Thesis/Project No.: CSER-23-68

# A Study on Non-Invasive Hypertension Classification Techniques from PPG Signal Using Deep Learning Models.

By

**Rifah Tasnim Haque Promi**

Roll: 1707052

&

**Rezwana Akter Nazri**

Roll: 1707120

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

**June 2025**

# A Study on Non-Invasive Hypertension Classification Techniques from PPG Signal Using Deep Learning Models.

By

**Rifah Tasnim Haque Promi**

Roll: 1707052

&

**Rezwana Akter Nazri**

Roll: 1707120

A thesis submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering

Supervisor:

S M Taslim Uddin Raju
Lecturer
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

Signature

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
**June 2025**

# Acknowledgement

All the praise to the almighty Allah, whose blessing and mercy succeeded us to complete this thesis work fairly. After that, we humbly acknowledge the valuable suggestions, advice, guidance and sincere co-operation of S M Taslim Uddin Raju, Lecturer, Department of Computer Science and Engineering, Khulna University of Engineering & Technology, under whose supervision this work was carried out. His intellectual advice, encouragement and guidance make us feel confident and scientific research needs much effort in learning and applying and need to have a broad view at problems from different perspective. We would like to convey our heartiest gratitude to all the faculty members, official and staffs of the Department of Computer Science and Engineering. Last but not least, we wish to thank our friends and family members for their constant supports.

**- Authors**

Rifah Tasnim Haque Promi (Roll: 1707052)

Rezwana Akter Nazri (Roll: 1707120)

# Abstract

Hypertension or high blood pressure is the leading risk factor with the largest contribution to the burden of disease and mortality. According to World Health Organization (WHO), hypertension is a serious medical condition that greatly increases the chances of getting diseases of the heart, brain, kidneys, and other organs. Therefore, regular blood pressure (BP) checks are necessary to track hypertension and reduce risk.In the light of this, we have devised a non-invasive method based on photoplethysmography (PPG) signal and demographic features using deep learning techniques for hypertension detection in the literature. PPG signals were acquired from 219 subjects, which undergo pre-processing steps. Finally, hypertension was detected by using 2 proposed models. Proposed model 1 constructed with Convolutional Neural Network (CNN) and Long Short-Term Memory( LSTM) achieved an Accuracy of 81.06 %, Recall of 86.80 %, Precision of 84.24% and F1-score of 85.5%. Proposed model 2 constructed with CNN and Gated Recurrent Unit (GRU) achieved an Accuracy of 85.00 %, Recall of 84.44 %, Precision of 87.77% and F1-score of 86.08%. Our purpose is to detect hypertension more precisely to enhance the medical field.

# Contents

# List of Figures

# List of Tables

# Chapter I

# Introduction

## 1.1 Introduction

Hypertension is a chronic clinical disorder characterized by persistently increased BP in the vessels. BP is the force that moves blood through the circulatory system. There are a variety of potentially fatal health issues that can be exacerbated by chronic hypertension, such as: strokes, heart disease, heart attacks, aortic aneurysms, vascular dementia etc. As stated by the World Health Organization (WHO) , an estimated 1.28 billion 30-to-79-year-old adults have hypertension. Over two-thirds of the world's hypertensive population resides in middle-income and low-income nations. Tragically, less than 20% of persons with hypertension have the condition under control. [2].

Convenient and precise BP assessment is critical for the prevention, evaluation, and medication of hypertension [3]. For this reason, an inexpensive, non-invasive and comfortable BP estimation method is needed. There are several non-invasive methods for BP estimation. Although standard cuff-based BP monitoring delivers precise readings, it cannot be utilized for regular monitoring and might be uncomfortable and inconvenient for the patient [4]. The reference BP can also be measured with a BP monitor [5]. However, if the heart is beating too fast or slow, assessment will be difficult using this method. Volume clamp method can also be utilized. But, this method often overestimates systolic BP. Another method is smartphone based approach with the complication that smartphone PPG signals an be easily affected by motion and noise artifact [6].

Recently, photoplethysmography (PPG) has been used as a potentially useful non-invasive method for continuous BP measurement [7]. It is a useful technique to detect blood volume changes at a low cost. PPG device consists of light emitting diodes (LED) to illuminate the dermal tissue and detect variations in blood volume by determining the intensity of the reflected light. There are some characteristics of PPG [1] which is important to measure blood pressure are systolic peak, diastolic peak, dicrotic notch, reflected wave peak. Recently, advancements in this technique have shown encouraging outcomes in terms of accuracy, practicality, and clinical acceptance [8]. So, researchers from all over the world have currently shown a lot of interest in obtaining useful information from the PPG signal.

## 1.2 Motivation

Cardiovascular diseases (CVD) are the leading cause of worldwide mortality, and since hypertension is a major predictor of CVD, proper monitoring and management of blood pressure (BP) is important. However, a discrete BP measurement, generally performed in clinical settings with a conventional cuff-type oscillometric device, can lead to misdiagnosis and cannot observe circadian fluctuations of BP. Hence conventional cuff-based method is not practical due to its inconvenient and cumbersome nature. Nowadays, PPG is considered as one of the best method for clinical applications, allowing simple, unobtrusive and inexpensive way of monitoring the physiological parameters ubiquitously which is also observed in recent research. Therefore, in this system we have developed a non-invasive hypertension classification technique based on PPG signal.

## 1.3 Problem Statement

Hypertension (or high blood pressure [BP]) is a significant health issue for adults that can result in serious complications, such as stroke and heart diseases. Continuous BP monitoring is critical for groups of patients other than those with hypertension. For example, in patients with spinal cord injury (SCI), an abrupt rise in BP (in excess of 20 mmHg SBP) caused by autonomic dysreflexia (AD) (an autonomic reflex response to nociceptive stimuli), could lead to disabling headache, seizure, cerebral hemorrhage, and even death. As such, accurate and continuous monitoring of BP is critical for early detection of episodes of AD in these patients, to enable them with seeking timely medical treatments. There have been many non-invasive methods for blood pressure measurement such as the mercury sphygmomanometer, oscillometry, volume clamping . Most non-invasive devices used for blood pressure measurement are cumbersome to carry, time-consuming to use, and inconvenient for portability

## 1.4 Objectives

Our main purpose of the thesis is to develop a non-invasive hypertension detection technique from PPG signal. Therefore, the objectives of the thesis are to:

- Study existing systems for hypertension detection.
- To find the best method for Non-Invasive hypertension detection.
- Prediction with more accuracy than existing other methods.
- Improve evaluation parameters.
- Build a robust and inexpensive system that can predict hypertension as accurately as possible

## 1.5 Methodology

In this system, our proposed hypertension detection methods is a non-invasive method based on PPG and demographic features. At first, PPG signal is preprocessed. Z-score normalization was performed on the data for data scaling. The signal filtered with butterworth low pass filter to eliminate noise. The first model is constructed with 3 CNNs and 1 LSTM. It is developped by stacking the 1D convolutional layers,ReLU activation, batch normalization and max-pooling layers in CNN. Then LSTM is added with a dropout layer to prevent overfitting, Which is followed by 1 flatten layer and 3 dense layer.

The second model is developped with 2 CNNs and 1 GRU. It is construced by stacking time distributed 1D convolutional layers, max-pooling layers in CNN which is follwed by flatten and dropout layes. Then 1 GRU is applied with a dropout layer. Finally, 3 dense layers were applied. The second model performed better than the first model with an Accuracy of 85.00 %.

## 1.6 Contributions

Our contributions in thesis system are listed as follows :

- Formation of a simple, accurate and inexpensive non-invasive hypertension detection method using PPG signal.
- Preprocess the input data to get noiseless PPG signals.
- Constraction of a hybrid model that will automatically extract features.
- Development of model 1 and model 2 for hypertension detection using extracted features.
- Application of deep learning models for multiclass classification.

## 1.7 Thesis Organization

**Chapter II:** In this chapter, we reviewed some earlier research related to hypertension detection. It includes deep learning and machine learning based approaches.

**Chapter III:**In this chapter, we presented some theoretical approaches for developing our model.

**Chapter IV:** In this chapter, we illustrated our proposed methodology and discussed about our proposed models.

**Chapter V:** In this chapter, we examined our model and demonstrated the results while comparing them with existing other model in different experiments.

**Chapter VI:** We summarized the entire thesis, outlined our future plans, and concluded the thesis in this chapter.

# Chapter II

# Literature Review

## 2.1 Introduction

Hypertension is an important disease seen among the public, and early detection of hypertension is significant for early treatment. There are several studies in the literature on the detection and classification of hypertension in recent years. They have used statistical features and some others used morphological features and wavelet transformed features from raw PPG signals to predict hypertension. The field of biomedical signal processing applications has shown increased interest in deep learning approaches.

## 2.2 Related work

Mangathayaru et al. [9] suggested an innovative approach for the prediction of hypertension utilizing a dual-tree complex wavelet transform (DT-CWT)-based feature extraction algorithm with a Gated Recurrent Unit (GRU) network. They have utilized the BIDMC PPG and Respiration Dataset which is freely available on PhysioBank. Using a Chebyshev II with bandpass filter of fourth order with a cutoff frequency range of 0.5–12 Hz, the pandemonium in PPG signals is removed. In Step 2, preprocessed signals were used to extract the most important characteristics. GRU classifier receives as input attributes that were extracted using DT-CWT method. The classifier assigns each Feature to its respective class label. The proposed method (when used in conjunction with DTCWT and GRU's) greatly increases the classification and performance of the system and provide precise detections with an accuracy rate of 98.8 %.

Wu et al. [10] established a new algorithm for predicting blood pressure classification from PPG by matching Continuous Wavelet Transform (CWT) type and segment length. Both wavelet type and segment length could affect the accuracy. The data were obtained from

PhysioNet's Multiparameter Intelligent Monitoring in Intensive Care-III (MIMIC-III) Waveform Database. Each PPG segment was initially processed with a moving average filter to smooth the PPG signal. Scalograms of the PPG signal are generated using a variety of wavelet transforms, including Frequency B-Spline wavelet (fbsp1-15-1), Shannon (shan15-1), Complex Gaussian wavelet (cgau1), Morlet wavelet(morl), Mexican hat wavelet (mexh), and Gaussian wavelet (gaus1). Using CNN for BP classification, the combination of cgau1 CWT and segment-300 (2.4 s) in the work is the optimal option. This method provided an accuracy of 90 % without obvious overfitting. The limitation of this paper is not all wavelets were evaluated on the MIMIC dataset, hence cgau1 might not be the optimum CWT for BP classification.

Tjahjadi et al. [11] suggested a comprehensive method for classifying BP based on PPG signal utilizing bidirectional long-short-term memory (Bi-LSTM) and time-frequency (TF) analysis. They used the PPG-BP dataset. In data preprocessing step, data exploration is carried out by using a spectrum. The TF analysis retrieves information from PPG signals using a short-time Fourier transform (STFT) in the time domain to generate two features, instantaneous frequency and spectrum entropy. Time frequency moment can be employed as a one-dimensional input to Bi-LSTM networks. Training the Bi-LSTM network with TF features greatly enhances classification performance and reduces training time. Bi-LSTM classifiers are then utilized to classify the BP.

Ivan et al. [12] constructed a BP classification model as a convolutional neural network (CNN) – long short term memory (LSTM) hybrid model, with layers of both CNN and LSTM. They used the "Cuff-Less Blood Pressure Estimation Data Set", published in the UCI Machine Learning Repository. The PPG signal is filtered with a filter of the fourth order and the ECG signal with a filter of the fifth order. The class imbalance problem in the dataset was solved by assigning weight to the classes. Multivariate sequential CNN-LSTM architecture was employed for the model. It is a hybrid neural network incorporating both CNN and LSTM layers. The proposed model achieves an accuracy rate of 83 % and stability across all metric classes (AUCROC for each class is 0.89, 0.83, and 0.90, correspondingly).

Eric et al. [13] proposed to utilize the wavelet scattering transform( WST) as a feature extraction technique to get features from PPG data and integrate it with clinical data to identify early hypertension stages by utilizing Early and Late Fusion. In contrast to other transforms such as the Fourier Transform (FT) and Wavelet Transform( WT), the WST produces a translation-invariant representation that is stable to mild time-warping deformations in order to build a signal representation that is suited for classification tasks. They collected the dataset from the Guilin People's Hospital in Guilin, China. To handle multimodal data in the context of ML, the literature opted for model-agnostic techniques. These methods are classified as Early Fusion (feature-based), Late Fusion (decision-based), and Hybrid Fusion. The fusion processes were performed without a deep neural network to prevent computational complexity and preserve the interpretability of the final model. The study utilized a number of machine learning( ML) methods such as Logistic regression( LR), Linear Discriminant Analysis( LDA), Decision trees( DT), K-nearest neighbor( KNN) and Support vector machines( SVM). This study demonstrated that the PPG characteristics generated from the wavelet scattering transform in association with a support vector machine can classify normotension and prehypertension with an accuracy of 71.42% and an F1-score of 74.9 %.

Yen et al. [14] proposed a deep learning based approach for hypertension detection using deep residual network convolutional neural network( ResNetCNN) and BILSTM from PPG signal. The study utilized the database created by Liang et al., which includes data collected from 219 participants aged 20–89 years at Guilin People's Hospital in mainland China. They compared the accuracy of two deep learning models for cardiovascular ailment classification, namely ResNetCNN + BILSTM and Xception + BILSTM. When the layers= 37, kernel= 32, and kernel size = 36, the Xception + BILSTM model achieved its best results, with an accuracy of 76 %, a recall of 45%, and a precision of 48%.

