

*University of Essex*

**Department of Computer  
Science and Electronic  
Engineering**

---

CE901-7: DISSERTATION

**Respiratory rate estimation: A deep learning  
approach for robust estimation of  
respiratory rate using PPG**

**Safi Sohail Azim**  
**2110836**

Supervisor: Dr. Delaram Jarchi

---

December 14, 2022  
Colchester

## **ABSTRACT**

Respiratory rate estimation is a critical metric in detecting patients' deteriorating health and is a vital metric in finding out many diseases such as pneumonia in children and in adults. Sadly, estimating respiratory rate is not very easy to do with inexpensive equipment and outside hospital settings. Robust estimation of respiratory rate can benefit millions of individuals. In my study, I have provided several machine-learning approaches and deep-learning approaches. This study process 9 machine learning algorithms, one RNN algorithm, and one ANN algorithm. The mentioned algorithms are trained and tested on the BIDMC dataset which is an open-sourced data set extracted from a larger dataset called MIMIC II dataset. The study begins with the implementation of 9 machine learning algorithms to learn and predict respiratory rate from the PPG data acquired from inexpensive equipment such as a pulse oximeter and then to predict the RR with the least possible error close to the actual reading taken from sophisticated and expensive equipment. The study moves forward with the implementation of deep learning models such as LSTM and Dense neural networks to show how data-intensive algorithms can further improve the results. A comparative study is carried out to measure the efficiency of each approach. The metrics which are used for each algorithm are mean absolute error and root mean squared error. The data window of 32-s PPG is used to test all the models. The best-performing model out of the researched model was the random forest regression model with an MAE of 0.04 but in the deep learning approach, the implementation of the dense neural network provides an MAE of 0.169 which is better than the LSTM approach which has an MAE of 3.86.

## **LIST OF KEYWORDS:**

Mean absolute error, Long-short term memory, Root mean square error, artificial neural network, dense neural network, Respiratory rate, recurrent neural network, photoplethysmogram, electrocardiogram

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Problem Statement . . . . .	10
1.1.1	Monitoring in the hospital setting . . . . .	10
1.1.2	Monitoring outside the hospital setting . . . . .	11
<b>2</b>	<b>Related Works</b>	<b>12</b>
2.1	Respiratory rate estimation from the PPG using multiparameter approach . .	12
2.2	Deep learning approach for respiratory rate estimation using PPG . . . . .	17
2.3	Estimation of respiratory rate from pulse oximeters . . . . .	20
2.4	Implementing respiration rate estimation methods using PPG from the Cap-nobase dataset . . . . .	22
2.5	A machine learning approach to the non-invasive estimation of the Respiratory rate using a corrupted PPG signals dataset . . . . .	24
<b>3</b>	<b>Data Acquisition and proposed methodology</b>	<b>29</b>
3.1	Data Files . . . . .	29
3.2	Data preparation . . . . .	31
3.3	Data preprocessing . . . . .	31
3.4	Methodology . . . . .	34
3.4.1	Linear Regression . . . . .	36
3.4.2	Elastic Net . . . . .	36
3.4.3	Bayesian Ridge . . . . .	37
3.4.4	KNN Regression . . . . .	37
3.4.5	Random Forest Regression . . . . .	37
3.4.6	Ada Boost Regression . . . . .	38
3.4.7	Lasso Regression . . . . .	38

3.4.8 Ridge Regression . . . . .	38
3.4.9 Extreme gradient boost regression . . . . .	38
3.4.10 Neural Network . . . . .	39
3.4.11 Long-short-term memory (LSTM) . . . . .	39
3.5 Evaluation Metrics . . . . .	41
<b>4 Architecture</b>	<b>43</b>
4.1 Machine learning models for RR estimation . . . . .	43
4.2 RNN model for respiratory rate estimation . . . . .	44
4.3 ANN model for respiratory rate estimation . . . . .	45
<b>5 Experimental results and analysis</b>	<b>47</b>
5.1 Experimental analysis . . . . .	47
5.1.1 Machine learning models analysis of results . . . . .	48
5.1.2 LSTM model analysis of results . . . . .	50
5.1.3 DNN analysis of results . . . . .	53
5.2 Comparison of results . . . . .	54
<b>6 Conclusion and Future Works</b>	<b>55</b>
6.1 future works . . . . .	55
<b>A Code link</b>	<b>57</b>

