet, was established in 1999 under the auspices of the National Institutes of Health (NIH), as described further below. The original and ongoing missions of the PhysioNet Resource were to conduct and catalyze biomedical research and education, in part by providing free access to large collections of physiological and clinical data and related open-source software. PhysioNet hosts an annual series of challenges in collaboration with the annual Computing in Cardiology conference, focusing research on unsolved clinical and basic science problems. Members of the PhysioNet team are actively involved in cutting-edge research on the analysis of physiologic signals, both from a basic and translational standpoint. Members of the MIT Laboratory for Computational Physiology manage the PhysioNet platform. The Margret and H.A. Rey Institute for Nonlinear Dynamics at Beth Israel Deaconess Medical Center is the other core laboratory of the PhysioNet Resource.The MIMIC II database was gathered as part of a Bioengineering Research Partnership (BRP) contract from the National Institute of Biomedical Imaging and Bioengineering called "Integrating Data, Models, and Reasoning in Intensive Care".

In the MIMIC II database, there are essentially two types of data; clinical data stored in a relational database, and bedside monitor waveforms and their associated derived parameters and events stored in flat binary files (with ASCII header descriptors) and sorted with one directory per patient. For tens of thousands of Intensive Care Unit (ICU) patients, the MIMIC II (Multiparameter Intelligent Monitoring in Intensive Care) Databases contain physiologic signals and vital signs time series captured from patient monitors, as well as comprehensive clinical data obtained from hospital medical information systems. Data were collected from a variety of ICUs (medical, surgical, coronary care, and neonatal) in a single tertiary teaching hospital between 2001 and 2008. Clinical data from bedside workstations and hospital archives are included in the MIMIC II Clinical Database. The MIMIC II Waveform Database contains records of continuous high-resolution physiologic waveforms as well as numeric time series (trends) of physiologic measurements taken minute by minute.

A version of the Multi-Parameter Intensive Care Intelligent Monitoring (MIMIC) II online waveform database provided by the PhysioNet organization has used as the data set. The processing that is followed to reproduce the UCI dataset from the MIMIC II dataset has been explained in [64]. The Physionet's multiparameter Intelligent Monitoring in Intensive Care (MIMIC) II database is used as a source for the ECG and PPG signals as well as the arterial blood pressure (ABP) signal. The MIMIC II Waveform Database is illustrated in Figure 2.35. The database is formed ECG acquired from bipolar member lead form II channel, PPG using fingertip photoplethysmography and ABP (Arterial Blood Pressure) signals from 12,000 instances sampled at 125 Hz. Samples of signal in record is shown in Table 1.1.

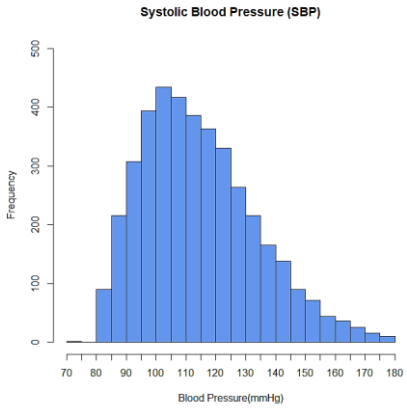
Figure 2.35 The MIMIC II Waveform Database

Certain subjects who were completely unregistered with very high or very low BP had 12,000 data after data was removed (SBP>=180, DBP>=130, SBP<=80, DBP<=60) and, in the pre-processing, each attributed to a unique identity.

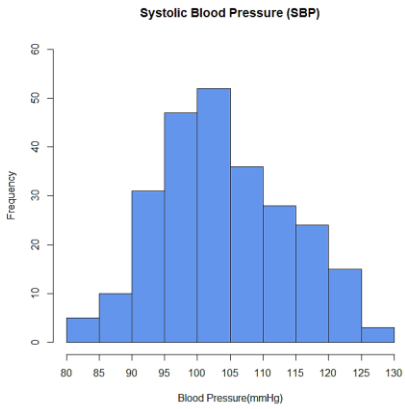
The database is in the form of MAT files consisting of the cell of array matrices where each cell displays the individual record. Each sample is also given a unique ID to avoid collision of data in the dataset. The data set is then divided by 80% for training and 20% for testing. Figure 2.36 shows distribution of train and test dataset using reference blood pressure. The signals are then processed before they are used to build a deep learning model.

Table 2.1 Samples of signal in record

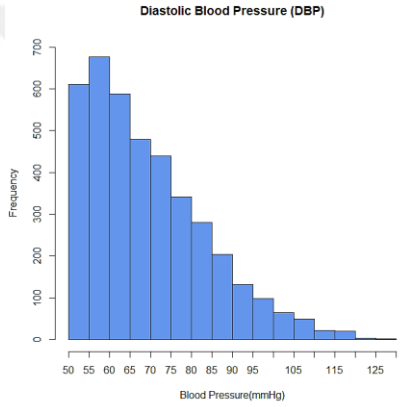
Time (sec)	**II (mV)**	**III (mV)**	**AVL (mV)**	**AVF (mV)**	**ABP (mmHg)**	**PAP (mmHg)**
0.000	0.176	-	-	1.000	109.600	-
0.008	0.112	-	-	1.011	106.400	-
0.016	0.056	-	-	1.021	103.200	-
0.024	0.008	-	-	1.032	100.000	-
0.032	-0.024	-	-	1.043	96.800	-
0.040	-0.048	-	-	1.053	92.800	-
0.048	-0.064	-	-	1.064	89.600	-
0.056	-0.072	-	-	1.074	86.400	-
0.064	-0.080	-	-	1.085	83.200	-
0.072	-0.080	-	-	1.096	80.000	-



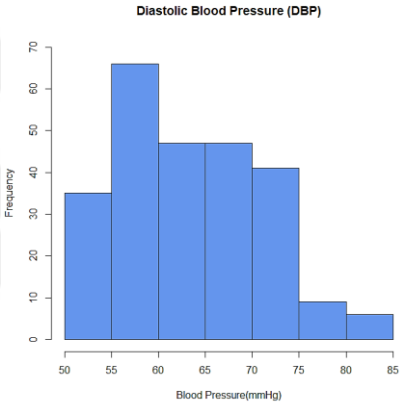
(a)



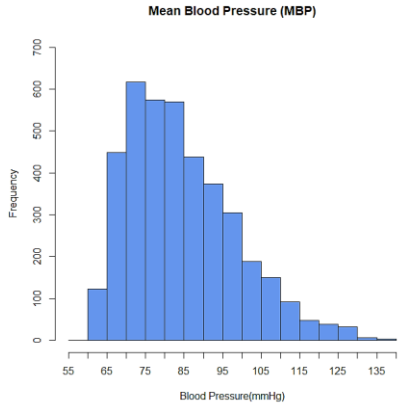
(b)



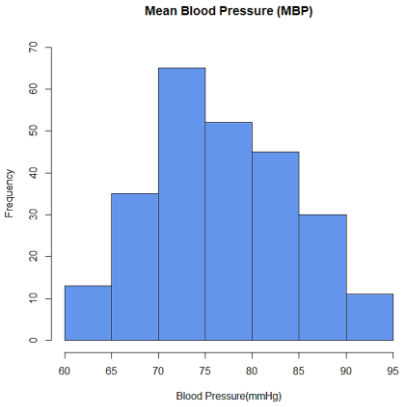
(c)



(d)



(e)



(f)

Figure 2.36 Distribution of train and test dataset (a) SBP train values (b) SBP test values (c) DBP train values (d) DBP test values (e) MBP train values (f) MBP test values

2.5 Electrocardiography

The electrocardiography (ECG), which is used for the diagnosis of cardiovascular diseases, is a device that provides a graphical recording of the electrical activity that occurs during the contraction of the heart. A weak electrical activity occurs during the contraction and relaxation of the atria and ventricles, also known as the heartbeat. This activity contains a lot of data about heart rhythm, frequency, and spread. Electrocardiography, or in other words, an ECG device is used to measure electrical activity. Electrodes affixed on the skin to the chest, arm, and leg area allow the electrical activity of the heart to be measured. An illustration of ECG of a heart in normal sinus rhythm is in Figure 2.37.

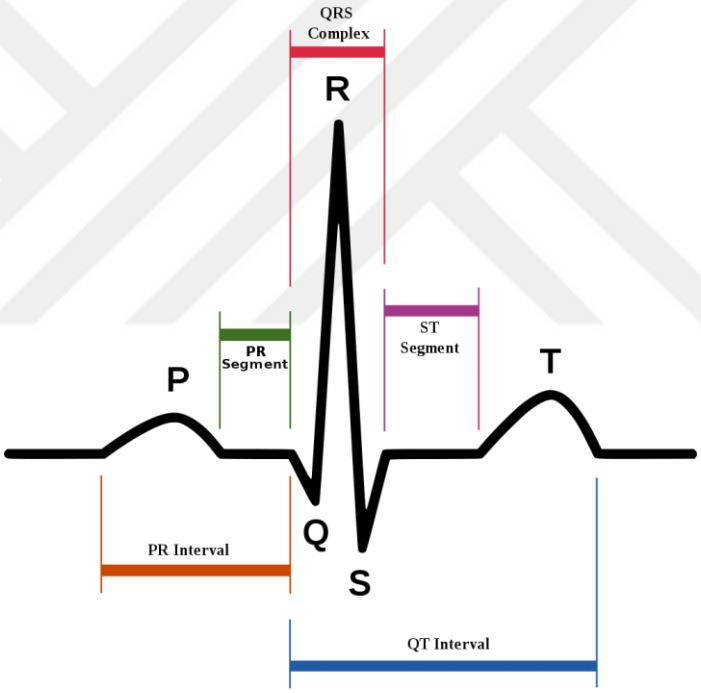


Figure 2.37 ECG of a heart in normal sinus rhythm [65]

ECG is taken with the help of electrodes attached to the skin. Thanks to the measurement of the electrical activity created by the heart every time it contracts, detailed information about the heart is obtained. Electrodes are attached to the patient's body to measure the contraction and relaxation movement created by the atria and ventricles in the heart, the stimulation of the heart by cell groups, and the electrical activity that occurs during the transmission of this stimulus.