Yunendah et al. [15] suggested a diagnosis system for blood pressure level classification using concatenated CNN based on physiological data derived from PPG and ECG signals. They used the MIMIC III dataset. In order to produce one cycle of PPG and ABP signals, the segmentation method was applied based on the R-R peak detection of the ECG signal. To eliminate noise from PPG and ECG signals, a bandpass filter with cutoffs of 0.5–10 Hz and 0.5–40 Hz was used. In this study, PPG and ECG signals were employed as input to a concatenated 1-D CNN. To evaluate the generalization capacity of deep neural networks in classifying unseen data, they utilized 5-fold cross-validation. The optimum structure of the suggested model was determined by comparing five architectures of CNN models based on layer depth. Concatenated CNN models with five convolutional layers produced the most accurate classification of blood pressure values.

Elisa et al. [16] proposed a system to evaluate the applicability of Pulse Rate Variability (PRV) derived features for the classification of BP values using ML algorithms. They used the MIMIC II waeform database. MATLAB® (version 2020a) was used for signal processing. A fourth order lowpass Butterworth filter was used. Time and frequency-domain indices, and non-linear indices extracted from Poincaré plot, entropy, phase, and detrended-fluctuation (DFA) analyses were obtained from PRV data from each segment. After the filtering process, a sequential forward selection (SFS) scheme with k-fold cross-validation (k = 10) was applied to identify the best combination of features for each machine learning algorithm applied. The result showed around 70 % accuracy and around 75 % specificity. The sensitivity, precision and F1 scores were around 50 %.

# Chapter III

# Theoretical Considerations

In this chapter, we review some of the theoretical concepts which helped us to develop the methodology for our proposed system.

## 3.1 Photoplethysmography (PPG) Signal

Photoplethysmography (PPG) is a quick and inexpensive optical method that can be used to detect changes in blood volume in the microvascular bed of tissue. It is also known as pulse oximeter waveform. It is widely used to obtain measurements at the skin's surface non-invasively. Due to their wearable application against traditional ECG technology, PPG signals are growing in popularity for heart rate (HR) monitoring. These signals are obtained through pulse oximeters embedded in a small wearable device worn on the earlobes, fingertips, or at the wrists. Digital signal processing techniques may be used to evaluate PPG and provide new physical characteristics with the extension of PPG's potential benefits. There are several characteristics of PPG signal such as systolic peak, reflected wave peak, dicrotic notch, diastolic peak. These characteristics are important to measure BP and detect hypertension. 3.1 depicts the PPG signal with its characteristic features.

## 3.2 Activation Function

A neuron's activation is controlled by an activation function. During the prediction phase, it performs simpler mathematical operations to decide whether or not the neuron's input to the network is crucial. It describes how a node or nodes in a network layer produce an output from the weighted sum of the input. Different activation functions may be used in various parts of the model, and the choice of activation function has a substantial impact on the capabilities and performance of the neural network.

To increase the nonlinearity of a neural network, an activation function is used. A neural network without activation functions will simply apply a linear transformation to the inputs

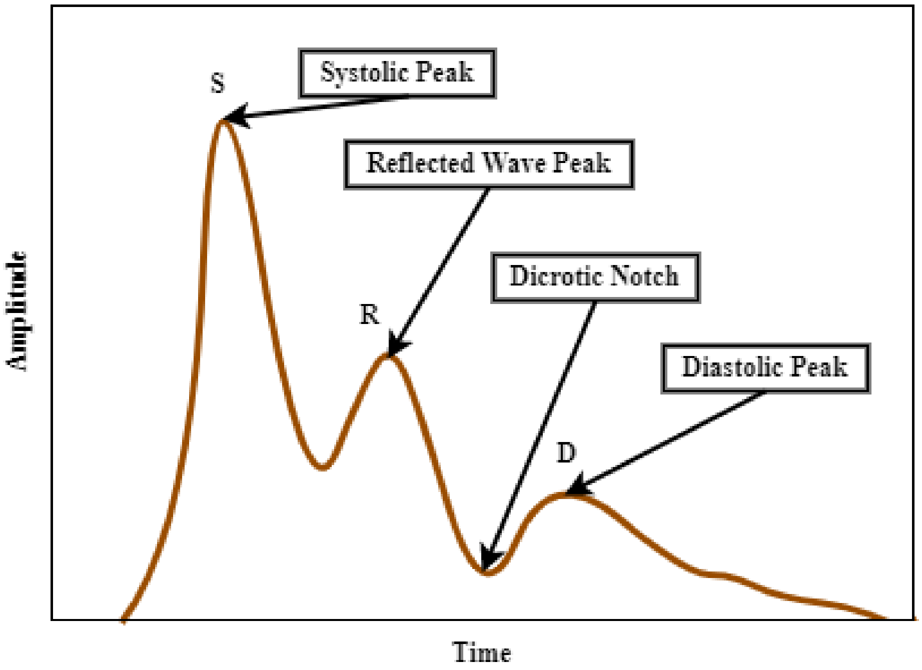


**Figure 3.1:** PPG signal with its characteristic features[1].

using the weights and biases. No matter how many hidden layers we add to the neural network because the combination of two linear functions is a linear function, all layers will behave the same. Therefore, without an activation function, learning any complex job would be impossible.

### 3.2.1 Rectified Linear Unit (ReLU)

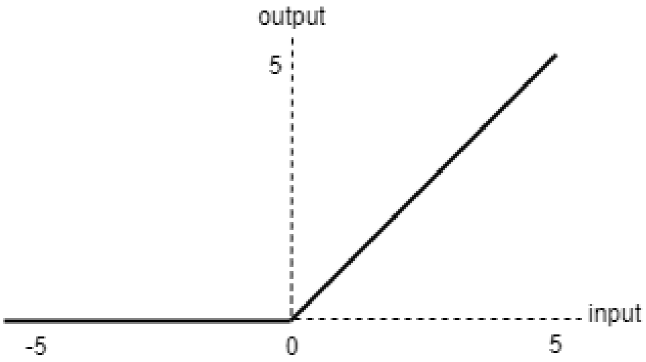


**Figure 3.2:** Rectified linear unit.

The rectified linear activation function, or ReLU activation function is the most common function used for hidden layers. It is one of the most prevalent activation functions in deep learning models. Almost all convolutional neural networks or deep learning models use the ReLU function. The ReLU function takes the maximum value. It is a piecewise linear function that, if the input is positive, outputs the value directly; otherwise, it outputs zero. The ReLU function's equation is given by:

$$f(x) = max(0,x) \tag{3.1}$$

where $x$ denotes the input to neuron. It is a Ramp function.

This is depicted graphically as Fig 3.2

### 3.2.2 Softmax

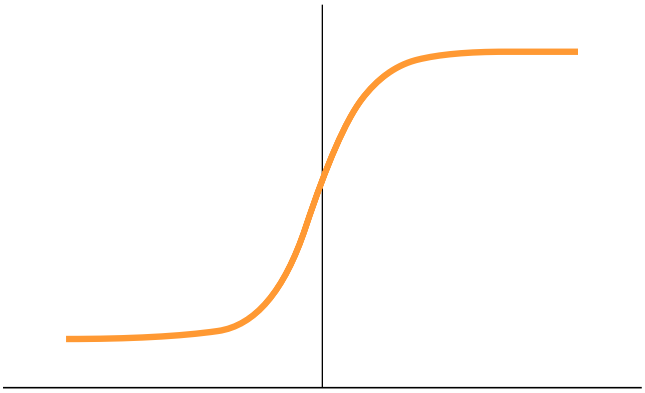


**Figure 3.3:** Softmax.

The neural network's unprocessed outputs are converted into a vector of probabilities—basically, a probability distribution over the input classes—by the softmax activation function. Fig. 3.3 shows softmax activation curve. The follwing is the equation of softmax activation function:

$$softmax(z)_i = \frac{e^{zi}}{\sum_{j=1}^{N} e^{zj}} \tag{3.2}$$

Where z is the vector of raw outputs from the neural network. The $i^{th}$ entry in the softmax output vector softmax(z) can be thought of as the predicted probability of the test input belonging to class i.

## 3.3 Adam Optimizer

Adaptive Moment Estimation is a technique for optimizing gradient descent algorithms.

### Momentum

By taking into account the ‘exponentially weighted average’ of the gradients, this approach is utilized to speed up the gradient descent algorithm. Using averages accelerates the algorithm’s convergence to the minima.

$$w_{t+1} = w_t - \alpha_t m_t \tag{3.3}$$

where,

$$m_t = \beta m_{t-1} + (1-\beta)[\frac{\partial L}{\partial w_t}] \tag{3.4}$$

Here, $m_t$ denotes the aggregate of gradients at time $t$ (initially, $m_t$ = 0), $m_{t-1}$ denotes the aggregate of gradients at time $t-1$, $w_t$ and $w_{t+1}$ refer to the weights at time $t$ and $t+1$ respectively, $\alpha_t$ is the learning rate at time $t$, $\partial L$ and $\partial w_t$ refer to the partial derivative of loss function and weights at time $t$ and $\beta$ is the moving average parameter.

### Mathematical Model of Adam Optimizer

Using the formulas of the above methods we get, $m_t = \beta_1 m_{t-1} + (1-\beta_1)[\frac{\partial L}{\partial w_t}]$ and $v_t = \beta_2 v_{t-1} + (1-\beta_2)[\frac{\partial L}{\partial w_t}]^2$. Because $m_t$ and $v_t$ were both initialized as 0 (based on the previous approaches), it is noted that as both $\beta_1$ and $\beta_2 \approx 1$, they become ‘biased towards 0’. This Optimizer solves the problem by computing $m_t$ and $v_t$ that are ‘bias-corrected.’ This is also done to keep the weights under control as they approach the global minimum, preventing high oscillations when they get too close. The following are the formulas that were used:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t} \tag{3.5}$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \tag{3.6}$$

We are, intuitively, adapting to the gradient descent after each iteration so that the process remains controlled and unbiased, hence the name Adam.

We will now use the bias-corrected weight parameters $\widehat{m_t}$ and $\widehat{v_t}$ instead of our standard weight parameters $m_t$ and $v_t$. When we plug them into our general formula, we obtain:

$$w_{t+1} = w_t - \widehat{m_t}\left(\frac{\alpha}{\sqrt{\widehat{v_t}} + \varepsilon}\right) \tag{3.7}$$

## 3.4 Loss Function

A loss function analyzes how effectively the neural network models the training data by comparing the target and forecasted output values. Loss function optimize parameters for the neural network. When training, we aim to minimize this loss between the predicted and target outputs. We used the categorical cross-entropy loss function here.

### 3.4.1 Categorical Cross-entropy

The cross entropy loss function is an optimal function that is used while training a classification model that classifies data by determining whether the data belongs to one class or the other. Categorical cross-entropy is suitable for multiple classification task. It is used as a loss function in multi-class classification models with two or more output labels. In this function the class number and the output node will be equal. Final value of the layer is forwarded to a softmax activation function that will give a probability between [0-1]. A perfect model has a cross-entropy loss of 0. The following formula computes the categorical cross-entropy:

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i) \tag{3.8}$$

Where $t_i$ is prediction and $p_i$ is the softmax probability for the $i^{th}$ class.

## 3.5 Learning Rate

The learning rate is a hyperparameter that specifies how much the model should change in response to the predicted error each time the model weights are updated. Choosing the learning rate is difficult since a value too low may result in a prolonged training process that might get stuck, but a value too high may result in learning a suboptimal set of weights too quickly or an unstable training process. It is a small positive value, often in the range between

0.0 and 1.0. The learning rate determines how fast the model adapts to the problem. Smaller learning rates need more training epochs because to the smaller changes made to the weights with each update, whereas larger learning rates result in quick changes and necessitate fewer training epochs.

## 3.6 Stratified KFold Cross Validation

A variation on standard kfold cross validation, stratified kfold cross validation is used for classification issues where, as opposed to entirely random splits, the ratio of target classes is the same in each fold as it is across the entire dataset. When our data are unbalanced and on the small side in terms of size, stratified kfold cross validation is often helpful. In order to address class imbalance, we will occasionally over- or under-sample our data. However, there are other occasions when we wish to keep the class imbalance because it is indicative of or provides information about the phenomenon we are trying to classify. The dataset is divided into 10 parts: 9 of them are utilized for training and the rest is utilized for testing.

## 3.7 CNN

A convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyze visual imagery. The role of the CNN is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction. CNN is identical to the classic ANN. It consists of neurons that optimize themselves through learning. Each neuron receives an input and carries out an operation. The fundamental concept underlying the operation of Neural Networks has always been to emulate the functioning of the human brain to the greatest extent possible. CNN assists to this by working with the visual sensory organs of live organisms and, in the process, recognizing various forms of object, be it Digit, Image, or a particular activity in any object, by employing a series of approaches in a particular order. CNN is typically composed of three types of layers (or building blocks): convolution, pooling, and fully connected layers as shown in Fig. 3.4. The first two, convolution and pooling layers, perform feature extraction, whereas the third, a fully connected layer, maps the extracted features into final output, such as classification. As one layer feeds its output into the next layer, extracted features can hierarchically

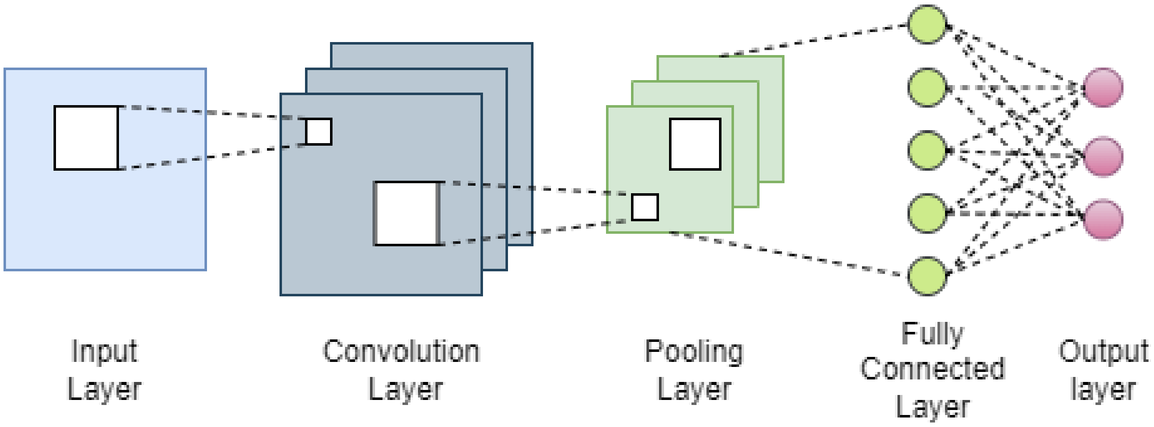


**Figure 3.4:** CNN basic architecture.

and progressively become more complex. These layers have distinct importance during the learning phase. A brief description of these layers follows.