---

# List of Figures

2.1	PPG biosignals extraction [13] . . . . .	12
2.2	IMS Algorithm implementation [13] . . . . .	14
2.3	RMS metrics results [13] . . . . .	16
2.4	Modulation of PPG in the paper [3] . . . . .	18
2.5	ResNet Architecture [3] . . . . .	19
2.6	Physiosignals extraction method [25] . . . . .	21
2.7	Architectural representation of implementation [14] . . . . .	23
2.8	Modulation of the data [31] . . . . .	25
2.9	Features of the Data [31] . . . . .	26
2.10	Threshold of the RR [31] . . . . .	26
3.1	1 Hz table . . . . .	32
3.2	125 Hz table . . . . .	32
3.3	1 Hz and 125 Hz combined . . . . .	32
3.4	persons final dataset . . . . .	33
3.5	persons dropped for errors . . . . .	33
3.6	normalized RR . . . . .	33
3.7	PPG signals . . . . .	34
3.8	PPG signals . . . . .	34
3.9	DNN architecture [16] . . . . .	35
3.10	LSTM model [20] . . . . .	40
3.11	MAE equation [41] . . . . .	41
3.12	RMSE equation [41] . . . . .	41
3.13	RMSE table [41] . . . . .	42
4.1	LSTM architecture . . . . .	45

5.1	ML algorithms output . . . . .	48
5.2	Erros in train and test set . . . . .	49
5.3	Features valuation . . . . .	49
5.4	Feature importance . . . . .	50
5.5	Enhanced RR . . . . .	50
5.6	Entire dataset RR . . . . .	51
5.7	Derivative of RR . . . . .	51
5.8	Thresholded Derivative of RR . . . . .	51
5.9	RR rate data . . . . .	52
5.10	Training and Validation loss . . . . .	52
5.11	Training and Validation loss . . . . .	53
5.12	Training and Validation loss graph . . . . .	53
5.13	Training and Validation loss MAE . . . . .	54

---

## List of Tables

5.1 Results comparison .....	54
------------------------------	----

---

## Introduction

The administration of technology in the medical field plays an important role in saving human lives. One of the prominent diagnostic identifiers in the medical field is the respiratory rate which predicts several diseases ranging from Pneumonia in young children [40] to cardiovascular diseases [33], early detection of patient deterioration health [1], and respiratory dysfunction. Respiratory rate is an essential criterion in the early diagnosis and treatment of the patient [11]. However, measuring accurate respiration rate is often not easy to measure and is quite expensive outside of medical settings [21].

Obtaining the correct respiratory rate from an easy-to-use and inexpensive device is the approach used to implement the extraction of the respiratory rate that can be used to administer life-saving actions. Measurement of continuous respiratory rate can be extracted by using one or a combination of impedance pneumography (IP), acceleration, piezoelectric, electrocardiogram (ECG), and photoplethysmogram (PPG) [29]. These physiological measurements are highly subjective and are prone to inaccuracy due to the way they are extracted.

Pulse oximeters [8] are the go-to approach that offers the possibility of extracting respiratory rate if used with inexpensive devices such as wearables, smart watches, etc. Pulse oximeters use Beer-Lambert law to estimate the respiratory rate in which red and infrared wavelengths measure relative changes in blood profusion over time [22]. This provides a variation of blood volume over time with a photoplethysmogram (PPG) which contains a pulsatile component and a constant component that can be used for respiratory rate [30].