These electrodes are connected to the ECG device with cables. The electrical activity data is converted into graphs by the ECG device. The ECG device records heartbeats consisting of P, Q, R, S, T, and U waves on the electrocardiogram. The graphical values created by these waves provide information to the physician about the patient's heart health. ECG, which provides rapid measurement of many data about heart health, is a diagnostic method frequently used by cardiologists today.

2.6 Photoplethysmography

The photoplethysmography (PPG) is a non-invasive method of obtaining physiological parameters on the body. PPG signals obtained using this method have many advantages, as they contain many physiological information and can be continuously monitored in a non-invasive manner. The basis for obtaining the PPG signal is the volumetric changes in the blood vessels that occur as a result of the heart pumping blood to the blood vessels. An illustration of a typical PPG waveform with notch, systolic peak, and diastolic peak is in Figure 2.38.

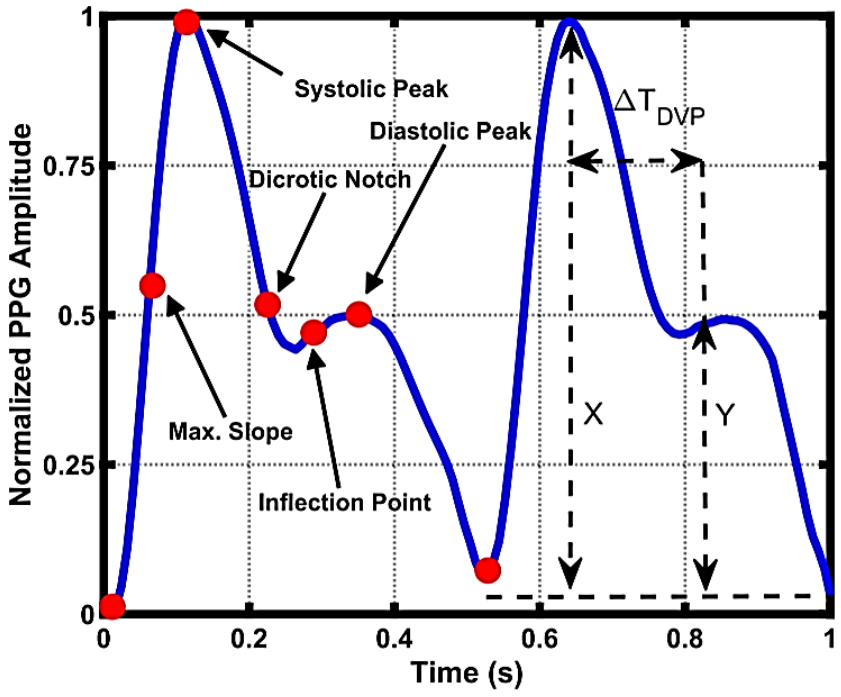


Figure 2.38 A typical PPG waveform with notch, systolic peak, and diastolic peak [66]

PPG measurement can be performed easily on the skin, such as fingertips and earlobes, where capillaries are concentrated, as well as on the wrist. During this measurement, a LED emitting light of a certain wavelength is used as a transmitter in the sensor part and a photodiode sensitive to this wavelength, which is also used as a receiver.

The amplitudes and waveforms of PPG signals contain information about many physiological parameters of the person, and different health data are reached by changing the wavelengths from which PPG signals are obtained.

2.7 Systolic Blood Pressure

Systolic Blood Pressure (SBP) is the blood pressure that occurs in the artery wall with the contraction of the heart. It is the blood pressure that occurs from the moment the heart pumps blood from the left ventricle to the aorta until the aortic valve is closed. In systolic blood pressure, the pressure in the aorta reaches its highest level.

2.8 Diastolic Blood Pressure

Diastolic Blood Pressure (DBP) is the blood pressure that occurs in the artery wall as a result of the relaxation of the heart. When the heart relaxes, blood pressure in the aorta drops to its lowest level.

2.9 Mean Blood Pressure

Mean Blood Pressure (MAP) is the average blood pressure in the artery. In Equation 2.14, the equation where the average blood pressure is calculated is given.

$$MAP = \frac{(2*DP)+SP}{3} \quad \textbf{(2.15)}$$

2.10 Feature Extraction

In this study, a total of 43 features were extracted from the ECG, PPG, and combination of this signal using Matlab, and it is tabulated in Table 2.2. The feature set consists of ECG-based features such as Womersley number, SDI, QRS, and QT_c interval, and PPG-based features such as photoplethysmograph intensity ratio (PIR). The feature set is used to minimize the error in blood pressure estimation and to explore the relationship of these features with blood pressure. The extracted features were trained with convolutional neural networks, long short-term memory algorithms, and combining the two algorithms.

2.10.1. Pulse Transit Time (PTT)

The pulse transit time (PTT) difference between the peak of the R-wave in electrocardiogram (ECG) and the fingertip photoplethysmogram (PPG) signals, can be used to predict the systolic blood pressure (SBP) and diastolic blood pressure (DBP).

2.10.2. Photoplethysmograph Intensity Ratio (PIR)

PIR is the ratio of PPG peak intensity IH to PPG valley intensity IL of the pulse signal.

2.10.3. Womersley number

The Womersley number also called the alpha parameter is a dimensionless number that has been used the study of biofluid mechanics and biofluid dynamics. It was first derived by John R. Womersley (1907-1958) in his work with blood flow in arteries. This parameter is the ratio of inertial force to viscous forces. The Womersley number is critical in keeping dynamic similarity when measuring an experiment. When the alpha parameter is decreased, the viscous effect tends to dominate and, in the period, impress the blood pressure in case of pathological status already in larger arteries.

PIR represents the change in artery diameter, which is one of the major causes of systemic vascular resistance. It is a PPG peak-to-valley intensity ratio. As the artery diameter (Ds) grows during systole, the incident light intensity is absorbed by the blood in the arteries, causing the transmitted light intensity to fall to a minimal value (IL). During diastole, the arterial diameter (Ds) decreases, thus incoming light intensity is absorbed by the least amount of blood in the arteries, causing transmitted light intensity to the peak. To calculate the Womersley number, R is assumed to represent the IL of the PPG signal, as it denotes the maximum degree to which the artery may widen to contain the blood during systole.

Womersley parameter extraction, R is the diameter of the artery during systole, where ρ is the constant density of blood, ω is the frequency of a heart rate (HR), and μ is the viscosity of blood from the pulse signal are as follows. The equation for a is written as,

$$a = R\sqrt{\frac{\rho\omega}{\mu}} \quad \textbf{(2.16)}$$

2.10.4. Highest Intensity (IH)

Highest intensity of PPG signal.

2.10.5. Lowest Intensity (IL)

Lowest intensity of PPG signal.

2.10.6. AC Component Max Amplitude (Meu)

Meu represents the viscosity of the blood calculated based on pulse pressure wave propagation.

2.10.7. Extraction of ECG Based Features

The ECG signal is used to extract time-domain characteristics such as the QRS complex, PR interval, RR interval, QR interval, QT interval, QT interval corrected for heart rate, TQ interval, Systolic diastolic interval (SDI), SDI corrected for heart rate, the amplitude of P, Q, R, S, T, and heart rate. In Figure 2.39 shows ECG signal features.

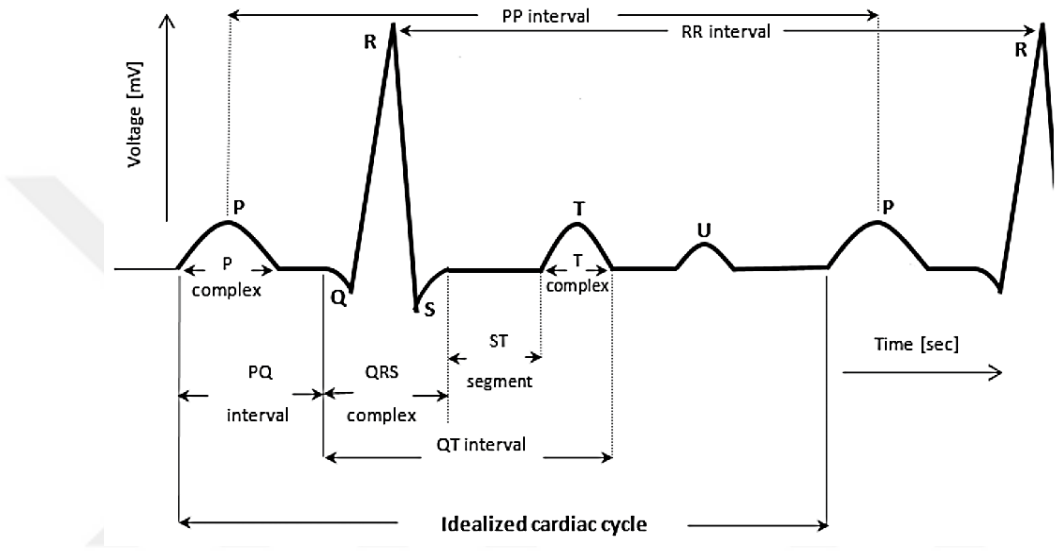


Figure 2.39 Extraction of ECG signal features [67]

2.10.7.1. QRS Complex

The QRS complex starts from the Q wave and ends at the base of the S wave. QRS complexes are usually the ECG component with the greatest amplitude and highest frequency. The QRS complex is denoted by Q, R, S if the amplitude of the waveform is large and q, r, s if it is small. Indicates the duration of ventricular activation which is measured in seconds from the beginning to the end of the QRS complex. Its normal value varies between 60 milliseconds and 100 milliseconds. The QRS complex, which is a combination of the Q, R, and S waves, represents ventricular depolarization.

2.10.7.2. PR interval

The PR interval indicates the conduction time from the onset of atrial depolarization to the onset of ventricular depolarization. The longest PR is measured in the interval from the beginning of the P wave to the first deflection of the QRS complex. The PR interval is the time from the beginning of the P wave to the beginning of the QRS complex. The normal PR interval is between 0.12-0.20 seconds.

2.10.7.3. RR interval

RR interval represents the time interval between two consecutive R waves of the ECG signals.