### 3.7.1 Convolution

The objective of convolution layers is feature extraction from images. Convolution involves sliding the kernel over the input signal which is also known as shift-compute procedure. By shifting the filters along the input and computing the dot product of the weights and input, followed by the addition of a bias term, the layer convolutionally transforms the input. 1-D convolution uses two signals. The convolution takes input vector f and kernel g, and say where f has length n, and g has length m. The convolution f  g of f and g is defined as:

$$(f*g)(i) = \sum_{j=1}^{m} g(j) \cdot f(i-j+m/2) \tag{3.9}$$

The size of the output vector is the same as the size of the input.

### 3.7.2 Pooling

Pooling conducts downsampling by decreasing the size and transmits only the essential data to subsequent CNN layers. It is a sample-based discretization procedure. The primary objective of pooling is to minimize the size of feature maps, which as a result accelerates computation by reducing the number of training parameters. Pooling types include maximum pooling and average pooling. Max pooling pools or takes the maximum value while average pooling takes the region's average value. With the preceding activation functions, convolution and max pooling are implemented to the hidden layer to introduce nonlinearity. There are two primary purposes, namely feature extraction and categorization.

### 3.7.3 Feature Extraction

Convolution layer identifies the significant features. Additional features are obtained when more layers are added. The layers in close proximity to the input layer identify the features. The deeper layer combines the simple features and generates complicated features that are difficult for humans to comprehend, making categorization easier. Utilizing a filter, an output feature map is created.

## 3.8 LSTM

Long short term memory (LSTM) is a special kind of RNN, which is designed to solve the gradient disappearance and gradient explosion problems during the training of long sequences. In simple terms, it means that LSTM can perform better on longer sequences than normal RNN. The first step in the LSTM is to decide what information we're going to throw away from the cell state. The next step is to decide what new information we're going to store in the cell state. Then the old cell state is updated. Finally, we need to decide what we're going to output. Fig. 3.5 shows the basic structure of LSTM. Four neural networks and several memory cells make up the chain structure of the LSTM. The cells store information, while the gates perform memory alterations. There are three gates, forget gate, input gate and output gate. The input gate determines what data is added to the memory cell. The forget gate determines what data is deleted from the memory cell. And the output gate determines what data is output from the memory cell.

- **Forget gate**

  Two inputs, $x_t$ (at-the-time input) and $h_{t-1}$ (previous cell output), are supplied into the gate and multiplied using weight matrices before bias is added. The result is transmitted to an activation function, which produces a binary output. If the output for a certain cell state is 0, the information is lost; if the output is 1, the information is saved for future use.

- **Input gate**

  The information is monitored using the sigmoid function, and the values to be remembered are filtered using the $h_{t-1}$ and $x_t$ inputs, similar to the forget gate. The tanh function is then used to generate a vector with values ranging from -1 to +1 that encompasses all of the possible values from $h_{t-1}$ and $x_t$. Finally, the vector and regulated values are multiplied to provide relevant information.

- **Output gate**

  By using the tanh function on the cell, a vector is first created. Following that, the data is controlled by the sigmoid function and filtered by the values that need to be remembered using the inputs h t-1 and x t. In order to provide the values as an output and an input to the following cell, the vector's values and the controlled values are finally multiplied.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{3.10}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{3.11}$$

$$o_i = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{3.12}$$

$$\tilde{C_t} = tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \tag{3.13}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C_t} \tag{3.14}$$

$$h_t = o_t \odot tanh(C_t) \tag{3.15}$$

Equation 3.23 – 3.25 defines the forget, input and output gates of each LSTM block. The block input, which consists of a *tanh* layer and an input gate is specified as $\tilde{C_t}$ in equation 3.26 at time t. They collaborate to select what data will be kept in the cell state, $C_t$. At time t, the cell state is updated from the previous cell state. W and U are weight matrices and b is a bias vector. Finally, the hidden state $h_t$ is block output at time $t$.

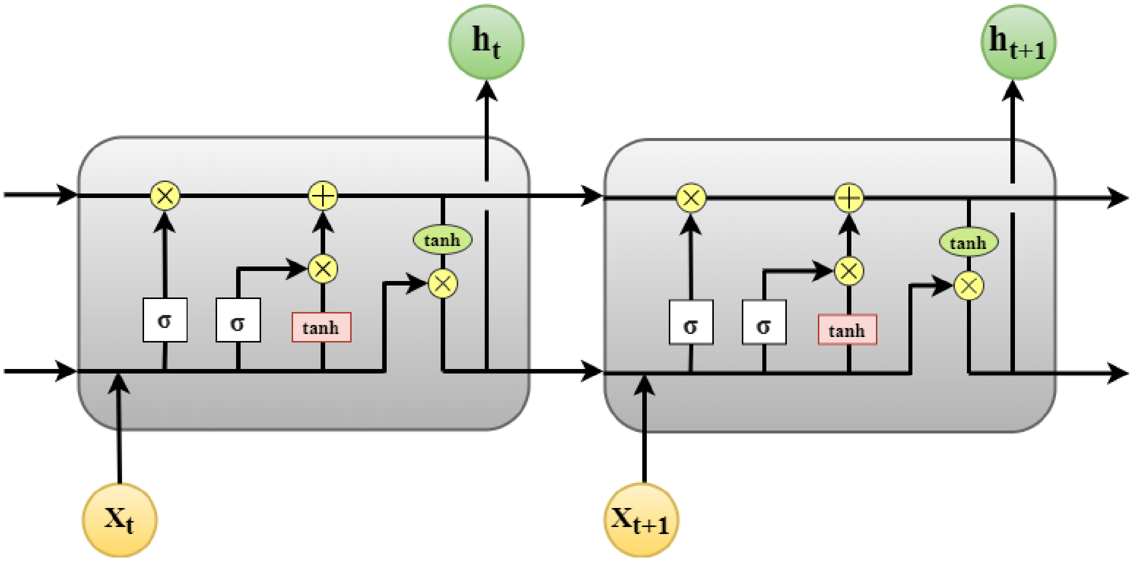


**Figure 3.5:** LSTM basic architecture.

### 3.8.1 Activation Function

Activation function in neural network describes how a node or nodes in a layer of the network convert the weighted sum of the input into an output. In LSTM blocks, the two most popular activation functions, sigmoidal and hyperbolic tangent are used. The sigmoid function has a range of [0, 1]. The formula is given by:

$$\sigma(x) = \frac{1}{e^{-x} - 1} \tag{3.16}$$

The hyperbolic tangent formula referred to as the hyperbolic function is given by:

$$tanh(x) = \frac{sinh(x)}{cosh(x)} \tag{3.17}$$

## 3.9 GRU

Gated Recurrent Unit (GRU) is a type of recurrent neural network. It resembles an LSTM. With the benefit of being faster to calculate, the gated recurrent unit (GRU) provided a simplified version of the LSTM memory cell that frequently achieves equivalent performance. GRUs are typically simpler/faster to train than their LSTM counterparts because of the architecture's simplicity. It also has three gates: an update gate, a reset gate and current memory gate. Fig 3.6 specifies the architecture of GRU. The different gates of a GRU are as described below:

- **Update Gate**

  It determines the amount of past data that must be sent into the future. It is comparable to the Output Gate in an LSTM recurrent unit. In equ 3.18, $W_z$ and $U_z$ are the parameter matrices and $b_z$ is the bias vector.

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{3.18}$$

- **Reset Gate**

  It specifies how much of the past information should be forgotten. It is comparable to how the Input Gate and Forget Gate work together in an LSTM recurrent unit. In equ 3.19, $W_r$ and $U_r$ are the parameter matrices and $b_r$ is the bias vector.

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{3.19}$$

- **Current Memory Gate**

  Current Memory Gate is incorporated into the Reset Gate. It is used to introduce some non-linearity into the input and to also make the input Zero-mean.

## 3.10 Time Distributed Layer

Time Distributed Layer is one type of wrapper used for time data or vedio frames. Every temporal slice of an input can have a layer applied to it using this wrapper. Each input must have a minimum of three dimensions, with the temporal dimension being the index-one dimension of the first input. For example, if we apply this around a Conv2D layer, the identical set of weights are applied at each timestamp since TimeDistributed uses the same instance of Conv2D for all of the timestamps.

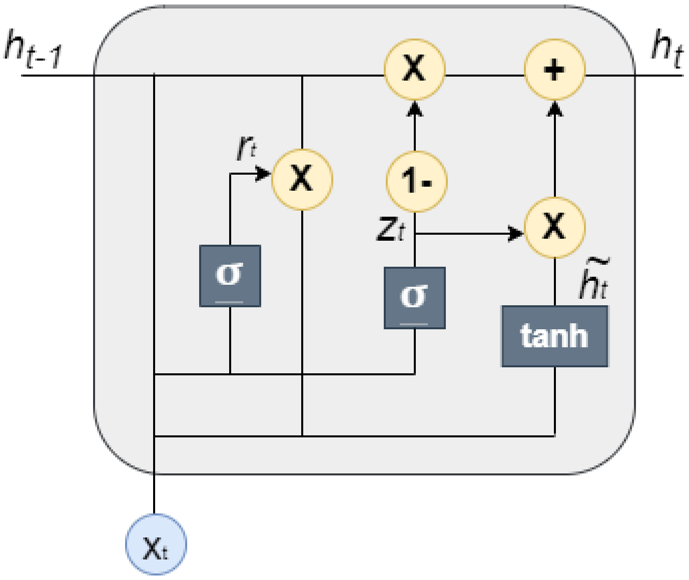


**Figure 3.6:** GRU basic architecture.

## 3.11 Dense Layer

A layer utilized in the last phases of the neural network is referred to as a dense layer, also known as a fully connected layer. This layer contains densely connected neurons. Each of the individual neurons of the layer takes the input data from all the other neurons before a currently existing one. Internally, the dense layer is where various multiplication of matrix vectors is carried out. This layer assists in changing the dimensionality of the output from the preceding layer , allowing the model to more clearly identify the relationship between the values of the data it is working with. This is why the dense layer is most often used for vector manipulation to change the dimensions of the vectors. The vector's translation, scaling, and rotation may all be accomplished by the dense layer.

## 3.12 Flatten Layer

Flatten layer performs the process of transforming data into a 1-dimensional array for inputting it into the following layer. The output of the convolution layers is flattened to form a single long feature vector. It is also linked to the final classification model, which is known as a fully-connected layer or dense layer. It collapses the spatial dimensions of the input into the channel dimension. Flatten layer flattens the input and does not affect the batch size.

## 3.13 Dropout Layer

Dropout layer prevents the problem of overfitting in the training data. If they aren't present, the first batch of training samples influences the learning in a disproportionately high manner. This in turn, would prevent the learning of features that appear only in later samples or batches. Dropout contributes to a reduction in overfitting by decreasing the squared norm of the weights. In drop out, nodes are dropped randomly and thus the model can learn independent representations. Overall, a better performance is achieved by applying drop out.

# Chapter IV

# Proposed Method

In this chapter, the overall methodology of the proposed system is depicted briefly. The proposed system is designed to take PPG signals as input and classify Hypertension that is non-invasive and faster approach. This chapter is comprised of 6 sections: section 4.1 depicts the overall system architecture, section 4.3 illustrates the data preprocessing, section 4.5 depicts the first hybrid model and section 4.6 describes the second hybrid model and section 4.7 defines the model construction method and techniques.

## 4.1 System Architecture

In this section, the overall system architecture of the proposed system containing two Deep Learning models is depicted in Figure. 4.1. The system is composed of two models - the 1st model is a hydrib model CNN-LSTM and the 2nd model is a hybrid of CNN-GRU. Both models are Deep Learning models with several neurons and hidden layers. The system takes PPG signal as input from patients and classify 4 state of Hypertension: i) Normal, ii) Prehypertension , iii) Stage 1 hypertension and iv) Stage 2 hypertension.

## 4.2 Dataset

We used dataset from Liang et al. [17] for this study. This dataset includes 219 subjects' 657 PPG signal samples [18]. The PPG signal has a signal duration of 2.1 s, 2100 data points per signal, and is sampled at a rate of 1000 Hz. A raw PPG signal is shown in Fig. 4.2. There is also the patient's demographic information such as age, gender, height, and weight along with systolic pressure, diastolic pressure, and heart rate. Table 1 gives an overview of the demographic dataset.