These PPG signals can then either be modulated into baseline wander (BW) which shows changes in intrathoracic pressure and vasoconstriction of arteries during breathing, Amplitude modulation (AM) which shows changes in stroke volume and intrathoracic pressure during breathing cycles and frequency modulation (FM) which shows the respiratory sinus arrhythmia that increases when the person inhales and to decrease when the person exhales. Besides physiological signal modulation features extraction can be implemented from PPG to extract features that are needed to estimate respiratory rate such as kurtosis, skewness, mean variation, etc [5].

$$Kurt = \mu^4 / \sigma^4 \quad (1.0.1)$$

$$Skewness = \sum_i^N \frac{(\chi_i - \bar{\chi})}{(N - 1) * \sigma^3} \quad (1.0.2)$$

Since the respiratory rate is very difficult to estimate as different environments provide different results for example, in the position of rest the person has an average respiratory rate between 12-20 breaths per minute while a child sick with pneumonia has an elevated respiratory rate of more than 40 breaths per minute [6]. Due to such complex nature of respiratory rate estimation, there are numerous approaches for RR estimation which are: sensor fusion, digital filtering, signal quality estimates, time or frequency domain analysis, fiducial points from waveforms or features, and signal decomposition [28]. These are all classical approaches for the estimation of RR as they use handcrafted rules and empirical parameters which is highly optimized for specific algorithmic approach and even patients.

In contrast, the approach that I have used in my project revolves around the implementation of machine learning and deep learning algorithms involving PPG with time series signals. Deep neural networks and Recurrent neural networks (RNN) are the main approaches followed in my study. However, there is one major limitation to the deep neural networks and RNN i.e., they require a large amount of relatively annotated data for the optimal performance of the algorithms [24]. These models are therefore trained using a combination of real-world data, synthetic data, and augmented data in order to fulfill their data-intensive

needs. The performance is assessed using different combinations and tuning of the algorithm with benchmark approaches and measurement metrics.

The aim of the research is to implement several modern machine learning algorithms, deep neural networks, and recurrent deep neural networks to determine the performance of each approach and to present an efficient processing algorithm for robust estimation of respiratory rate using data collected through an inexpensive device within or outside the clinical environment as in the clinical measurement of RR with inexpensive equipment has shown to have poor reliability and repeatability.

## 1.1 Problem Statement

Respiratory rate is a vital signal in several types of physiological deterioration in patients [34]. Its correct values play an important part in monitoring and administrating help in the medical and outside medical settings. The below-mentioned problems raise concerns about the solution to the robust estimation of RR using physiological signals.

### 1.1.1 Monitoring in the hospital setting

In hospitals and clinic settings usually vital signals such as Respiratory rate are assigned pre-determined scores and these scores are monitored every 4-6 hours by senior medical staff. These scores are very important to a patient's health as they can predict the deterioration in a patient's health [10]. However, these score values cannot be measured continuously. If they were measured continuously, they can provide early warning on a continuous basis. Even though heart rate and peripheral blood oxygen saturation can be continuously measured using devices such as pulse oximeter and a few extra devices, the robust continuous estimation of respiratory rate can only be done through bio-signals like PPG and ECG acquired through a pulse oximeter [38]. Hence, this creates a need for the development of an approach that can be used to continuously to estimate the RR of patients even in medical settings with expensive types of equipment.

### 1.1.2 Monitoring outside the hospital setting

In contrast to hospital settings, using expensive and adhesive sensors and devices are difficult to use when the patient is at home, therefore devices such as pulse oximeters are used in which the patient just need to insert their finger in the probe for continuous readings. Pulse oximeters can provide a robust estimation of heart rate and SpO<sub>2</sub>, but they cannot provide a robust estimation of RR using PPG signals [12].

## Related Works

Previously, many academics and researchers have worked on this problem to provide a solution to the robust estimation of respiratory rate from physiological signals such as PPG and ECG. Furthermore, researchers and academics are working on better and faster technologies, tools, and approaches using advanced and upcoming technological advancements in the field of MedTech. The following literature describes a few of the recent and advanced technologies used in this domain to estimate RR.