2.10.7.4. QR interval

QR interval represents the time interval from the start of the Q wave to the start of the QRS complex.

2.10.7.5. QT interval

The QT interval is the time from the beginning of the Q wave to the end of the T wave. Indicates the time taken for ventricular depolarization and repolarization. QT interval and heart rate are inversely proportional. The corrected QT interval is denoted by QT_c. Usually lead II, V5 or V6 gives the most accurate result.

2.10.7.6. QT_c interval

QT_c interval calculates the QT interval at a heart rate that allows the comparison of QT values at different heart rates and the detection of patients at increased risk of arrhythmia. There are many formulas used to calculate QT_c.

Bazzet's formula, which is presented as, is used to adjust the QT interval for heart rate.

$$QT_c = \sqrt{\frac{QT}{RR}} \quad \textbf{(2.17)}$$

2.10.7.7. TQ interval

TQ interval represents the time interval between T and Q of successive ECG signal.

2.10.7.8. Systolic and diastolic interval (SDI)

Systolic and diastolic interval is calculated using one of two techniques suggested in the literature [68]. SDI is defined by Fossa et al. as a ratio of QT and TQ intervals inside each cardiac cycle. Simultaneously, Imam et al. describe the ratio of QT and RR intervals, which is referred to as SDI_n in this study. Furthermore, SDI_c, which is defined as the corrected SDI for heart rate.

2.10.7.9. P Wave

P wave displays the electrical forces created by atrial activation and measures the interval from the beginning to the end of the P wave, also represents the atrial depolarization. The first part of the P wave is depolarization of the right atrium, and the second part is depolarization of the left atrium. When the valves between the atria and ventricles open, 70% of the blood in the atria falls through with the help of gravity, but primarily owing to suction generated by the ventricles expanding. Its width is < 0.11 sec, amplitude < 2.5 mm.

2.10.7.10. Q Wave

The Q wave is the negative deflection that precedes the R wave. The Q wave shows left to right depolarization of the interventricular septum. Q waves are typically seen in the left-sided leads (I, aVL, V5 and V6). While the electrical stimulus is passing through bundle and before it splits into the two bundle branches, it begins to depolarize the septum from left to right. Because there is only a little amount of conduction (thus the Q wave is smaller than two small squares), and it goes in the opposite direction to the major conduction (right to left), the Q wave points in the opposite direction to the huge QRS complex.

2.10.7.11. R Wave

The R wave is the first upward deflection after the P wave. The R wave represents early ventricular depolarization. The largest wave in the QRS complex is the R wave.

2.10.7.12. S Wave

The S wave represents depolarization in the Purkinje fibers. Because the Purkinje fibers extend throughout the ventricles from top to bottom and then back up through the ventricle walls, the S wave travels in the opposite direction as the big R wave.

2.10.7.13. T Wave

The T wave is the positive deflection after each QRS complex. T wave represents the electrical forces generated by ventricular repolarization.

2.10.8. Extraction of PPG Based Features

The PPG signal is defined by its amplitude and the length of particular components of the cardiac cycle. A number of parameters might be utilized to describe the PPG pulsatile component. Based on [11], the 20 features of the PPG signal are extracted, such as systolic upstroke time (SUT) and diastolic upstroke time (DUT), as well as the ratio of SUT and DUT. However, only ten features have been finalized since the remaining features, which are closer to zero, were ignored. The extracted features are based on [5]. In Figure 2.40 shows PPG signal features.

2.10.8.1. Systolic Upstroke Time (SUT)

SUT is the period between the start of the pulse and the peak value of the pulse is referred to as time that is computed by determining the time interval between the maximum peak value location and the minimum peak value location.

ST (10, 25, 33, 50, 66, and 75): This is the time interval in the systolic range where the amplitude of PPG is 10%, 25%, 33%, 50%, 67%, or 75% of the peak value.

2.10.8.2. Diastolic Time (DT)

Diastole is the relaxation of the cardiac muscle. When the heart relaxes, the chambers of the heart fill with blood, and blood pressure decreases. DT represents the time interval between the peak and end of a pulse. The time interval between adjacent maximum and minimum peaks.

DT (10, 25, 33, 50, 66, and 75): This is the time interval in the diastolic range where the amplitude of PPG is 10%, 25%, 33%, 50%, 67%, or 75% of the peak value.

The calculation of ST and DT, respectively, is done as follows:

First, the signal's peak value is computed. Then, all PPG values between the minimum and maximum peak (systolic range or diastolic range) are transferred to a temporary array. From the peak, 10% of the peak is determined, and the temporary array holding PPG values is checked for the existence of the 10% value or a value that is near to but slightly lower than the necessary value.

The following 20 parameters, including the times of systolic, diastolic parts, and the ratio between them, were extracted according to Figure 2.40 and used to train the deep learning models.

- SUT, DT
- At 10%: DT10, ST10+DT10, DT10/ST10
- At 25%: DT25, ST25+DT25, DT25/ST25
- At 33%: DT33, ST33+DT33, DT33/ST33
- At 50%: DT50, ST50+DT50, DT50/ST50
- At 66%: DT66, ST66+DT66, DT66/ST66
- At 75%: DT75, ST75+DT75, DT75/ST75

ST or DT 66, 75 features may have many zero values because the time interval between peak and time of occurrence of the 75 percent peak value may be so short that it becomes zero (NaN) value. Therefore, these features are neglected.

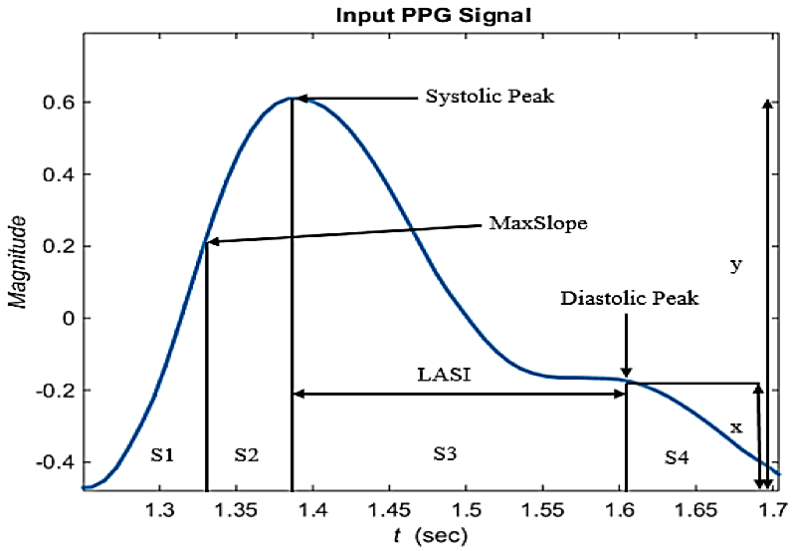


Figure 2.40 Extraction of PPG signal features [69]

2.10.9. Extraction of Physiological Parameters

The majority of these characteristics are derived from the form of PPG signals and are inspired by various cardiovascular factors. Others are measures of the time shift between a few locations on the ECG and PPG signals, which are connected to PAT parameter measurements. The following is a comprehensive list of these features.

2.10.9.1. PAT Features

PAT values are calculated by estimating the time interval between the ECG R-peak points and three places on the PPG signal. Figure 2.40 shows the PPG maximum peak (PAT_p), the PPG minimum (PAT_f), and the point at which the maximum slope of the PPG waveform occurs (PAT_d).

2.10.9.2. Heart Rate (HR)

The peak-to-peak time period of the PPG or ECG signals is used to calculate heart rate.

2.10.9.3. Augmentation Index (AI)

The augmentation index is defined as the ratio of the systolic peak to the inflection point to the pulse pressure. The inflection point corresponds to the time when the artery's blood flow reaches its maximum. The equation for AI is written as,

$$AI = \frac{x}{y} \quad \textbf{(2.18)}$$

2.10.9.4. Large Artery Stiffness Index (LASI)

The time interval between recorded systolic and diastolic peaks in each cycle is known as LASI, and it is used to calculate arterial stiffness. LASI is an indication of arterial stiffness and is inversely linked to the time interval between the systolic peak and the first inflection point following it.

2.10.9.5. Inflection Point Area Ratio (IPA)

The ratio of the areas under the PPG curve between chosen locations, represented by S1, S2, S3, and S4 in Figure 2.40, is defined as IPA. The ratio of heart pumping to pulse wave reflection in the PPG signal waveform may be utilized to calculate the impedance mismatch between different sections of the vascular system. As a result, IPA may be used to predict total peripheral resistance [70]. S1, S2, S3, and S4 regions are utilized directly as features in this research.

Table 2.2 Feature name and its definition

Feature Name	**Definition**
PIR	Photoplethysmogram Intensity Ratio
IH	Maximum amplitude of the pulse signal
IL	Minimum amplitude of the pulse signal
Meu_o	Viscosity of the blood
Sw10	Systolic width at 10% of the pulse amplitude
St10+Dt10	Addition of systolic and diastolic time at 10% of pulse amplitude
Dt10/St10	Division of diastolic time and systolic time at 10% of the pulse amplitude.
St25+Dt25	Addition of systolic time and diastolic time at 25% of the pulse amplitude
Dt25/St25	Division of diastolic time and systolic time at 25% of the pulse amplitude
St33+Dt33	Addition of systolic time and diastolic time at 33% of the pulse amplitude
Dt33/St33	Division of diastolic time and systolic time at 33% of the pulse amplitude.
Dt50/St50	Division of diastolic time and systolic time at 50% of the pulse amplitude
ST	Systolic Time
DT	Diastolic time
AI	Augmentation Index
LASI	Large Artery Stiffness Ratio
S1, S2, S3, S4	Inflection point area ratio
PAT_d, PAT_f, PAT_p	Pulse Arrival Time
Meu_n	Viscosity of the blood calculated via pulse pressure wave propagation

Table 2.2 (continued) Feature name and its definition

Feature Name	**Definition**
Omega	The frequency of heart rate
$Alpha_o$	Womersley number
$Alpha_n$	Womersley number (2^{nd} method)
SDI	Systolic Diastolic Time Interval
SDI_n	New Systolic Diastolic Interval
QT_c	The time interval between Q and T of ECG signal corrected for heart rate
RR	The time interval between two consecutive R waves of the ECG signals
QT	The time interval between Q and T of ECG signal
TQ	The time interval between T and Q of successive ECG signal
PR	From the beginning of the P wave to the beginning of the QRS complex
QR	From the beginning of the Q wave to the beginning of the QRS complex
QRS	From the start to the end of the QRS complex
P	Amplitude of P wave
Q	Amplitude of Q wave
R	Amplitude of R wave
S	Amplitude of S wave
T	Amplitude of T wave
SDI_c	Systolic diastolic interval corrected for RR interval
HR	Heart Rate

CHAPTER 3

IMPLEMENTATION

In this chapter, the implementations for blood pressure estimation by training the features extracted from the signals of the deep learning algorithms methods described in Chapter 2 are explained. Basically, the procedures for the diagnosis of hypertension were carried out in 5 basic steps. Firstly, data extraction and pre-processing were performed for subjects with blood pressure values (SBP greater than equal or 180, DBP greater than equal or 130, SBP greater than equal or 80, DBP greater than equal or 60) taken as reference from the MIMIC database. For the second step, feature extraction was performed from ECG and PPG signals using MATLAB.