## 4.3 Data Pre-Processing

The raw PPG signals are preprocessed by following methods:

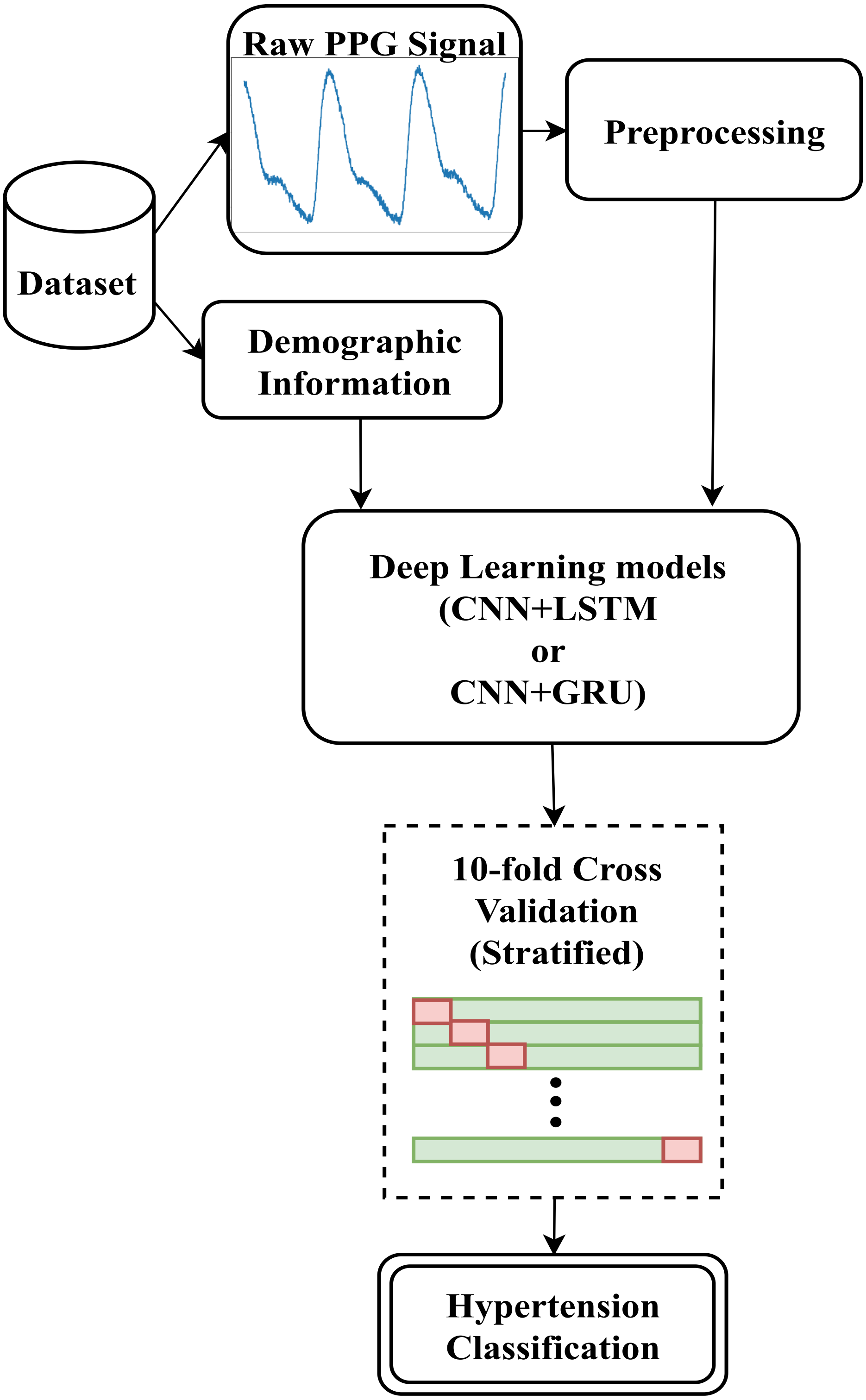

Raw PPG Signal
Preprocessing
Dataset
Demographic Information
Deep Learning models (CNN+LSTM or CNN+GRU)
10-fold Cross Validation (Stratified)
Hypertension Classification

**Figure 4.1:** Proposed system architecture

**Table 4.1:** Patient's demographic information.

| **Physical Index** | **Numerical Data** |
|---|---|
| Females | 115(52%) |
| Age( years) | 57 ± 15 |
| Height( cm) | 161 ± 8 |
| Weight(kg) | 60 ± 11 |
| Hypertension | 0-4 |

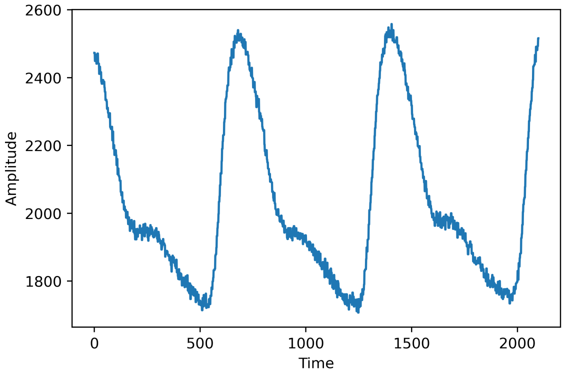


**Figure 4.2:** Raw PPG signal.

### Data Scaling

Z-score normalization enables a data administrator to comprehend the likelihood that a score will occur within the data's normal distribution. Here we use z-score normalization to get amplitude-limited data. The equation is given below-

$$Zscore\ Normalized\ Signal = \frac{Signal - Signal\ Mean}{Standered\ Deviation\ of\ Signal} \tag{4.1}$$

### Signal Filtration

We first filtered the waveforms because the original dataset had noise, baseline drift, and strange waveforms. A Butterworth low pass filter with a cutoff frequency of 15 Hz is used in this study to filter the data. The high frequency component of the signals is eliminated by the low pass filtering. Thus reduces noise in the signal. Fig. 4.3 shows the low pass filtered signal(orange line) over the normalized raw signal(blue line).

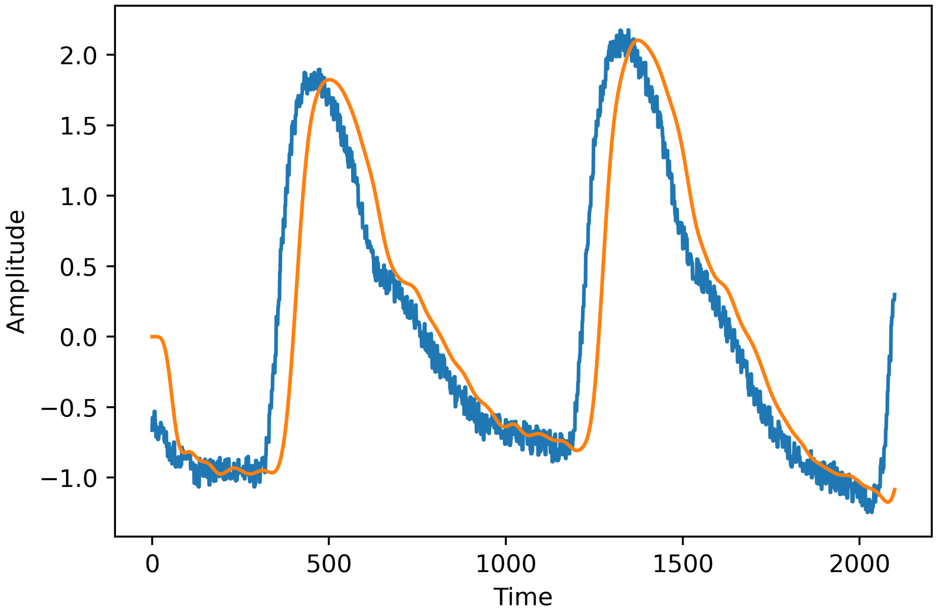


**Figure 4.3:** Blue line shows normalized signal, orange line shows signal after butterworth low pass filter.

## 4.4 Testing and Training

We split the dataset into 80% training and 20% testing based on similar ratio of levels using stratify. We use this training set for further train and validation and then calculate prediction accuracy by this testing dataset.

## 4.5 Proposed Model: 1

This proposed deep learning framework, uses the CNN and LSTM technique. All sorts of different classifications can be spit out of this network. The onset of hypertension is anticipated. Model 1's topology is depicted in Fig. 4.4; it was developed by stacking the CNN, BatchNormalization, LSTM, and fully connected layer. As a feature extractor, CNN uses two 1D convolutional layers that are connected by ReLU activation, batchNormalization, max-pooling, and drop-out layers. In order to forecast physiological parameters, the LSTM model processes the characteristics extracted from the preceding layer's output and feeds them to 3 fully connected layer through Flatten. One LSTM layers with a relu activation and dropout layer, make up the LSTM model. Below, we'll go over the specifics of the suggested model's architecture:

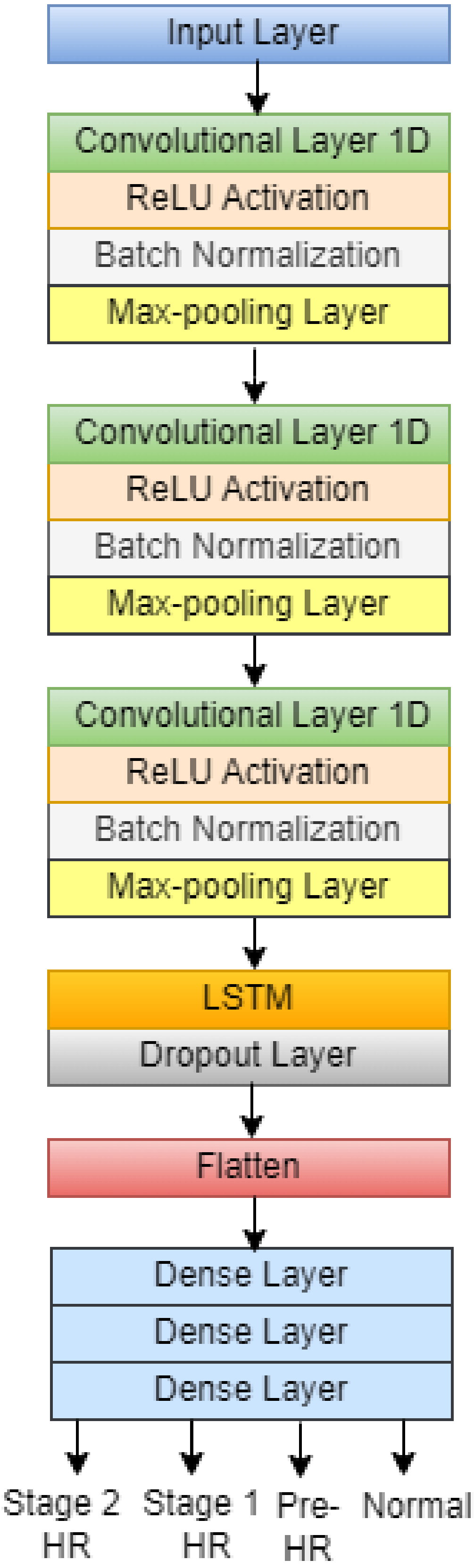


**Figure 4.4:** Model 1's topology.

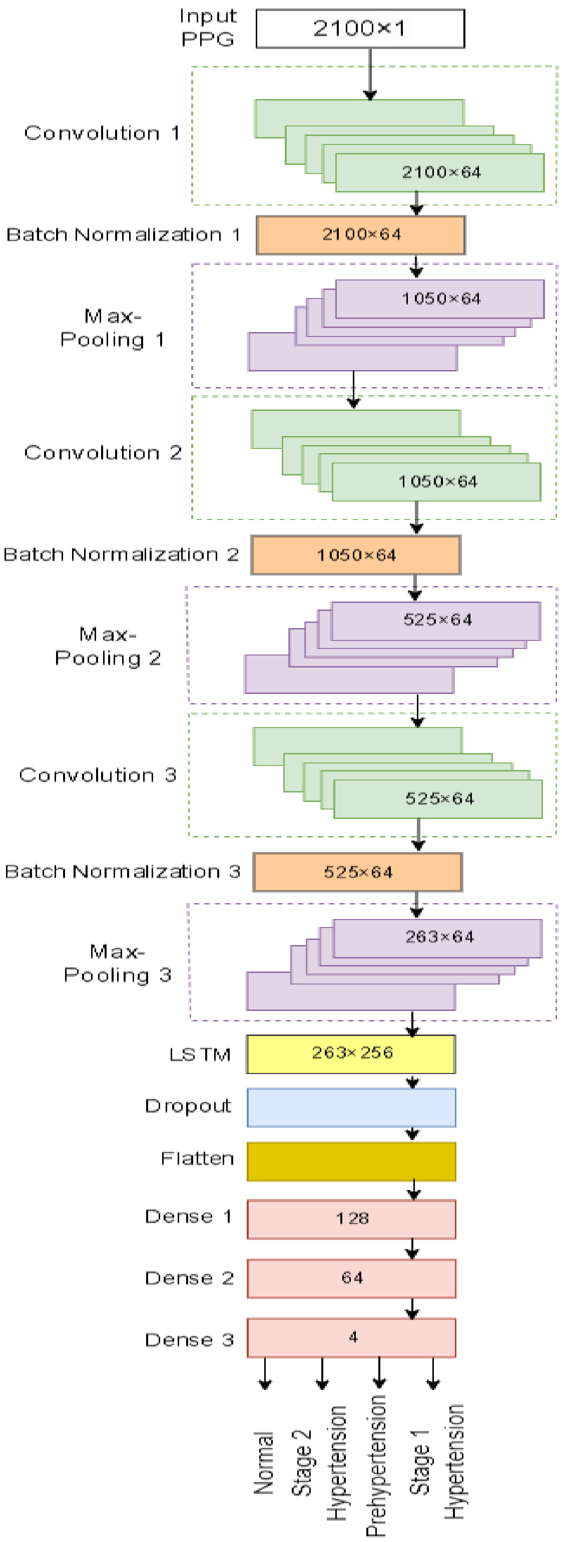


**Figure 4.5:** Architecture of the proposed Model 1.

The hybrid architecture of the proposed Model 1 is depicted in Fig. 4.5. It is made up of three 1D convolutional layers with Batchnormalization. Each of the convolutional layer have 64 filters with kernel size 3x1. ReLU activation function is used. This results in the 64 activation maps, which at each geographic position where each feature map captures a separate low-level feature provide the responses of associated filters. Then, using the max operation, pooling is implemented along the spatial dimensions with size 2. The size of the space is gradually decreased by this procedure. Next, one LSTM layer of 256 memory cell is united with CNN model. A flatten layer is also added. Softmax is used to get the final classification scores, where 4 classes are predicted by each of the 3 fully linked layers' 128, 64, and 4 output neurons.