### 2.1 Respiratory rate estimation from the PPG using multiparameter approach

Multiparameter respiratory rate estimation from the PPG is the approach carried out in this paper [13]. The researchers worked on PPG signals obtained from pulse oximeters and used these bio-signals to extract 3 of the most important respiratory-induced variations.

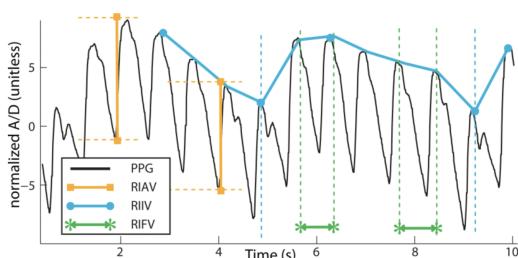


Figure 2.1: PPG biosignals extraction [13]

These bio-signals are obtained using an incremental merge segmentation algorithm to extract them from the PPG waveform. The incremental merge segmentation algorithm is a proprietary algorithm made to extract respiratory rate-induced signals. Then Fast Fourier transformation is used to analyze the frequency content of each respiratory-induced variation. These respiratory-induced variations are:

1. Respiratory-induced frequency variation (RIFV): The variation of Heart rate to synchronize with the respiratory cycle due to an autonomic response to respiration which is also referred to as respiratory sinus arrhythmia (RSA). Due to the increase and decrease in heart rate caused by inspiration and expiration, respiratory-induced variation is deduced.
2. Respiratory-induced intensity variation (RIIV): The exchange of blood between the pulmonary circulation and the systemic circulation caused by intrathoracic pressure variation results in the variation of perfusion baseline.
3. Respiratory-induced amplitude variation (RIAV): The change in peripheral pulse strength which corresponds to the decrease in cardiac output due to reduced ventricular filling [4].

In this paper i.e., multiparameter respiratory rate estimation from the PPG, evidence from another study was taken as the baseline which states that there is a correlation between each respiration-induced waveform variation with the respiratory cycle [18]. The result of this related study showed that there is no significant variation, as the performance of the correlation of these bio-signals depends upon many factors such as RR, gender, and body position. Outside of the controlled environment, extraneous variables can affect the respiratory segment of the PPG due to measurement artifacts and physiological variability between patients. Examples of such extraneous variables can be that the amplitude of variation may be stronger in dehydrated patients, in chronic disease conditions such as diabetes the RSA may be very minute, etc.

Considering all these technological advancements, RR measurements through pulse oximeters were still a very new approach as this research was published in 2013. Instead, in clinical settings more reliable methods such as capnometry and spirometry were used for RR estimation. However, due to these devices being expensive and hard to get, a pulse oximeter was the approach used for RR estimation even though they provide PPG signals

with significant artifacts and noise as it is prone to clinical errors. In this research paper, the researchers want to implement an algorithm that can be used to take PPG data from the pulse oximeter, send it to a mobile device and use the PPG with their algorithm to estimate RR. The researchers in this paper have implemented RR estimation using multiple parameters from PPG. These are:

1. Regular heartbeat pulsations denoted by a maximal volume peak for the extraction of RIAV
2. A lower frequency pattern belonging to the RIIV is denoted by a sinusoidal variation of the signal baseline.

These 2 parameters are then extracted from the PPG signals by implementing an incremental merge segmentation algorithm which splits the PPG signals into pulses. PPG signals can be denoted by consecutive lines as they are composed of morphological shapes hence permitting the desired output calculation in an efficient manner. For each parameter, the respiratory frequency information is then extracted using a maximal spectral power approach. The researcher's algorithm performs artifact detection during the IMS stage as artifact identification and removal are essential in PPG processing. They finally then estimated RR by fusing the respiratory frequency from RIFV, RIIV, and RIAV. Artifact and these 3 components extracted from bio-signals are used to estimate the quality of RR estimation. Specifically, a DC component is removed from the PPG signal by implementing a high pass filter. However, the researchers have implemented this step before the PPG is provided from the pulse oximeter as the cutoff frequency is controlled by the pulse oximeter. The IMS which is used for pulse segmentation contains a sliding window that is easy to implement, quick, and can be computed in real-time.