Afterward, models were made using deep learning algorithms with the obtained features. With the models obtained, blood pressure estimation was made and compared with reference blood pressure values. In the next step, the performance measurement criteria values such as MAE, RMSE, MSE, and deviation were calculated according to the deep learning models. In addition, performance comparisons were made with the results of similar studies in the literature.

Finally, the development environment, development language, and its methods will be discussed. In addition, the steps of application of mentioned deep learning algorithms will be explained.

3.1. Implementation Architecture

The features of the data used in this thesis were extracted using MATLAB, which provides great convenience in terms of signal processing methods and techniques. Both the functions provided by the Python software language and the algorithms we apply were used in model development. The general view of the method proposed for the diagnosis of hypertension in the thesis is shown in Figure 3.1.

During the implementation process of the study, two different development environments, MATLAB and PyCharm, were used. Firstly, MATLAB is used to create the dataset and is used to convert data to CSV format. Python language is used as the programming language. Python 3.6.3 is selected as the python version. Python has many libraries for deep learning applications and can be applied quickly and easily. Torch, Torchvision, Numpy, Pandas, Matplotlib, Scikit-learn, and Os libraries are used in this study. Panda is used for reading and writing CSV files. The next library is Keras, which has packages for the classification of a convolutional neural network. Sklearn, the last and the most important library, supplies the most significant packages for this study. These packages are preprocessing, model_selection, metrics, CNN, and LSTM methods.

Thanks to these libraries and their packages, feature extraction, selection, model fitting, and evaluation can be done easily.

3.2. Application of Learning Methods

In this study, blood pressure estimations were made using ECG and PPG signals for different deep learning algorithms and their results were compared. This work consists of six steps for each algorithm as shown in Figure 3.1.

Figure 3.1 shows the block diagram of the proposed blood pressure prediction method, which consists of the following steps.

i. Database block, collection of a database.
ii. Preprocessing block to smooth and remove certain subject’s data whose records with very high blood pressure or very low blood pressure.
iii. Feature extraction block to extract useful features from signals.
iv. Partitioning block to partition the samples into three subsets train, validation, and test.
v. Deep Learning block to train the convolutional neural network (CNN) and long short-term memory (LSTM) models.
vi. Model Evaluation block to evaluate the trained model's performance.

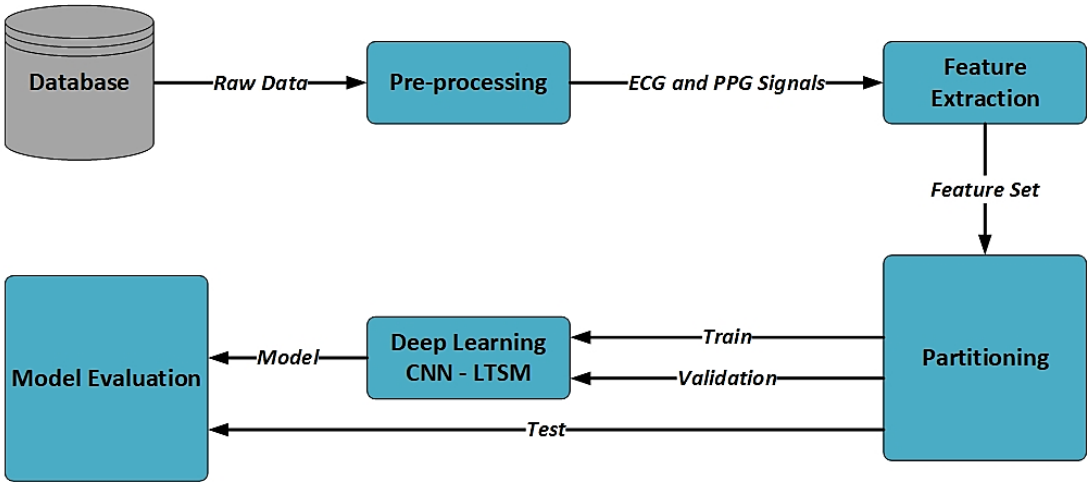


Figure 3.1 Block diagram of a proposed blood pressure prediction system

3.3. Training and Testing the Network

Once the features are extracted, the following are the various deep learning models considered to find the possible association between input features and blood pressure. The feature vectors are trained using all the models and compared with each other. Also, separate models are trained for predicting the targets such as SBP, DBP, and MBP.

In our work, we mainly discuss two ways that predict blood pressure value on the basis of the pulse wave parameters: Convolutional Neural Network and Long Short-Term Memory. Firstly, we divide the data set is then divided by 80% for training and 20% for testing. The following step is the creation of a model for each method with these train and test data. Training duration for 20 epochs with learning hyperparameters, that all methods were applied. Then, blood pressure is classified according to the model. Finally, evaluation metrics are calculated and interpreted by comparing results with related works. Figure 3.2 shows the flowchart of the proposed blood pressure prediction method, which consists of the following steps.

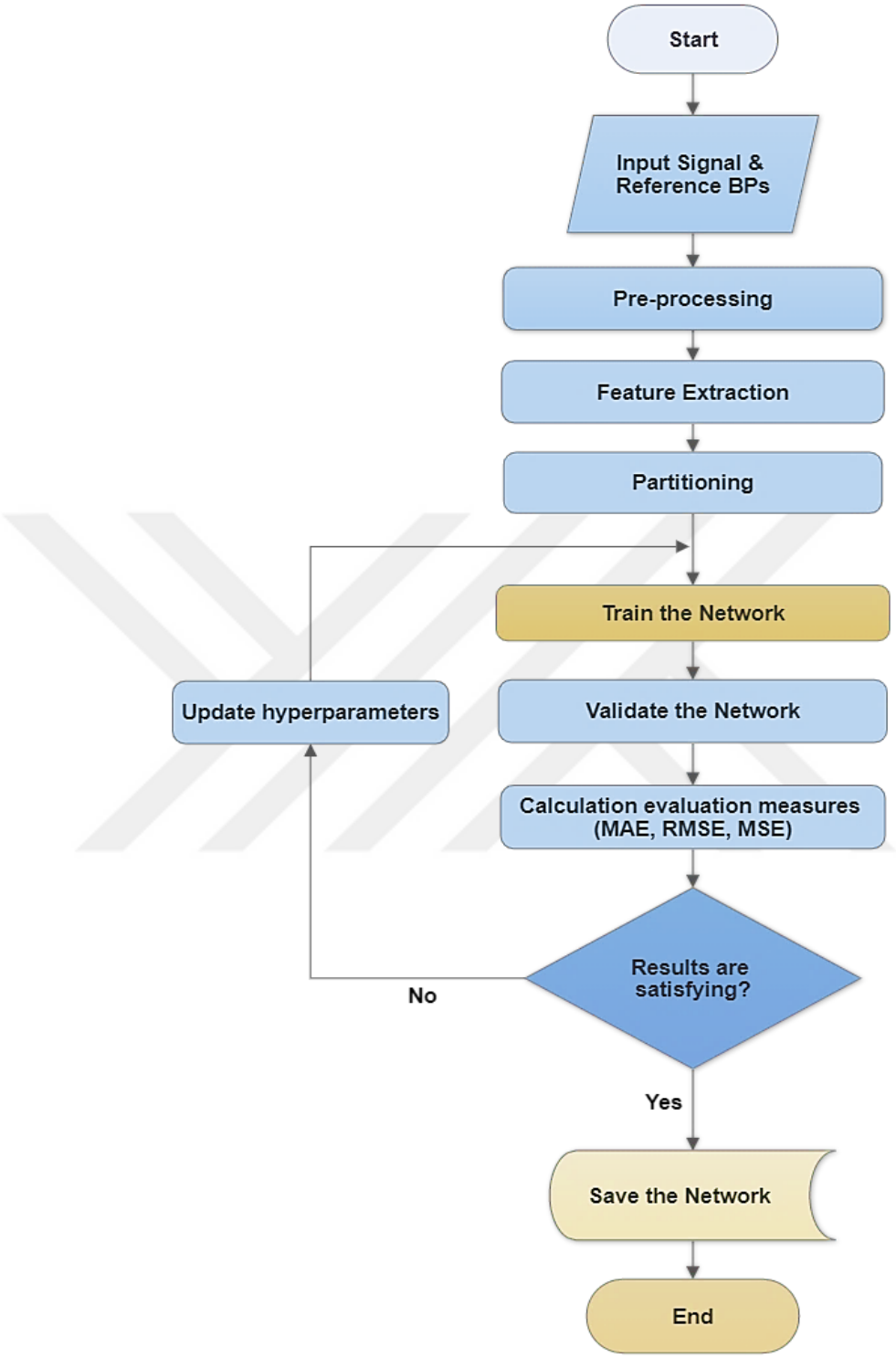


Figure 3.2 Flowchart of a proposed blood pressure prediction system

According to this pseudo code, all steps are executed in all methods and are applied to all datasets. The pseudo code of learning application is as follows:

```
BEGIN
READ dataset
SET train size to 0.80
SET test size to 0.20
specify input and output of dataset
FOR iteration = 0 to 20
  split the dataset into the training set and test set
  Scale feature
  Case based on Model
  Case "CNN"
     Fitting CNN with parameters to the Training Set
  Case "LSTM"
     Fitting LSTM with parameters to the Training Set
  Case "BiLSTM"
     Fitting BiLSTM with parameters to the Training Set
  Case "CNN-LSTM"
     Fitting CNN-LSTM with parameters to the Training Set
  END Case
  Make classifications
  Evaluate the model
END FOR
Print results to the console
END
```

3.4. Performance Criterias

In this study, evaluation metrics such as mean absolute error (MAE), root mean square error (RMSE), mean square error (MSE) and variance score were calculated and the results obtained in similar studies were compared.