## 4.6 Proposed Model: 2

This proposed deep learning framework, incorporates TimeDistributed CNN-GRU. The network architecture is also intended to support multiclass output and anticipate hypertension. Figure. 4.6 depicts the Model 2's topology, which was created by stacking the TimeDistributed CNN, GRU, and fully connected layer. CNN is also used as a feature extractor in this case but with timeDistributed wrapper and two 1D convolutional layers interleaved with ReLU activation, maxpooling. The previous layer's output features are passed through Flatten then the GRU model and then fed to 3 fully connected layer to predict physiological parameters. The GRU model is made up with relu activation and a dropout layer. The proposed model's architectural details are discussed further below:

The hybrid architecture of the proposed Model 2 is depicted in Fig. 4.7. It is made up of two 1D convolutional layers with TimeDistribution. 32 and 64 filters were used in these convolutional layer with kernel size 3x1. ReLU activation function is used. This results in the 64 activation maps, which at each geographic position where each feature map captures a separate low-level feature provide the responses of associated filters. Then, using the max operation, pooling is implemented along the spatial dimensions with size 2. The size of the space is gradually decreased by this procedure. Next, one GRU layer of 256 memory cell is united with CNN model. Softmax is used to get the final classification scores, where 4 classes are predicted by each of the 3 fully linked layers' 128, 64, and 4 output neurons.

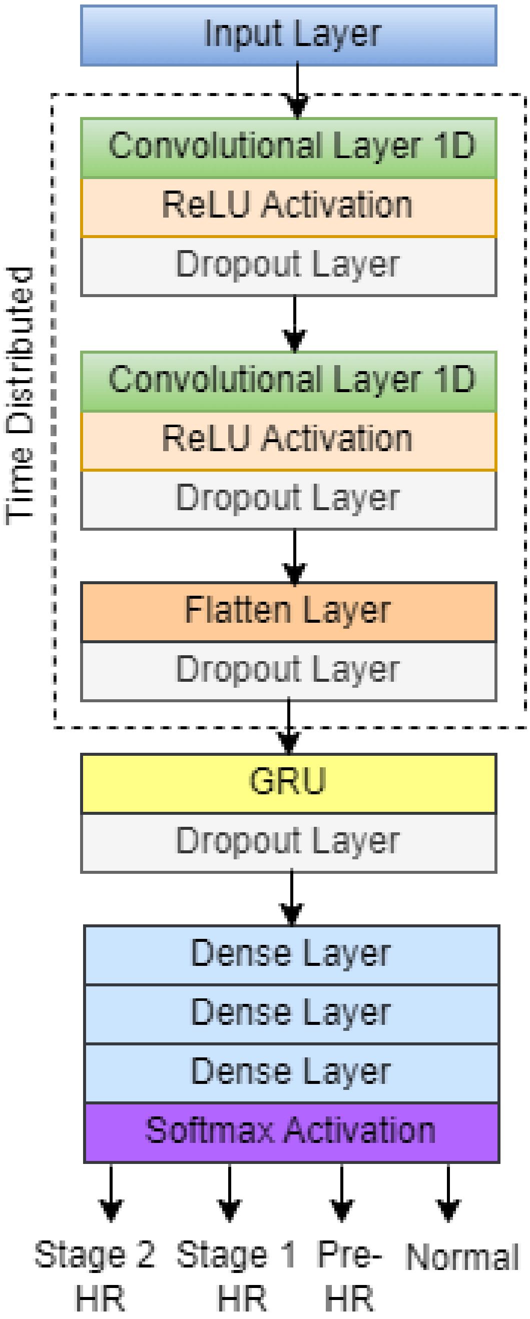


**Figure 4.6:** Model 2's topology.

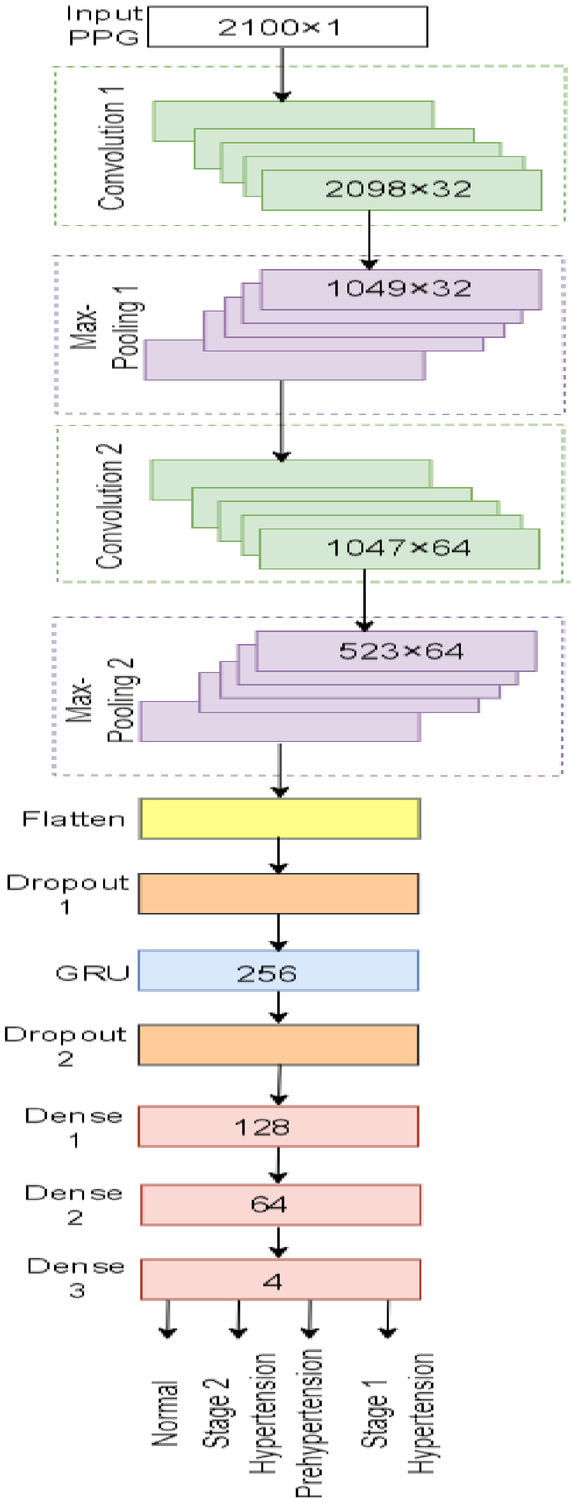


**Figure 4.7:** Architecture of the proposed Model 1.

## 4.7 Training and Evaluation

For training both model, 120 epochs with batch size of 32 are used. But the 2nd model is given an early stop with patience 60. The optimization during the training is performed by Adam optimizer with learning rate 0.001 and Categorical crossentropy are used as the loss function to evaluate the performance of proposed framework. The models are validated using the Stratified KFold cross-validation approach owing to the fact that it always provides less optimistic and less biased estimation compared to simple train/test split method and also better classification than Kfold as it ensures that each fold of dataset has the same proportion of observations with a given label. For this experiment, k is set to 10 and we split the training data for training and validation. The performance evaluation for the models are done by averaging the model scores of all the k testing sets wherein accuracy, precision and recall are considered as effective metrics for classification tasks.

# Chapter V

# Result Analysis and Discussion

## 5.1 Experimental Setup

All the experiments in this proposed system are performed on Colab cloud platform. Machine configuration of colab platform includes - Nvidia K80 / T4 GPU having 0.82GHz / 1.59GHz GPU Memory Clock and 12GB / 16GB Memory, 4.1 TFLOPS / 8.1 TFLOPS Performance, 10.5 / 12.7 GB RAM and 25.8 / 107.7 GB disk space.

## 5.2 Experimental Result

1st model's loss and accuracy of the training set along with validation are depicted in Fig.5.1 and 5.2

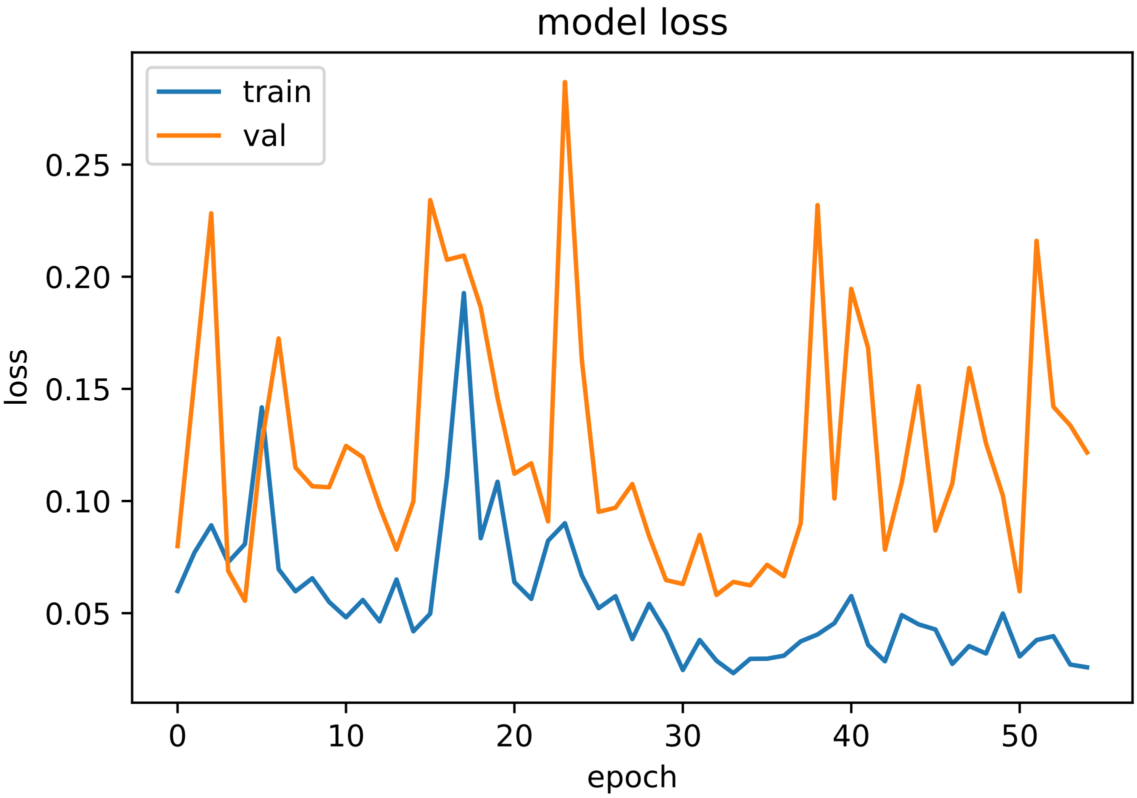


**Figure 5.1:** Loss and validation loss for Model 1 (best fold)

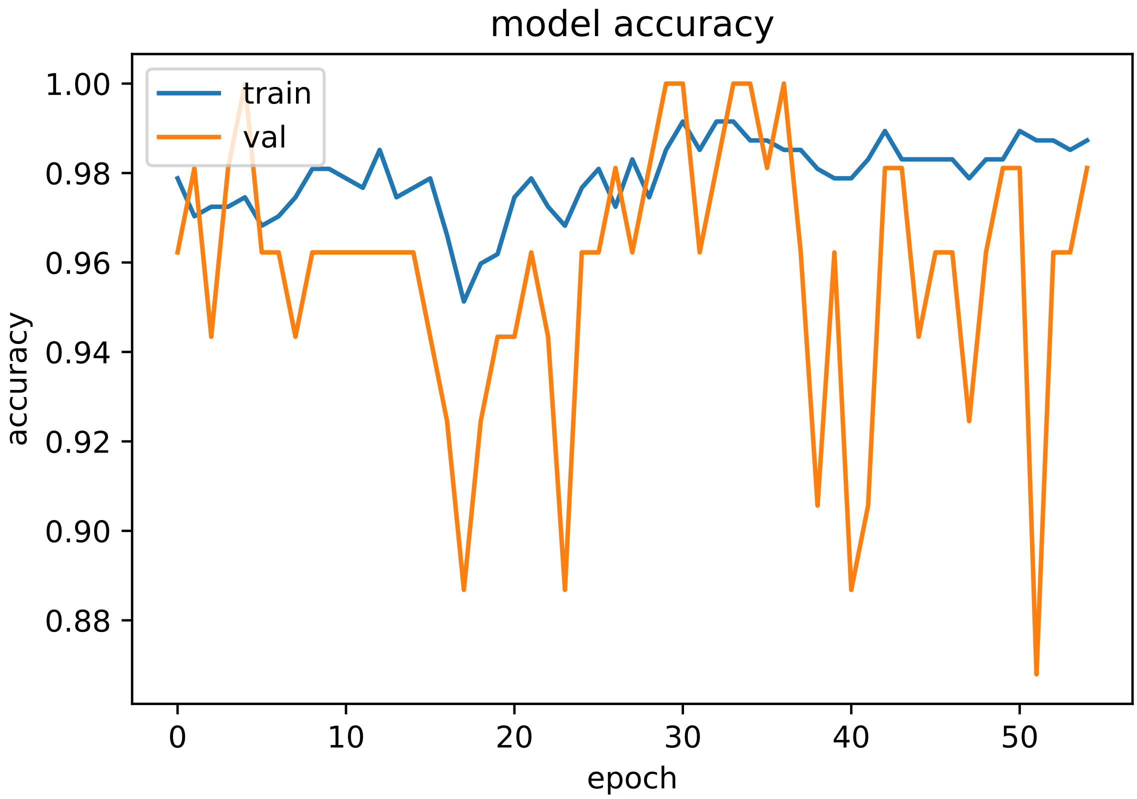


**Figure 5.2:** Accuracy and validation accuracy for Model 1 (best fold)

As it can be seen from Fig.5.1 and Fig. 5.2, the training and validation loss remain within 0.05 and 0.25 and the training and validation accuracy remain within 0.85 and 0.99 over 120 epochs.