Scikit-learn is one of the most widely used Python packages, is used to calculate these evaluation metrics. In the following, there are explanations and formulas of the metrics used respectively.

3.4.1. Mean Absolute Error (MAE)

The mean absolute error is a measure of the difference between two continuous variables. MAE calculates the average magnitude of errors in a series of forecasts without considering their direction. It is the average of the absolute differences between prediction and actual observation over the test sample, where all individual deviations are given equal weight. The following formula shows the calculation of the mean absolute error.

$$MAE = \frac{1}{\mathrm{n}} \sum_{j=1}^{n} e_j \quad \textbf{(3.1)}$$

Mean Squared Error can be calculated using the library "from sklearn.metrics import mean_absolute_error". As shown in the below code block, the first parameter taken by the function consists of the actual values and the second parameter consists of the estimated values.

```
mean_absolute_error(y_actual, y_predicted)
```

3.4.2. Mean Squared Error (MSE)

The mean squared error tells how close a regression curve is to a set of points. MSE measures the performance of a machine learning model, estimator, it is always positive, and it can be said that estimators with MSE value close to zero perform better.

The MSE can be calculated using the "from sklearn.metrics import mean_squared_error" library. The following formula shows the calculation of the mean squared error. It is also shown in the code block below.

$$MSE = \frac{1}{\mathrm{n}} \sum_{j=1}^{n} (y_j - \hat{y_j})^2 \quad \textbf{(3.2)}$$

```
mean_squared_error(y_actual, y_predicted)
```

3.4.3. Root Mean Squared Error (RMSE)

The root mean square error (RMSE) is a quadratic scoring method that calculates the average magnitude of the mistake. Root Mean Square Error (RMSE) is the square root of the MSE. It is used more often than MSE because sometimes the MSE value can be too large to be easily compared. Therefore, the MSE is calculated with the square of the error, thus facilitating interpretation. But MSE is very sensitive to outliers. While calculating the RMSE, numpy's sqrt function is used to take the square root. The following formula shows the calculation of the root mean square error. It is also shown in the code block below.

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2} \quad \textbf{(3.3)}$$

```
np.sqrt(mean_squared_error(y_actual, y_predicted))
```

3.4.4. Variance Score

The variance score is a measure of fitness for linear regression models. For the same data set, higher R-squared values represent smaller differences between observed data and fitted values. R^2 is a statistical measure of how close the data are to the fitted regression line. Also known as coefficient of determination or multiple coefficients of determination for multiple regression. The following formula shows the calculation of variance score. It is also shown in the code block below.

$$R^2 = 1 - \frac{Explained\ Variation}{Total\ Variation} \quad \textbf{(3.4)}$$

```
r2_score(y_actual, y_predicted)
```

3.5. Experimental Results

This scope of work, blood pressure estimation was made using four different deep learning algorithms. There are result tables for each algorithm. Table 3.1, Table 3.2, and Table 3.3 shows the performance measurement results obtained for SBP, DBP and MBP estimation.

The following Table 3.4, Table 3.5, and Table 3.6 below show a comparative analysis of performance measurement criteria like MAE, RMSE, MSE, and variance score for blood pressure estimation using machine learning and deep learning algorithms from relevant works in the literature with our proposed model. Additionally, in the proposed model, combinations of two algorithms display the improvement in the performance compared to CNN and LSTM deep learning algorithms separately. Finally, the proposed methods in this study, dataset, the algorithm used are feasibility in the reliable estimation of blood pressure is demonstrated.

Table 3.1 Performance results of different methods for SBP estimation

Method	**MAE**	**RMSE**	**MSE**	**r**
CNN	14.60	15.55	241.80	-
LSTM	9.52	10.20	104.04	-
BiLSTM	9.03	9.44	89.11	-
CNN-LSTM	5.75	6.90	47.61	0.98

Table 3.2 Performance results of different methods for DBP estimation

Method	**MAE**	**RMSE**	**MSE**	**r**
CNN	9.35	11.38	129.50	-
LSTM	8.68	9.12	83.18	-
BiLSTM	8.08	9.03	81.55	-
CNN-LSTM	3.88	4.80	23.04	0.92

Table 3.3 Performance results of different methods for MBP estimation

Method	**MAE**	**RMSE**	**MSE**	**r**
CNN	10.28	13.04	170.05	-
LSTM	7.79	9.18	84.27	-
BiLSTM	7.59	9.12	83.17	-
CNN-LSTM	4.08	5.10	26.01	0.97

Table 3.4 Comparison of performance with other works for SBP results

Work	**Method**	**MAE**	**RMSE**	**MSE**	**r**
Thambiraj et al. [4]	Random Forest (RF)	9.00	-	13.83	0.85
Kachuee et al. [5]	SVR	12.38	-	-	-
Kachuee et al. [5]	Random Forest (RF)	11.17	-	-	0.59
Attarpour et al. [71]	ANN	4.94	5.87	-	0.94
Tanveer et al. [8]	ANN-LSTM	1.09	1.56	-	-
Yang et al. [22]	ANN	8.19	-	-	-
Yang et al. [22]	SVM	8.41	-	-	-
Yang et al. [22]	LASSO	9.16	-	-	-
Simjanoska et al. [16]	Machine Learning	8.64	10.97	-	-
Chowdhury et al. [17]	GPR	3.02	6.74	45.49	0.95
Jeong et al. [20]	CNN-LSTM	0.02	-	-	-
Eom et al. [72]	CNN, Bi-GRU	4.06	5.42	-	-
Proposed Model	CNN-LSTM	5.75	6.90	47.61	0.98

Table 3.5 Comparison of performance with other works for DBP results

Work	**Method**	**MAE**	**RMSE**	**MSE**	**r**
Thambiraj et al. [4]	Random Forest (RF)	5.48	-	6.00	0.84
Kachuee et al. [5]	SVR	6.34	-	-	-
Kachuee et al. [5]	Random Forest (RF)	5.35	-	-	0.48
Attarpour et al. [71]	ANN	4.03	5.50	-	0.84
Tanveer et al. [8]	ANN-LSTM	0.58	0.84	-	-
Yang et al. [22]	ANN	5.12	-	-	-
Yang et al. [22]	SVM	5.81	-	-	-
Yang et al. [22]	LASSO	6.49	-	-	-
Simjanoska et al. [16]	Machine Learning	18.20	19.34	-	-
Chowdhury et al. [17]	GPR	1.74	3.59	12.89	0.96
Jeong et al. [20]	CNN-LSTM	0.16	-	-	-
Eom et al. [72]	CNN, Bi-GRU	3.33	4.30	-	-
Proposed Model	CNN-LSTM	3.88	4.80	23.04	0.92

Table 3.6 Comparison of performance with other works for MBP results

Work	**Method**	**MAE**	**RMSE**	**MSE**	**r**
Thambiraj et al. [4]	Random Forest (RF)	3.02	-	4.00	0.94
Kachuee et al. [5]	Random Forest (RF)	5.92	-	-	0.59
Simjanoska et al. [16]	Machine Learning	13.52	15.07	-	-
Proposed Model	CNN-LSTM	4.08	5.10	26.01	0.97

The Bland Altman plot, which represents the mean difference and average blood pressure, was created to demonstrate the agreement between the estimated and reference methods. The mean difference and the limits of agreement are seen in the following Figure 3.3, Figure 3.4, and Figure 3.5 for SBP, DBP, and MBP respectively. From the plot it is obvious that a maximum number of observations is within the limits of the agreement demonstrates the consensus between the estimated and reference method.

Bland-Altman analysis indicates the mean difference blood pressure within approximately ±6 mmHg for SBP (8.30 mmHg), DBP (4.75 mmHg), and MBP (6.05 mmHg). The Pearson's correlation test also showed correlation coefficients close to 1.0 for proposed CNN-LSTM model (0.98, 0.92 and 0.97, respectively); this shows high linearity between reference and estimated blood pressure.