2nd model's loss and accuracy of the training set along with validation are depicted in Fig.5.3 and 5.4

As it can be seen from Fig.5.4 and Fig. 5.3, the training and validation loss remain within 0.05 and 0.30 and the training and validation accuracy remain within 0.92 and 0.99 over 120 epochs.

## 5.3 Evaluation Matrices

- **Confusion Matrix**

  Confusion Matrix is a summary table of predicted-labels vs. actual-labels. The amount of accurate and inaccurate predictions is summarized using count values and categorized by class.

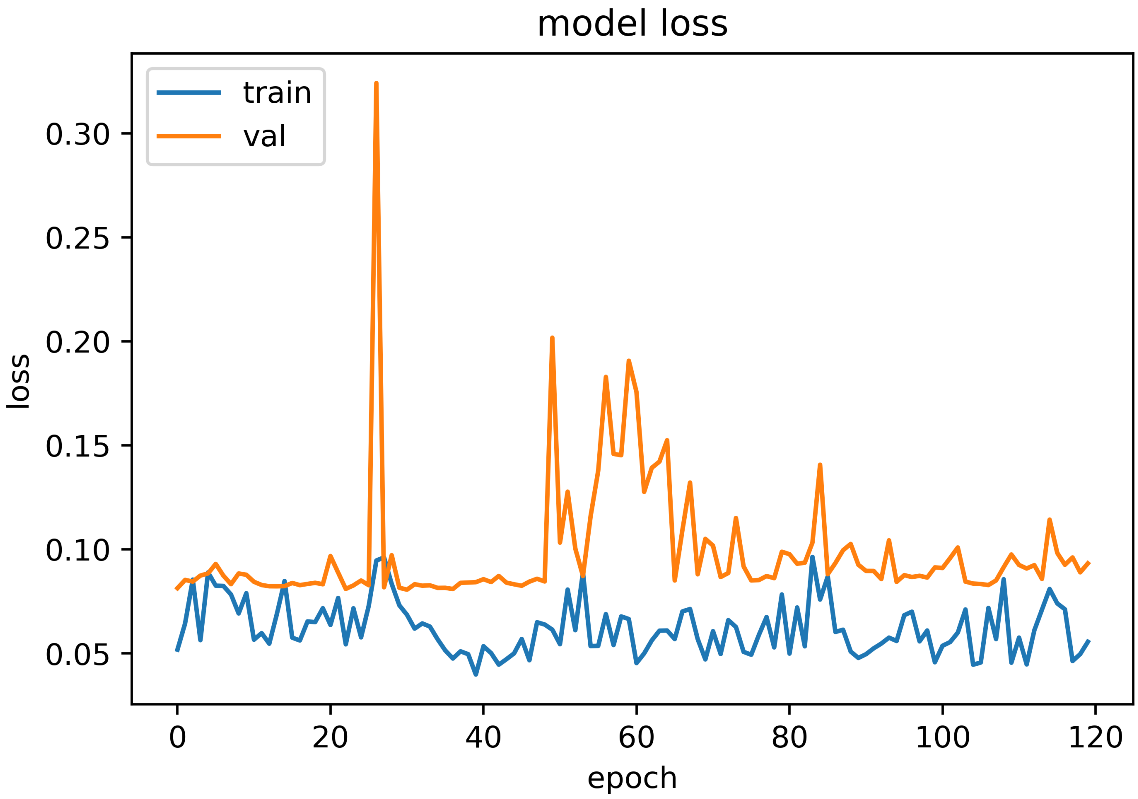


**Figure 5.3:** Loss and validation loss for Model 2 (best fold)

**True Positive (TP)**

A test result that accurately identifies the presence of a condition or feature.

**True Negative (TN)**

A test result that accurately identifies the absence of a condition or feature.

**False Positive (FP)**

A test result that inaccurately implies the presence of a particular condition or feature.

**False Negative (FN)**

A test result that inaccurately implies the absence of a particular condition or feature.

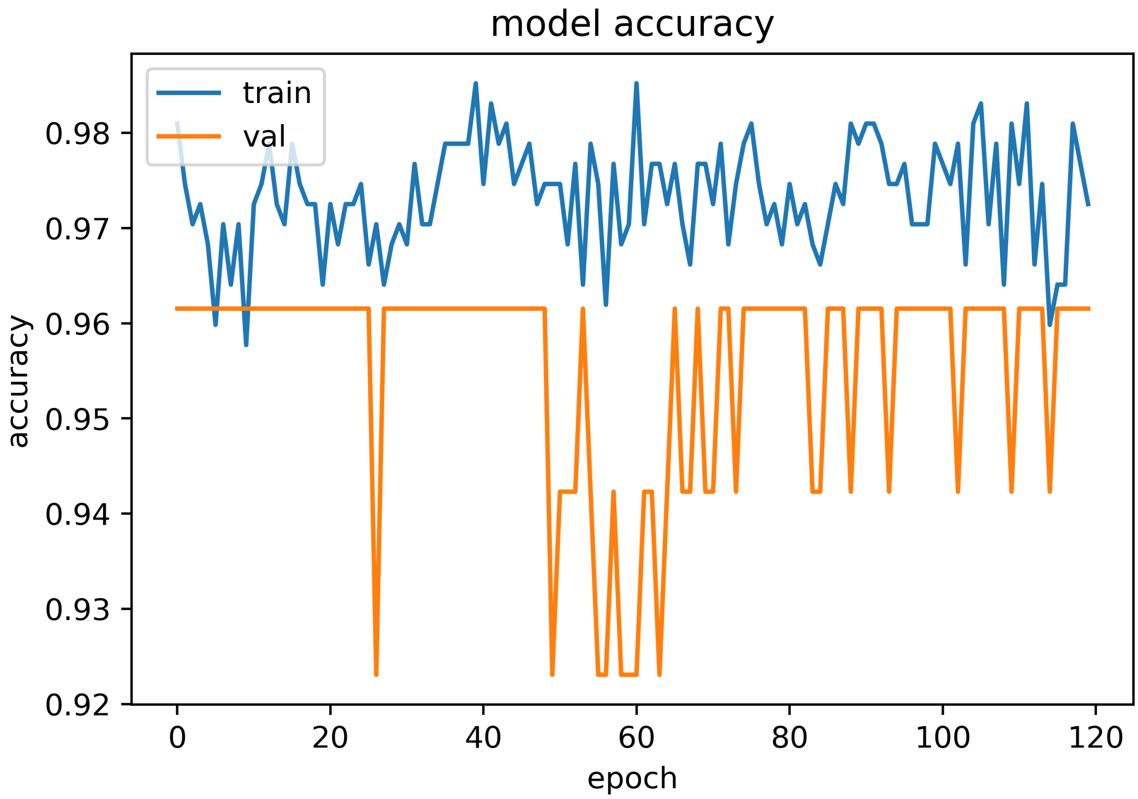


**Figure 5.4:** Accuracy and validation accuracy for Model 2 (best fold)

- **Accuracy**

  Accuracy represents the measurement of correctly predicted data over the total data. It describes the performance of the model across all classes. It is advantageous when all classes have the equal importance. An accuracy rate is determined by dividing the number of accurate predictions by the total number of predictions.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{5.1}$$

- **Precision**

  Precision is the ratio of correct positive predictions over the total predicted positives. It is a statistical measure of how well a model can identify positive data from a training set. Classifying all positive samples as positive and no negative samples as positive is what precision aims towards.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

- **Recall**

Recall is the ratio of correct positive predictions over the total positives. It is also known as sensitivity. The recall statistic evaluates how well a model can identify positive data. The higher the recall, the more positive samples detected. The recall is only concerned with the classification of positive samples. This is independent of the classification of the negative samples.

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

- **F1 score**

F1-Score is calculated by combining Precision and Recall. It was designed to consolidate the effectiveness of the accuracy and recall metrics. The F1 score gives equal weight to Precision and Recall. It is a proposed improvement of two simpler performance metrics.

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5.4}$$

- **Receiver Operating Characteristic curve (ROC curve)**

An ROC curve is a graph that demonstrates the performance of a classification model across all classification levels. It plots True Positive Rate (TPR) vs. False Positive Rate (FPR) at different classification thresholds. AUC stands for "Area under the ROC Curve." AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). The more closer the AUC is to 1, the better performance the system will give. It is the probability that the model rates an arbitrary positive example higher than an arbitrary negative example.

$$TRP = \frac{TP}{TP + FN} \tag{5.5}$$

$$FRP = \frac{FP}{TN + FP} \tag{5.6}$$

### Classification Result

Accuracy of the proposed system models with 20% test data are given in Table. 5.1.

**Table 5.1:** Performance measurement of 2 proposed models

| Models | Average Performance Measure Criteria(Accuracy) | | |
|---|---|---|---|
| | Training [best fold](%) | Validation [best fold](%) | Testing (%) |
| 3CNN + 1LSTM | 98.00 | 93.00 | 81.06 |
| 2CNN(TimeDistributed) + 1GRU | 97.50 | 95.00 | 85.00 |

As illustrated in Table. 5.1, Average Training, Validation, and Testing are predicted with values of 0.98 0.93 and 0.8106 for Model 1. For Model 2, the values are 0.9750, 0.95 and 0.85. The average training and validation accuracy are from best fold among 10 stratified fold.

Fig. 5.5 and 5.6 shows Confusion Matrix and ROC of all 10 fold for Model 1. Confusion Matrix is calculated for 4 class. A average of all confusion matrix are predicted in Fig. 5.7. The ROC's of all fold contains all classes ROC curve and their 'macro' and 'micro' average curves. As 'micro' curve gives better value, we showed all 'micro' ROC curve of 10 folds in Fig. 5.8.

Fig. 5.9 and 5.10 shows Confusion Matrix and ROC of all 10 fold for Model 1. Confusion Matrix is calculated for 4 class. A average of all confusion matrix are predicted in Fig. 5.11. The ROC's of all fold contains all classes ROC curve and their 'macro' and 'micro' average curves. As 'micro' curve gives better value, we showed all 'micro' ROC curve of 10 folds in Fig. 5.12.

Table 5.2 shows the comparative analysis of our proposed models with the previous methods. Our 1st Model obtain accuracy, precision, recall and f1-score of 81.06%, 84.24%, 86.80%

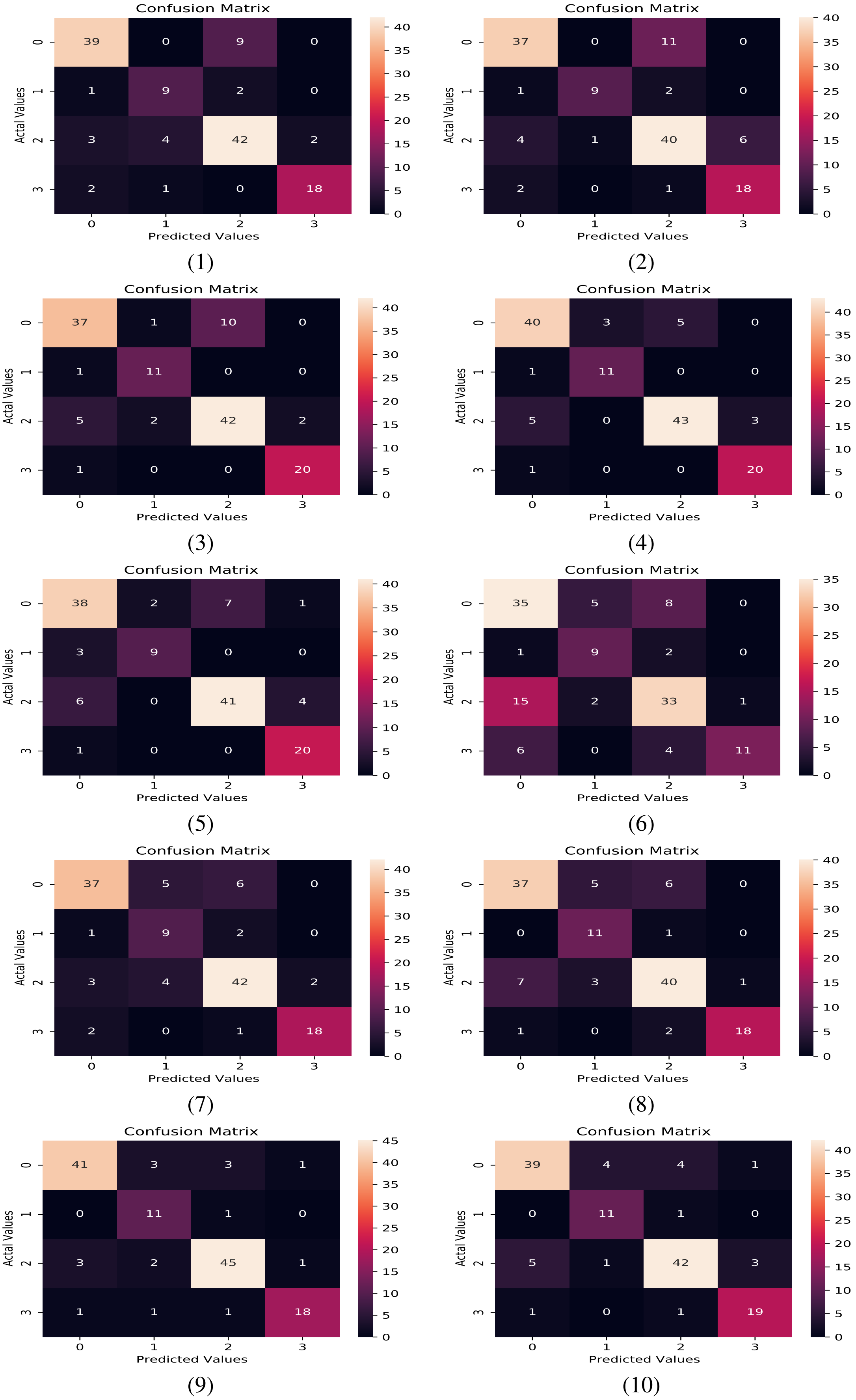
**Table 5.2:** Comparative Analysis With Previous Studies

| **Methods** | | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-score (%)** |
|---|---|---|---|---|---|
| SVM<br>(19 WST Features) [13] | | 71.42 | 65.51 | **90.47** | 76.00 |
| ResNetCNN<br>+<br>BILSTM [14] | | 74 | 49 | 39 | - |
| Xception<br>+<br>BILSTM [14] | | 76 | 48 | 45 | - |
| Proposed Models | Model 1 | 81.06 | 84.24 | 86.80 | 85.50 |
| | Model 2 | **85.00** | **87.77** | 84.44 | **86.08** |

and 85.50%. Our 2nd Model obtain accuracy, precision, recall and f1-score of 85.00%, 87.77%, 84.44% and 86.80%. Both model gives better result than previos methods except in recall. This analysis proves the robustness of the models.