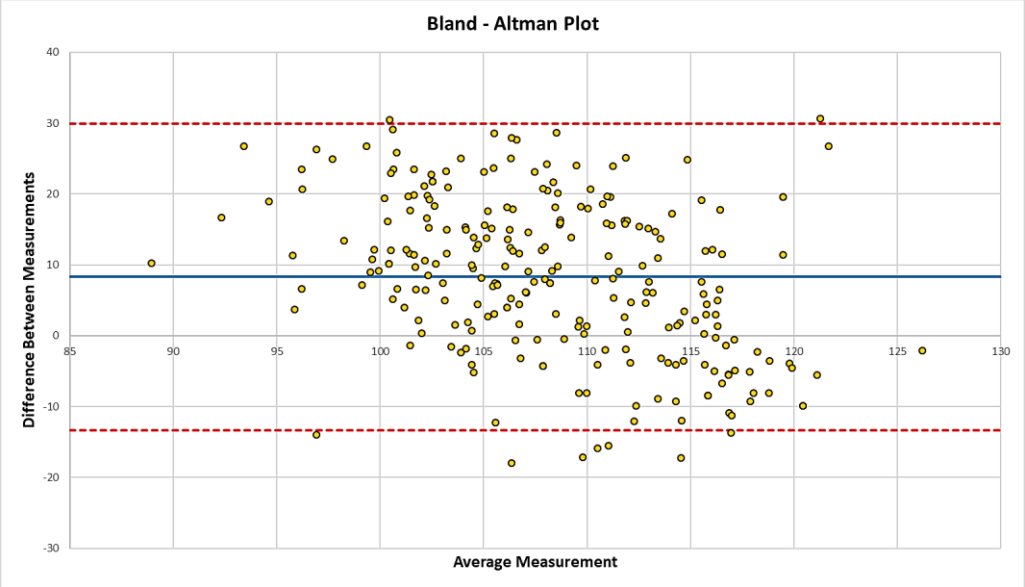


Figure 3.3 Bland–Altman plots of proposed model for SBP estimation

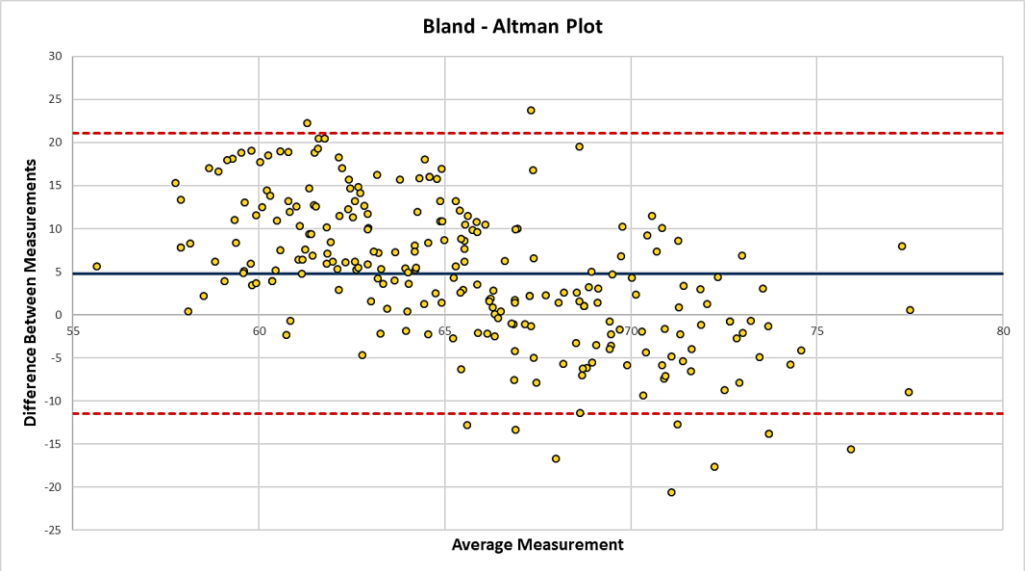


Figure 3.4 Bland–Altman plots of proposed model for DBP estimation

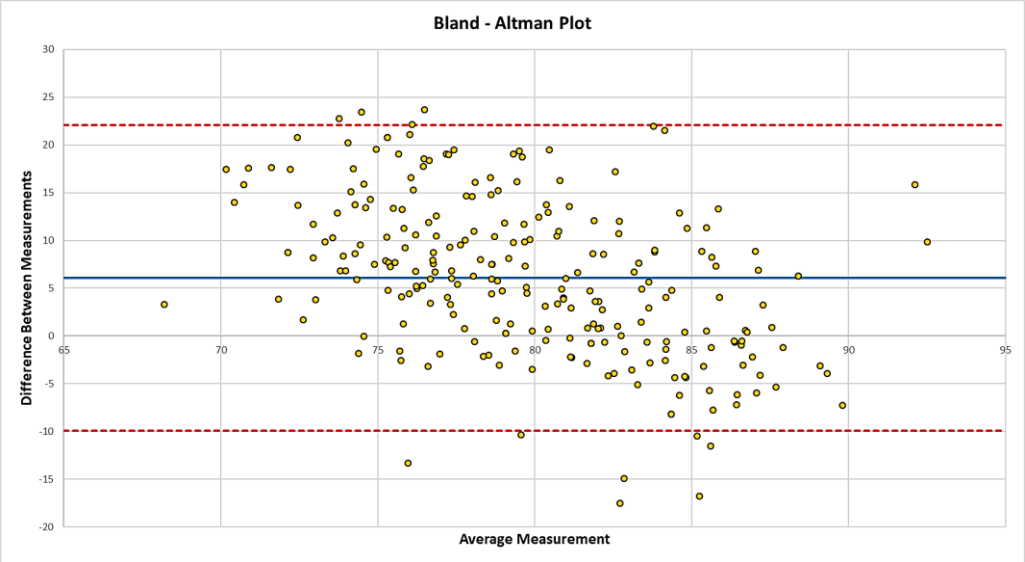


Figure 3.5 Bland–Altman plots of proposed model for MBP estimation

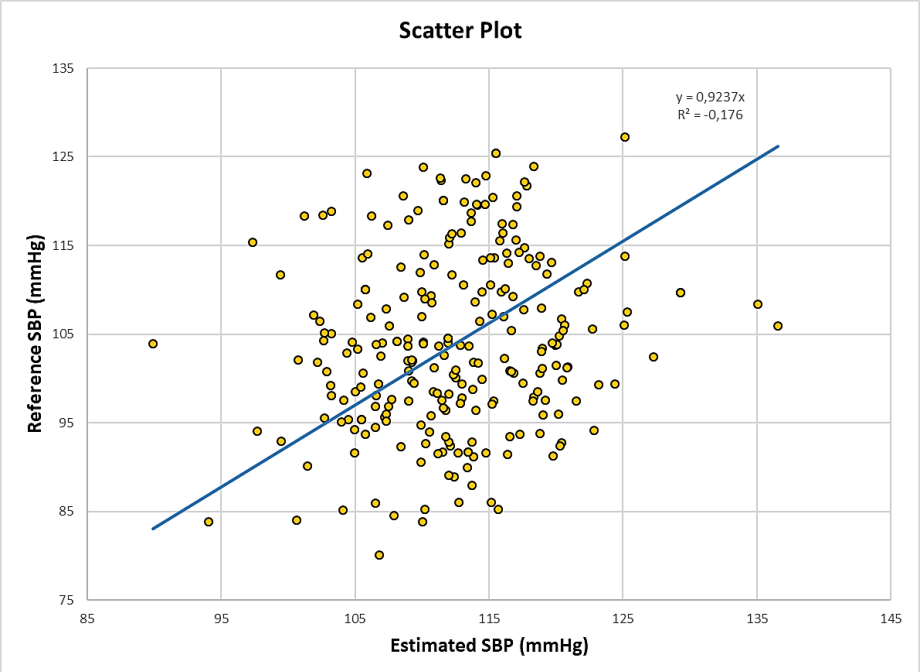


Figure 3.6 Scatter plots of of proposed model for SBP estimation

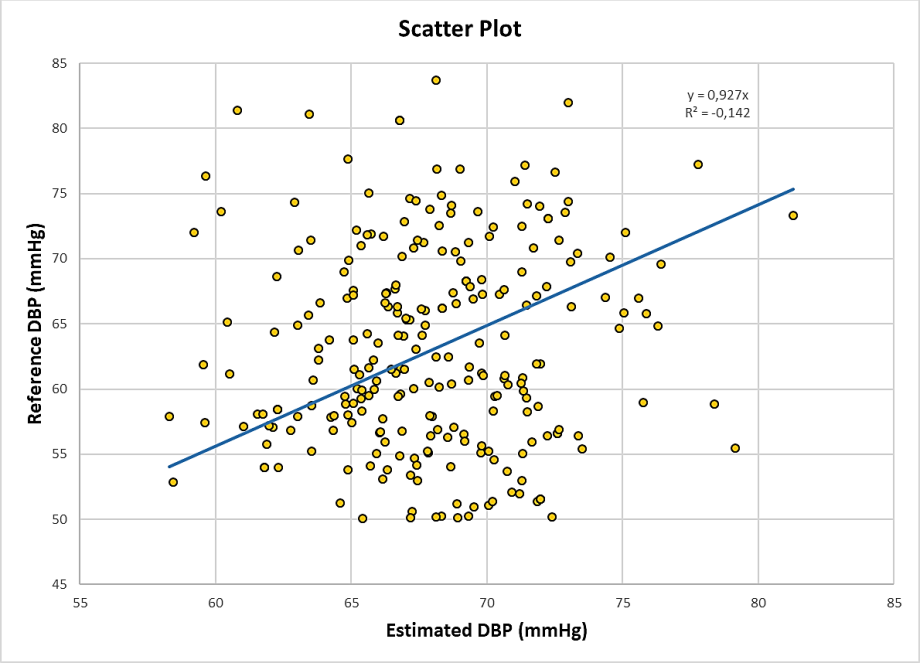


Figure 3.7 Scatter plots of of proposed model for DBP estimation

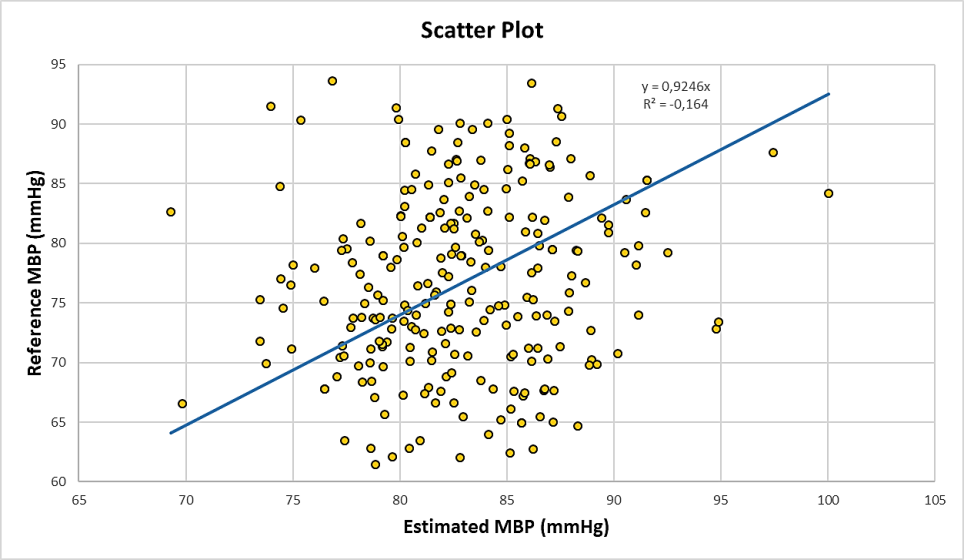


Figure 3.8 Scatter plots of of proposed model for MBP estimation

Scatter plots for the model results are presented in Figure 3.6, Figure 3.7, and Figure 3.8 with the coefficient of determination (R^2) indicated. The R^2 values were -0.176, -0.142, and -0.164 for the SBP, DBP, and MBP respectively. This means that the model learned more general patterns to estimate blood pressure with high reproducibility.