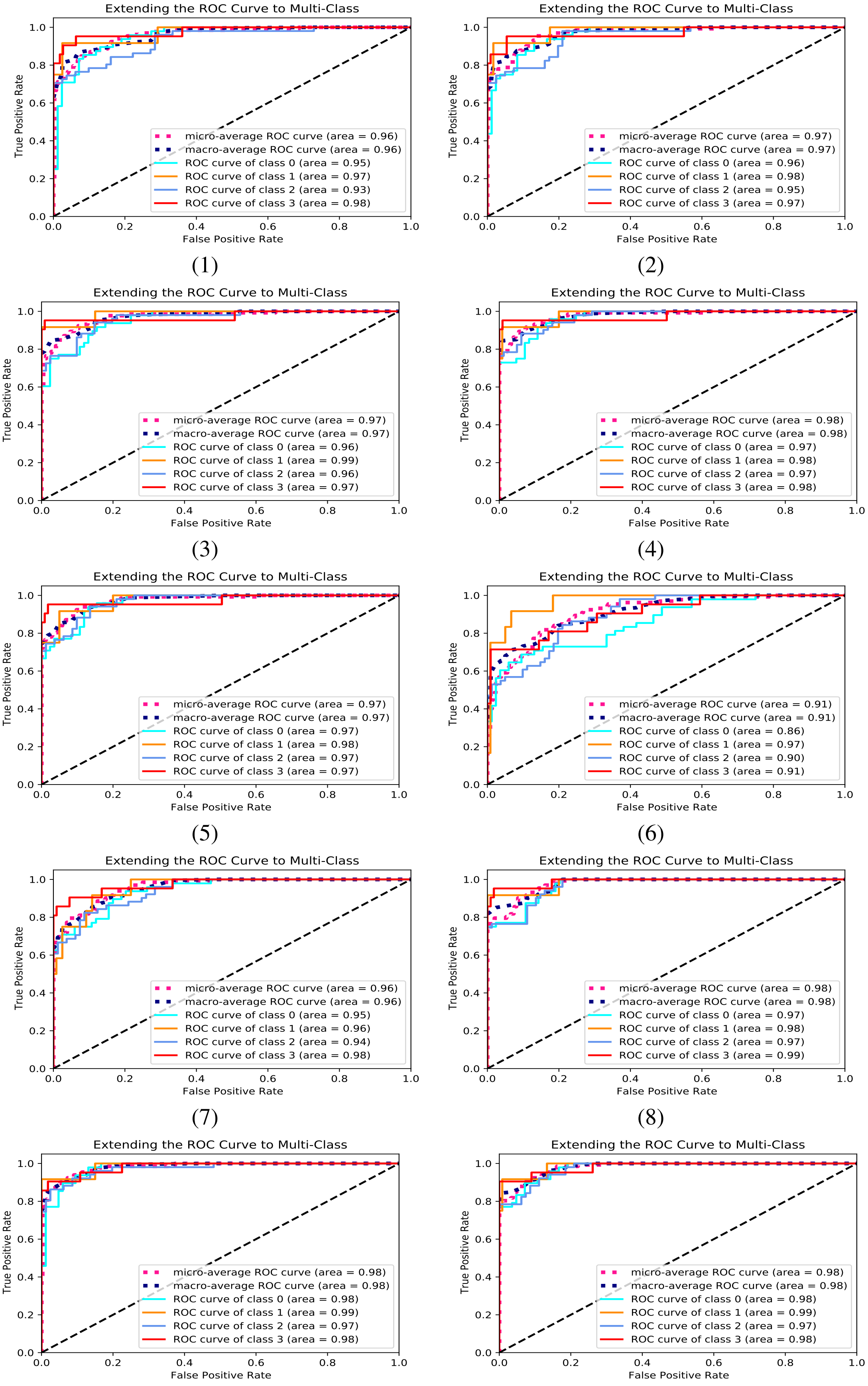
## 5.4 Discussion

In this chapter, the results of our both models are described briefly. Maximum average accuracy for Hypertension classification was found by using hybrid model of 2 Timedistributed CNN, 1 GRU and 3 dense layer with a 85.00% accuracy score. Our 2nd model of 3 CNN, 1 LSTM and 3 dense layer also gives good accuracy score of 81.06%.



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

**Figure 5.5:** Confusion matrix of Model 1 for 10 folds.



(1) (2)

(3) (4)

(5) (6)

(7) (8)

(9) (10)

**Figure 5.6:** ROC of Model 1 for 10 folds.

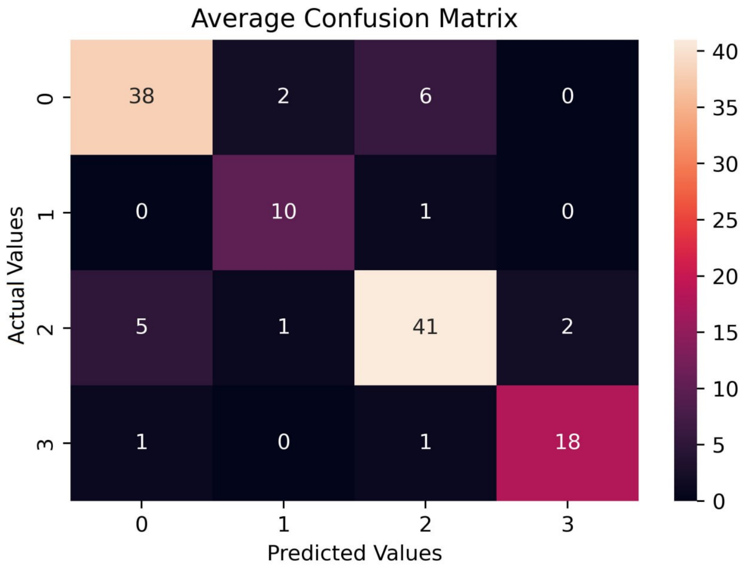


**Figure 5.7:** Average Confusion Matrix for Model 1

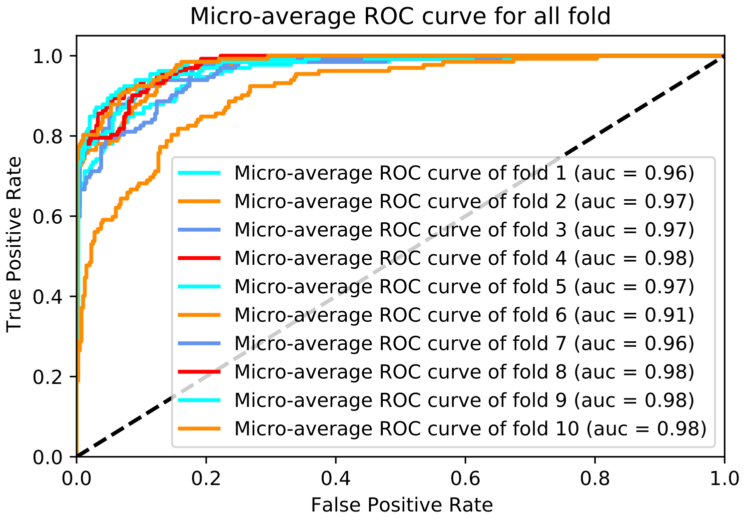
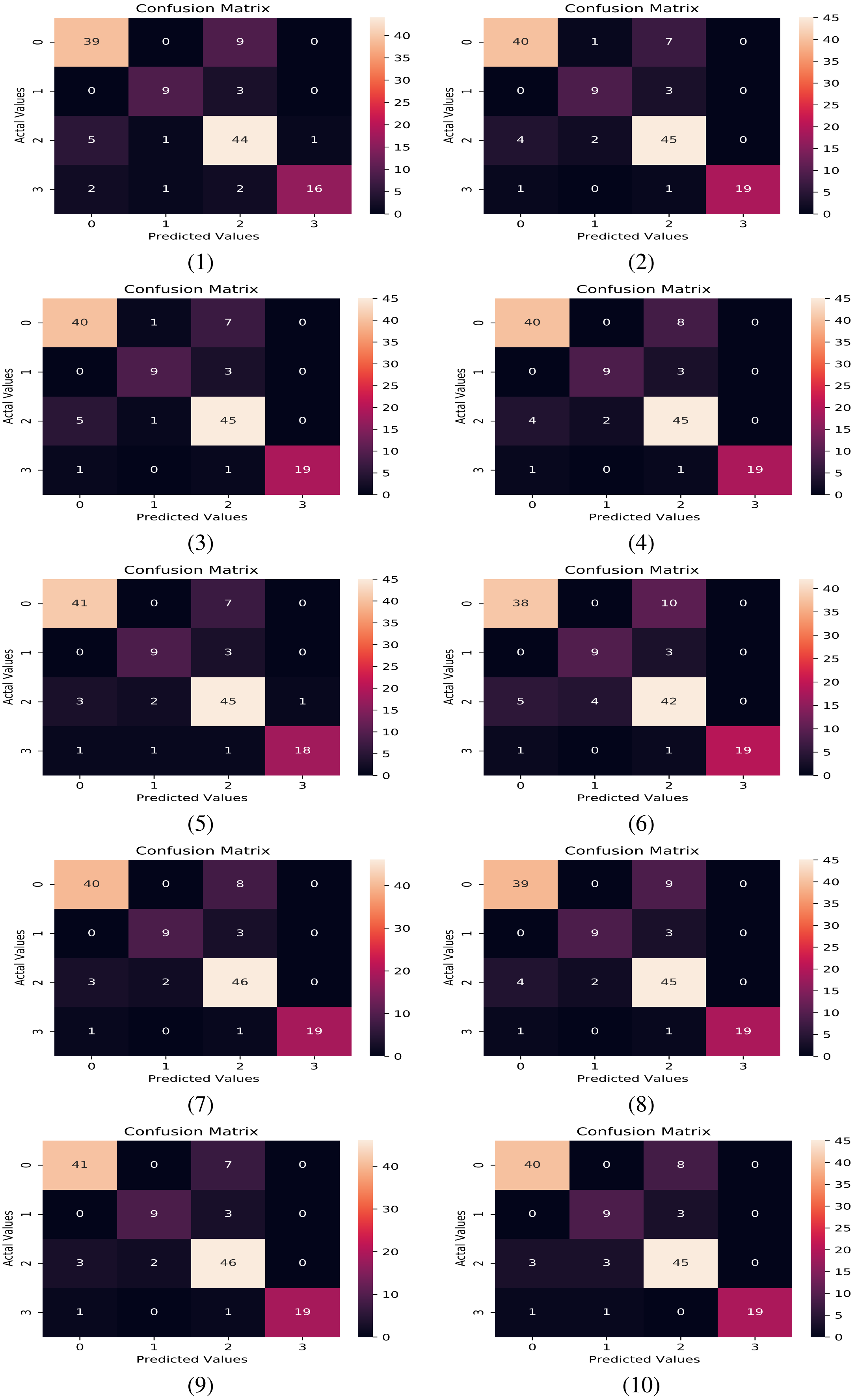


**Figure 5.8:** Average ROC for Model 1



(1)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 39 | 0 | 9 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 5 | 1 | 44 | 1 |
| 3 | 2 | 1 | 2 | 16 |

(2)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 40 | 1 | 7 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 4 | 2 | 45 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(3)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 40 | 1 | 7 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 5 | 1 | 45 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(4)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 40 | 0 | 8 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 4 | 2 | 45 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(5)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 41 | 0 | 7 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 3 | 2 | 45 | 1 |
| 3 | 1 | 1 | 1 | 18 |

(6)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 38 | 0 | 10 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 5 | 4 | 42 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(7)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 40 | 0 | 8 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 3 | 2 | 46 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(8)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 39 | 0 | 9 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 4 | 2 | 45 | 0 |
| 3 | 1 | 0 | 1 | 19 |