CHAPTER 4

CONCLUSION & FUTURE WORK

4.1. Conclusion

Determination of hypertension is of great importance in reducing the risk of heart, brain, vascular, and kidney diseases and the related morbidity and mortality rate. Diagnosis of hypertension is intricate, and it demands a continuous, accurate blood pressure measurement system. Therefore, it requires a continuous, accurate blood pressure measurement system to identify high-risk individuals. It is very important to measure blood pressure accurately for the diagnosis of hypertension. All existing continuous non-invasive techniques have challenges such as exact sensor placement, reconstructing arterial pressure from the finger cuff, frequent and subject-based calibration. In this study, a model was built from electrocardiography (ECG) and photoplethysmography (PPG) signals features for cuffless blood pressure prediction. The proposed method uses deep learning and signals processing algorithms to achieve this goal.

Non-invasive blood pressure assessment using deep learning algorithms is considered to be a matter of debate in terms of feature extraction and evaluation, accurate diagnostics, and the like. In this study, we have addressed the problem of continuous and non-invasive BP estimation in healthcare monitoring systems. Also, this study evaluates and compares the performance of various deep learning methods.

In this study, Python application has been developed with a convolutional neural network, long short-term memory, bidirectional long short-term memory deep learning methods. At the same time, the features were trained by combining the two algorithms. In this developed application, models were created by processing the data in CSV format obtained with the features extracted from the ECG and PPG signals. In the evaluation of the deep learning models obtained, metrics were calculated according to the reference blood pressure values and the estimated blood pressure values.

Primarily, the proposed methodology consists of feature extraction and pre-processing from electrocardiography and photoplethysmography signals. Afterward, we propose the method to predict blood pressure that extracts informative features like Womersley number, QRS, QTc interval, SDI from ECG and PPG signals, and deep learning techniques like CNN, LSTM, and BiLSTM which are employed to estimate blood pressure continuously. Finally, performance metrics like MAE, RMSE, MSE, and variance score are considered to validate the proposed method.

In the blood pressure estimation of our model, which we obtained by using convolutional neural networks and long-short-term memory algorithms together, it was concluded that the accuracy rate increased. At the end of this thesis, an automated version of blood pressure estimation from electrocardiography and photoplethysmography signals has been implemented, allowing the fastest and most accurate diagnosis of hypertension.

As conclusion, considering the results of the applied study, it has been shown that this computer aided diagnosis system can accelerate the clinical evaluation and treatment processes with high accuracy. According to the results of this study, it was revealed that the evaluation based on computational techniques in the diagnosis of non-invasive hypertension may provide more objectivity.

4.2. Future Works

Future work will be interested in improving the blood pressure prediction algorithm by increasing the number of data, the training set, and by testing different types of deep learning approaches. Finally, the methods used in the noninvasive blood pressure prediction we proposed, by integrating into wearable devices that are very popular in today's technology it can also be used to combat cardiovascular diseases and prevent their risky effects.

REFERENCES

[1] Türk Kardiyoloji Derneği, "17 Mayıs Dünya Hipertansiyon Günü Basın Açıklaması," 2018. http://tkd.org.tr/hipertansiyon-calisma-grubu/duyuru/3446/17-mayis-dunya-hipertansiyon-gunu-basin-aciklamasi (accessed May 20, 2021).

[2] G. Mancia *et al.*, "2013 ESH/ESC Guidelines for the management of arterial hypertension," *Eur. Heart J.*, vol. 34, no. 28, pp. 2159–2219, Jul. 2013, doi: 10.1093/eurheartj/eht151.

[3] K. J. Weber MA, Schiffrin EL, White WB, Mann S, Lindholm LH, *Clinical practice guidelines for the management of hypertension in the community: a statement by the American Society of Hypertension and the International Society of Hypertension*. J Clin Hypertens (Greenwich), 2014.

[4] G. Thambiraj, U. Gandhi, U. Mangalanathan, V. J. M. Jose, and M. Anand, "Investigation on the effect of Womersley number, ECG and PPG features for cuff less blood pressure estimation using machine learning," *Biomed. Signal Process. Control*, vol. 60, p. 101942, 2020, doi: 10.1016/j.bspc.2020.101942.

[5] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cuffless Blood Pressure Estimation Algorithms for Continuous Health-Care Monitoring," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 859–869, 2017, doi: 10.1109/TBME.2016.2580904.

[6] Y. Zhang and Z. Feng, "A SVM method for continuous blood pressure estimation from a PPG signal," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1283, pp. 128–132, 2017, doi: 10.1145/3055635.3056634.

[7] A. Paviglianiti, V. Randazzo, E. Pasero, and A. Vallan, "Noninvasive Arterial Blood Pressure Estimation using ABPNet and VITAL-ECG," *I2MTC 2020 - Int. Instrum. Meas. Technol. Conf. Proc.*, pp. 1–5, 2020, doi: 10.1109/I2MTC43012.2020.9129361.

[8] M. S. Tanveer and M. K. Hasan, "Cuffless blood pressure estimation from electrocardiogram and photoplethysmogram using waveform based ANN-LSTM network," *Biomed. Signal Process. Control*, vol. 51, pp. 382–392, 2019, doi: 10.1016/j.bspc.2019.02.028.

[9] S. Ghosh, A. Banerjee, N. Ray, P. W. Wood, P. Boulanger, and R. Padwal, "Continuous blood pressure prediction from pulse transit time using ECG and PPG signals," *2016 IEEE Healthc. Innov. Point-of-Care Technol. Conf. HI-POCT 2016*, pp. 188–191, 2016, doi: 10.1109/HIC.2016.7797728.

[10] R. Wang, W. Jia, Z. H. Mao, R. J. Sclabassi, and M. Sun, "Cuff-free blood pressure estimation using pulse transit time and heart rate," *Int. Conf. Signal Process. Proceedings, ICSP*, vol. 2015-Janua, no. October, pp. 115–118, 2014, doi: 10.1109/ICOSP.2014.7014980.

[11] Y. Kurylyak, F. Lamonaca, and D. Grimaldi, "A Neural Network-based method for continuous blood pressure estimation from a PPG signal," *Conf. Rec. - IEEE Instrum. Meas. Technol. Conf.*, pp. 280–283, 2013, doi: 10.1109/I2MTC.2013.6555424.

[12] A. Sahoo, P. Manimegalai, and K. Thanushkodi, "Wavelet based pulse rate and blood pressure estimation system from ECG and PPG signals," *2011 Int. Conf. Comput. Commun. Electr. Technol. ICCCET 2011*, pp. 285–289, 2011, doi: 10.1109/ICCCET.2011.5762486.

[13] Y. Choi, Q. Zhang, and S. Ko, "Noninvasive cuffless blood pressure estimation using pulse transit time and Hilbert-Huang transform," *Comput. Electr. Eng.*, vol. 39, no. 1, pp. 103–111, 2013, doi: 10.1016/j.compeleceng.2012.09.005.

[14] I. Sharifi, S. Goudarzi, and M. B. Khodabakhshi, "A novel dynamical approach in continuous cuffless blood pressure estimation based on ECG and PPG signals," *Artif. Intell. Med.*, vol. 97, no. April 2018, pp. 143–151, 2019, doi: 10.1016/j.artmed.2018.12.005.

[15] S. Yang, W. S. W. Zaki, S. P. Morgan, S. Y. Cho, R. Correia, and Y. Zhang, "Blood pressure estimation with complexity features from electrocardiogram and photoplethysmogram signals," *Opt. Quantum Electron.*, vol. 52, no. 3, 2020, doi: 10.1007/s11082-020-2260-7.

[16] M. Simjanoska, M. Gjoreski, M. Gams, and A. M. Bogdanova, "Non-invasive blood pressure estimation from ECG using machine learning techniques," *Sensors (Switzerland)*, vol. 18, no. 4, pp. 1–20, 2018, doi: 10.3390/s18041160.

[17] M. H. Chowdhury *et al.*, "Estimating blood pressure from the photoplethysmogram signal and demographic features using machine learning techniques," *Sensors (Switzerland)*, vol. 20, no. 11, 2020, doi: 10.3390/s20113127.

[18] S. G. Khalid, J. Zhang, F. Chen, and D. Zheng, "Blood Pressure Estimation Using Photoplethysmography Only: Comparison between Different Machine Learning Approaches," *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/1548647.

[19] J. Esmaelpoor, M. H. Moradi, and A. Kadkhodamohammadi, "A multistage deep neural network model for blood pressure estimation using photoplethysmogram signals," *Comput. Biol. Med.*, vol. 120, no. 350, p. 103719, 2020, doi: 10.1016/j.compbiomed.2020.103719.

[20] D. U. Jeong and K. M. Lim, "Combined deep CNN–LSTM network-based multitasking learning architecture for noninvasive continuous blood pressure estimation using difference in ECG-PPG features," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, 2021, doi: 10.1038/s41598-021-92997-0.

[21] D. Lee *et al.*, "Beat-to-beat continuous blood pressure estimation using bidirectional long short-term memory network," *Sensors (Switzerland)*, vol. 21, no. 1, pp. 1–15, 2021, doi: 10.3390/s21010096.

[22] S. Yang, Y. Zhang, S. Y. Cho, R. Correia, and S. P. Morgan, "Non-invasive cuff-less blood pressure estimation using a hybrid deep learning model," *Opt. Quantum Electron.*, vol. 53, no. 2, pp. 1–20, 2021, doi: 10.1007/s11082-020-02667-0.