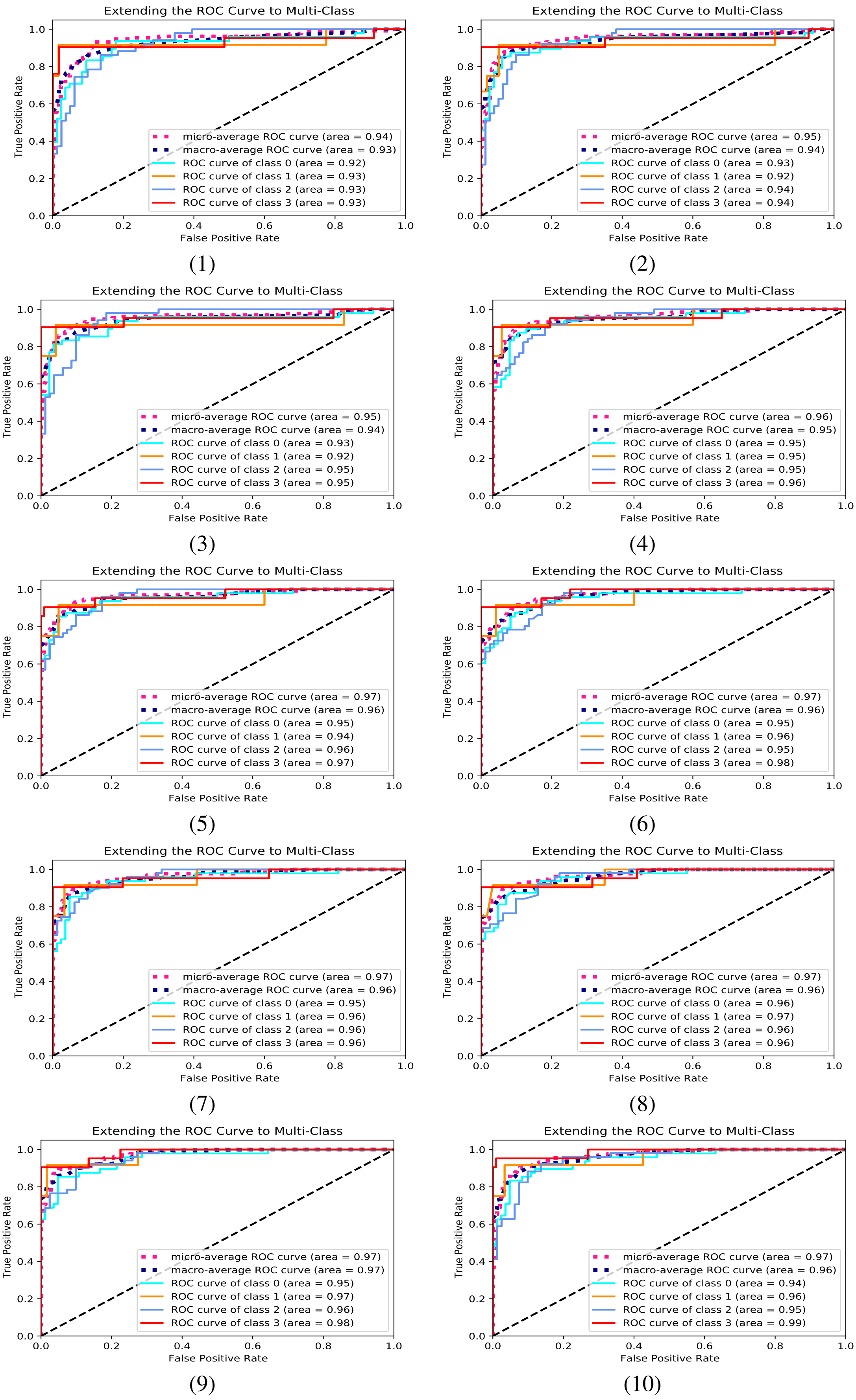
(9)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 41 | 0 | 7 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 3 | 2 | 46 | 0 |
| 3 | 1 | 0 | 1 | 19 |

(10)

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 40 | 0 | 8 | 0 |
| 1 | 0 | 9 | 3 | 0 |
| 2 | 3 | 3 | 45 | 0 |
| 3 | 1 | 1 | 0 | 19 |

**Figure 5.9:** Confusion matrix of Model 2 for 10 folds.



(1) (2)

(3) (4)

(5) (6)

(7) (8)

(9) (10)

**Figure 5.10:** ROC of Model 2 for 10 folds.

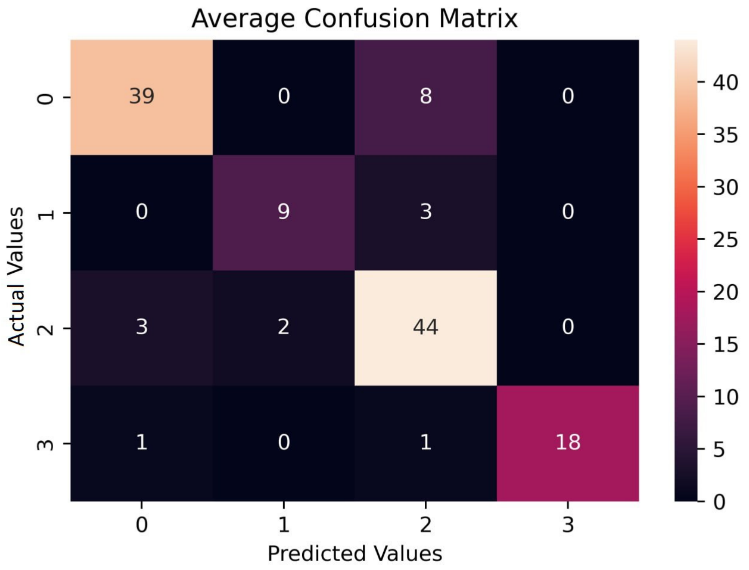


**Figure 5.11:** Average Confusion Matrix for Model 2

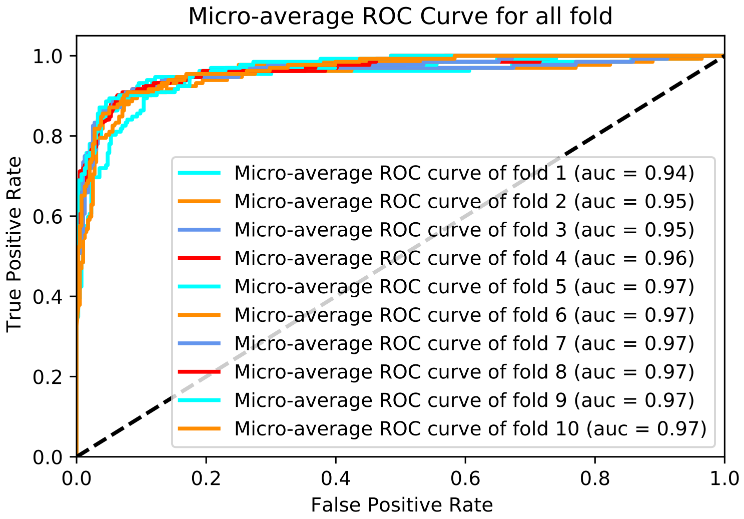


**Figure 5.12:** Average ROC for Model 2

# Chapter VI

# Conclusions and Future Works

## 6.1 Summary

In this thesis, we present a CNN-LSTM-GRU-based Hypertension classification system from PPG signals. We have considered the noises in the signals and use butterworth filter. We also use z-normalization. We didn't use any other feature extraction as our goal was to build a hybrid model which will classify Hypertension. In our both models, Convolutional layer is used as feature extractor. Then we use LSTM and GRU for classification as they are capable of learning long-term dependencies. To avoid overfitting, we use stratified kfold cross validation. This gives us great test accuracy. Our 1st Model obtain average accuracy, precision, recall and f1-score of 81.06%, 84.24%, 86.80% and 85.50%. Our 2nd Model obtain average accuracy, precision, recall and f1-score of 85.00%, 87.77%, 84.44% and 86.80%. They are comparatively better then other previous methods in classifying Hypertension.

## 6.2 Limitations

Though our proposed system is efficient, it has some limitations. This system's performance can be increased by bigger dataset.

## 6.3 Concluding Remarks

This study provided a method for predicting hypertension using Deep learning models. The results of our approach are promising. utilizing CNN-LSTM-GRU models and PPG Signal. It uses the signal to determine the 4 stages of Hypertension. Features are extracted from the PPG signal by Convolutional layer. Finally, 2 models: CNN-LSTM and CNN-GRU are constructed by employing stratified 10-fold cross-validation procedure. The effectiveness of our suggested solution has also been evaluated in comparison to existing approaches, demonstrating its applicability. We hope to favor the following methods in the future to further improve our work: larger data sets can be used, other deep learning methods can be implemented for better accuracy

## 6.4 Recommendations For Future Research

This thesis work can be extended in several ways. The following includes some possible areas that are recommended to extend the present work.

1. As our major goal is to present a model that will classify Hypertension without using other features, one can use other features to improve this model. .
2. Further studies can be conducted by training bigger dataset as our dataset is small.
3. Many variation of CNN LSTM GRU can be implemented to improve accuracy.

# References

[1] Fabian, V., Matera, L., Bayerova, K., Havlík, J., Křemen, V., Pudil, J., Sajgalik, P., and Zemanek, D., "Noninvasive assessment of aortic pulse wave velocity by the brachial occlusion-cuff technique: Comparative study," *Sensors*, vol. 19, p. 3467, 08 2019.

[2] Haddad, S., Boukhayma, A., and Caizzone, A., "Continuous ppg-based blood pressure monitoring using multi-linear regression," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 2096–2105, 2021.

[3] Muntner, P., Shimbo, D., Carey, R. M., Charleston, J. B., Gaillard, T., Misra, S., Myers, M. G., Ogedegbe, G., Schwartz, J. E., Townsend, R. R. *et al.*, "Measurement of blood pressure in humans: a scientific statement from the american heart association," *Hypertension*, vol. 73, no. 5, pp. e35–e66, 2019.

[4] Zhang, Q., Xie, Q., Duan, K., Liang, B., Wang, M., and Wang, G., "A digital signal processor (dsp)-based system for embedded continuous-time cuffless blood pressure monitoring using single-channel ppg signal," *Science China Information Sciences*, vol. 63, no. 4, pp. 1–3, 2020.

[5] Miao, F., Fu, N., Zhang, Y.-T., Ding, X.-R., Hong, X., He, Q., and Li, Y., "A novel continuous blood pressure estimation approach based on data mining techniques," *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1730–1740, 2017.

[6] Tabei, F., Gresham, J. M., Askarian, B., Jung, K., and Chong, J. W., "Cuff-less blood pressure monitoring system using smartphones," *IEEE Access*, vol. 8, pp. 11 534–11 545, 2020.

[7] Teng, X. and Zhang, Y., "Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 4, 2003, pp. 3153–3156 Vol.4.

[8] Maqsood, S., Xu, S., Springer, M., and Mohawesh, R., "A benchmark study of machine

learning for analysis of signal feature extraction techniques for blood pressure estimation using photoplethysmography (ppg)," *IEEE Access*, vol. 9, pp. 138 817–138 833, 2021.

[9] Mangathayaru, N., Padmaja Rani, B., Janaki, V., Patel, S. A., Sai Mohan, G., and Lalith Bharadwaj, B., "An imperative diagnostic framework for ppg signal classification using gru," in *Advanced Informatics for Computing Research: 4th International Conference, ICAICR 2020, Gurugram, India, December 26–27, 2020, Revised Selected Papers, Part I 4*. Springer, 2021, pp. 606–621.

[10] Wu, J., Liang, H., Ding, C., Huang, X., Huang, J., and Peng, Q., "Improving the accuracy in classification of blood pressure from photoplethysmography using continuous wavelet transform and deep learning," *International journal of hypertension*, vol. 2021, 2021.

[11] Tjahjadi, H., Ramli, K., and Murfi, H., "Noninvasive classification of blood pressure based on photoplethysmography signals using bidirectional long short-term memory and time-frequency analysis," *IEEE Access*, vol. 8, pp. 20 735–20 748, 2020.

[12] Kuzmanov, I., Vasilevska, A., Madevska Bogdanova, A., Ackovska, N., Kostoska, M., and Lehocki, F., "Blood pressure classification using cnn-lstm model," 2022.

[13] Martinez-Ríos, E., Montesinos, L., and Alfaro-Ponce, M., "A machine learning approach for hypertension detection based on photoplethysmography and clinical data," *Computers in Biology and Medicine*, vol. 145, p. 105479, 2022.

[14] Yen, C.-T., Chang, S.-N., and Liao, C.-H., "Deep learning algorithm evaluation of hypertension classification in less photoplethysmography signals conditions," *Measurement and Control*, vol. 54, no. 3-4, pp. 439–445, 2021.

[15] Fuadah, Y. N. and Lim, K. M., "Classification of blood pressure levels based on photoplethysmogram and electrocardiogram signals with a concatenated convolutional neural network," *Diagnostics*, vol. 12, no. 11, p. 2886, 2022.

[16] Mejía-Mejía, E., May, J. M., Elgendi, M., and Kyriacou, P. A., "Classification of blood pressure in critically ill patients using photoplethysmography and machine learning,"

*Computer Methods and Programs in Biomedicine*, vol. 208, p. 106222, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260721002960

[17] Liang, Y., Liu, G., Chen, Z., and Elgendi, M., "PPG-BP Database," 2 2018. [Online]. Available: https://figshare.com/articles/dataset/PPG-BP_Database_zip/5459299

[18] Liang, Y., Elgendi, M., Chen, Z., and Ward, R., "An optimal filter for short photoplethysmogram signals," May 2018. [Online]. Available: https://www.nature.com/articles/sdata201876#citeas