[23] M. Panwar, A. Gautam, D. Biswas, and A. Acharyya, "PP-Net: A Deep Learning Framework for PPG-Based Blood Pressure and Heart Rate Estimation," *IEEE Sens. J.*, vol. 20, no. 17, pp. 10000–10011, 2020, doi: 10.1109/JSEN.2020.2990864.

[24] Microsoft, "What is machine learning?," 2021. https://azure.microsoft.com/en-gb/overview/what-is-machine-learning-platform/ (accessed May 20, 2021).

[25] Mitchell T. M., *Machine Learning*. New York: NY: McGraw-Hill, 1997.

[26] Qualcomm, "Data collection and pre-processing techniques." https://developer.qualcomm.com/software/qualcomm-neural-processing-sdk/learning-resources/ai-ml-android-neural-processing/data-collection-pre-processing (accessed May 15, 2021).

[27] Prem, "A Simple Introduction to Online Machine Learning," *iunera*. https://www.iunera.com/kraken/fabric/simple-introduction-to-online-learning-in-machine-learning/ (accessed May 12, 2021).

[28] R. van Loon, "Machine learning explained: Understanding supervised, unsupervised, and reinforcement learning," *Bigdata-Madesimple.Com*, 2018. https://bigdata-madesimple.com/machine-learning-explained-understanding-supervised-unsupervised-and-reinforcement-learning/ (accessed May 21, 2021).

[29] Y. Shweta Bhatt, "What is reinforcement learning?," *KDnuggets*. https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html (accessed Apr. 27, 2021).

[30] VectorStock, "Typical human neuron vector image." https://www.vectorstock.com/royalty-free-vector/typical-human-neuron-vector-10500269 (accessed Apr. 08, 2021).

[31] M. Z. Sampe, E. Ariawan, and I. W. Ariawan, "Predictive Analysis of Employee Loyalty: A Comparative Study Using Logistic Regression Model and Artificial Neural Network," *J. Indones. Math. Soc.*, vol. 25, no. 3, pp. 325–335, 2019, doi: 10.22342/jims.25.3.825.325-335.

[32] J. Feng, X. He, Q. Teng, C. Ren, H. Chen, and Y. Li, “Reconstruction of porous media from extremely limited information using conditional generative adversarial networks,” *Phys. Rev. E*, vol. 100, no. 3, p. 33308, 2019, doi: 10.1103/PhysRevE.100.033308.

[33] figshare, “Basic structure of ANN model.” https://figshare.com/articles/figure/Basic_structure_of_ANN_model_/13014973/1 (accessed May 11, 2021).

[34] McCulloch W. S. and Pitts W., *A logical calculus of the ideas immanent in nervous activity," The bulletin of mathematical biophysics*, 5th ed. 1943.

[35] Rosenblatt F., *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, 65th ed. 1958.

[36] Sebastian Raschka, “Single-Layer Neural Networks and Gradient Descent,” 2015. https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html (accessed Jul. 01, 2021).

[37] Kevin Swingle, “Single Layer Perceptrons.” http://www.cs.stir.ac.uk/courses/ITNP4B/lectures/kms/2-Perceptrons.pdf (accessed Jul. 02, 2021).

[38] Vinh Khuc, “Backpropagation In Neural Networks,” 2015. http://vinhkhuc.github.io/2015/03/29/backpropagation.html (accessed May 03, 2021).

[39] K. U. Yılmaz A., *Derin Öğrenme*. 2021.

[40] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, 7th ed. Found. Trends® Signal Process, 2014.

[41] Y. Bengio, *Learning Deep Architectures for AI*, 1st ed. Found. trends® Mach. Learn, 2009.

[42] H. A. Song and S.-Y. Lee, *Hierarchical Representation Using NMF,” in International Conference on Neural Information Processing*. 2013.

[43] Zbigatron, “Why Deep Learning Has Not Superseded Traditional Computer Vision,” 2018. https://zbigatron.com/has-deep-learning-superseded-traditional-computer-vision-techniques/ (accessed Apr. 23, 2021).

[44] E. Aminanto and K. Kim, “Deep learning in intrusion detection system: An overview,” 2016.

[45] H. Mishra, “How does the neuronal activity of the brain behave in the days following the onset of blindness?,” *Quora*. https://www.quora.com/How-does-the-neuronal-activity-of-the-brain-behave-in-the-days-following-the-onset-of-blindness (accessed May 29, 2021).

[46] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way,” *Towards Data Science*, 2018. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed Jun. 03, 2021).

[47] Y. L., “A Summary of Neural Network Layers,” *Medium*, 2018. http://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e (accessed May 21, 2021).

[48] T. Pandit, A. Kapoor, R. Shah, and R. Bhuva, “Understanding Inception Network Architecture for Image Understanding Inception Network Architecture for,” no. February, pp. 0–5, 2020, doi: 10.13140/RG.2.2.16212.35204.

[49] ReNom, “Convolutional Neural Network(CNN),” *ReNom*. .

[50] Anil Bhatt, “CNN - Activation Functions, Global Average Pooling, Softmax, Negative Likelihood Loss,” 2020. .

[51] V. Chouhan, “Skin Lesion Analysis towards Melanoma Detection with Deep Convolutional Neural Network,” no. November 2019, 2019.

[52] Chakrabarty N., “A Novel Strategy for Gender Identification from Hand Dorsal Images using Computer Vision,” 2019.

[53] SuperDataScience Team, “Convolutional Neural Networks (CNN): Step 3 - Flattening,” *SuperDataScience Team*, 2018. .

[54] Matthijs Hollemans, “Matrix Multiplication with Metal Performance Shaders,” 2017. .

[55] D. Battini, “Implementing Drop Out Regularization in Neural Networks.” .

[56] V. Zhou, “A Simple Explanation of the Softmax Function,” 2019. .

[57] D. J. E. M.Y. Rafiq, G. Bugmann, *Neural network design for engineering applications*. 2001.

[58] and P. H. Y. LeCun, L. Bottou, Y. Bengio, “Gradient-based learning applied to document recognition,” 1998.

[59] and L. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and D. Jackel, *Handwritten digit recognition with a backpropagation network*. .

[60] C. Olah, “Understanding LSTM Networks,” 2015. .

[61] Lee C., “Understanding Bidirectional RNN in PyTorch,” *Towards Data Science*, 2017. http://towardsdatascience.com/understanding-bidirectional-rnn-in-pytorch-5bd25a5dd66 (accessed May 22, 2021).

[62] Afshine Amidi and Shervine Amidi, “Recurrent Neural Networks.” .

[63] Andrej Karpathy, “The Unreasonable Effectiveness of Recurrent Neural Networks,” 2015. .

[64] C. P. and H. S. A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. Ivanov, R. Mark, J. Mietus, G. Moody, *Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals*, 101st ed. 2000.

[65] “Sinus rhythm,” *Wikipedia*. .

[66] N. Hasanzadeh, M. M. Ahmadi, and H. Mohammadzade, “Blood Pressure Estimation Hasanzadeh, N., Ahmadi, M. M., & Mohammadzade, H. (2020). Blood Pressure Estimation Using Photoplethysmogram Signal and Its Morphological Features. IEEE Sensors Journal, 20(8), 4300–4310. https://doi.org/10.1109/JSEN.2019.2961411,” *IEEE Sens. J.*, vol. 20, no. 8, pp. 4300–4310, 2020.

[67] A. Szczepański and K. Saeed, “A mobile device system for early warning of ECG anomalies,” *Sensors (Switzerland)*, vol. 14, no. 6, pp. 11031–11044, 2014, doi: 10.3390/s140611031.

[68] A. A. Fossa *et al.*, “Analyses of dynamic beat-to-beat QT-TQ interval (ECG restitution) changes in humans under normal sinus rhythm and prior to an event of torsades de pointes during QT prolongation caused by sotalol,” *Ann. Noninvasive Electrocardiol.*, vol. 12, no. 4, pp. 338–348, 2007, doi: 10.1111/j.1542-474X.2007.00183.x.

[69] M. Mehrabbeik and S. Rashidi, “Estimation of Cuffless Systolic and Diastolic Blood Pressure Using Pulse Transient Time,” *5th Iran. Conf. Signal Process. Intell. Syst. ICSPIS 2019*, no. May 2020, 2019, doi: 10.1109/ICSPIS48872.2019.9066035.

[70] M. Elgendi, *On the analysis of fingertip photoplethysmogram signals, Current Cardiol. Rev*, 8th ed. 2012.

[71] A. Attarpour, A. Mahnam, A. Aminitabar, and H. Samani, “Cuff-less continuous measurement of blood pressure using wrist and fingertip photo-plethysmograms: Evaluation and feature analysis,” *Biomed. Signal Process. Control*, vol. 49, pp. 212–220, 2019, doi: 10.1016/j.bspc.2018.12.006.

[72] H. Eom *et al.*, “End-to-end deep learning architecture for continuous blood pressure estimation using attention mechanism,” *Sensors (Switzerland)*, vol. 20, no. 8, pp. 1–20, 2020, doi: 10.3390/s20082338.

CIRRICULUM VITAE

PERSONAL INFORMATION

Name Surname **:** Tuğba YILMAZ

EDUCATION

Bachelor **:** Ankara Yildirim Beyazit University,
Computer Engineering (2013-2018) GPA: 3.17/4.00

Master's degree **:** Ankara Yildirim Beyazit University,
Computer Engineering (2019-2021) GPA: 3.75/4.00

WORK EXPERIENCE

Scientific Programs Assistant Specialist **:** November 2018 –
TÜBİTAK

Intern **:** June 2017 – July 2017
ASELSAN A.Ş., Ankara

Intern **:** July 2016 – August 2016
TUSAŞ - TAI Turkish Aerospace Industries, Ankara

TECHNICAL SKILLS

Programming Languages: Java, C, Python, .Net

UI Frameworks: JQuery UI

Database Programs: Oracle, Postgres, MS SQL

CONFERENCE PRESENTATIONS

Yılmaz, T.; Şen, B., "***Automated Diagnostic Tool for Hypertension Using Deep Learning Model***", Global Conference on Engineering Research, GLOBCER 2021, Balıkesir.

October 2021
ANKARA

Department of Computer Engineering

Tuğba YILMAZ