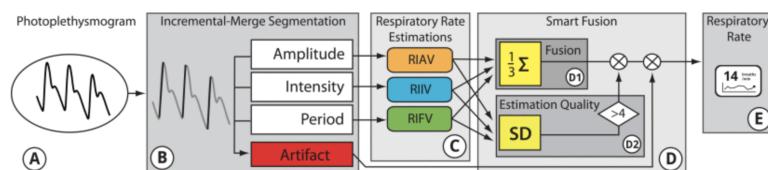


Figure 2.2: IMS Algorithm implementation [13]

Their algorithm just requires tuning of one parameter i.e., length of line segments in several sample points minus one, which is dependent on the sampling rate. Therefore, after the pulse segmentation process, each PPG pulse is represented as a straight line beginning at the pulse start and ending at its initial peak. This straight-line peak portrays the pulse amplitude and is proportional to the peripheral perfusion of the patient. In short, the IMS algorithm provides the maximum and minimum intensity of the pulse, pulse amplitude, and pulse period. All these features extracted from the PPG are used for RR estimation and noise detection.

They have detected artifacts from the pulse segment during the IMS processing which scans for abnormal pulse periods as data that contains pulse periods outside the normal range i.e., 230-2400 MS which are then automatically marked as artifacts. This method only detects artifacts through motion. Noise artifacts are detected and dropped during the smart fusion method.

Of the 3 components of the pulse segmentation, RIFV is calculated using Fast Fourier transform (FFT) as pulse periods are converted into tachograms. Tachogram is then resampled onto an even 4-Hz grid using Berger's algorithm as FFT requires evenly sampled data. Data are divided into sliding windows with the power of two sample points and the resulting power spectrum of each window is then analyzed for the frequency with maximum power within the expected respiratory frequency range of 4 to 65 breaths/min. To extract RIIV the maximum intensity of PPG pulses is used. Linear interpolation was used to resample intensity sample data onto a 4 Hz grid. Like RIFV, the hamming window is multiplied by the data divided into sliding windows with 1-s shifts. The power spectrum of each window in the FFT is then analyzed for the maximum frequency content within the RR frequency range. Lastly, for RIAV the data is again divided into sliding windows with 1-s shifts and multiplied with a Hamming window. The power spectrum of each window in the FFT is then analyzed for the maximum frequency content within the RR frequency range.

Moreover, the RR estimation is characterized for consistency as standard deviations are then calculated for all 3 pulse segments and any standard deviation above 4 breaths/min is marked as low RR estimation quality. Finally, all 3 segments of the pulse are fused by calculating their mean. This process is enhanced by implementing smart fusion which eliminates windows containing noise or low RR estimation quality. Data for this study were collected from 59 children from age 6 months to 16 years and from 35 adults aged 26 years to

75 years. The standard pulse oximeter is used to collect physiological signals from people.

The data is prepared in such a way that 42 eight-minute segments from 29 children and 13 adult cases containing reliable recordings of spontaneous or controlled breathing were randomly selected from each case for the Test Dataset. One 2 min segment was randomly selected from the remaining 52 cases for the train Dataset. For measurement metrics, the root means square metric was used. All RMS errors were tested for Gaussian distribution using the Lilliefors test at a level of  $p < 0.05$ . A balanced analysis of variance (ANOVA) with fixed effect was performed on the errors from RIAV, RIFV, RIIV, and Smart Fusion. 16s, 32s, and 64s windows were used to test the performance of the algorithm. The results using RMS metrics for the 3 windows are shown in 2.3.

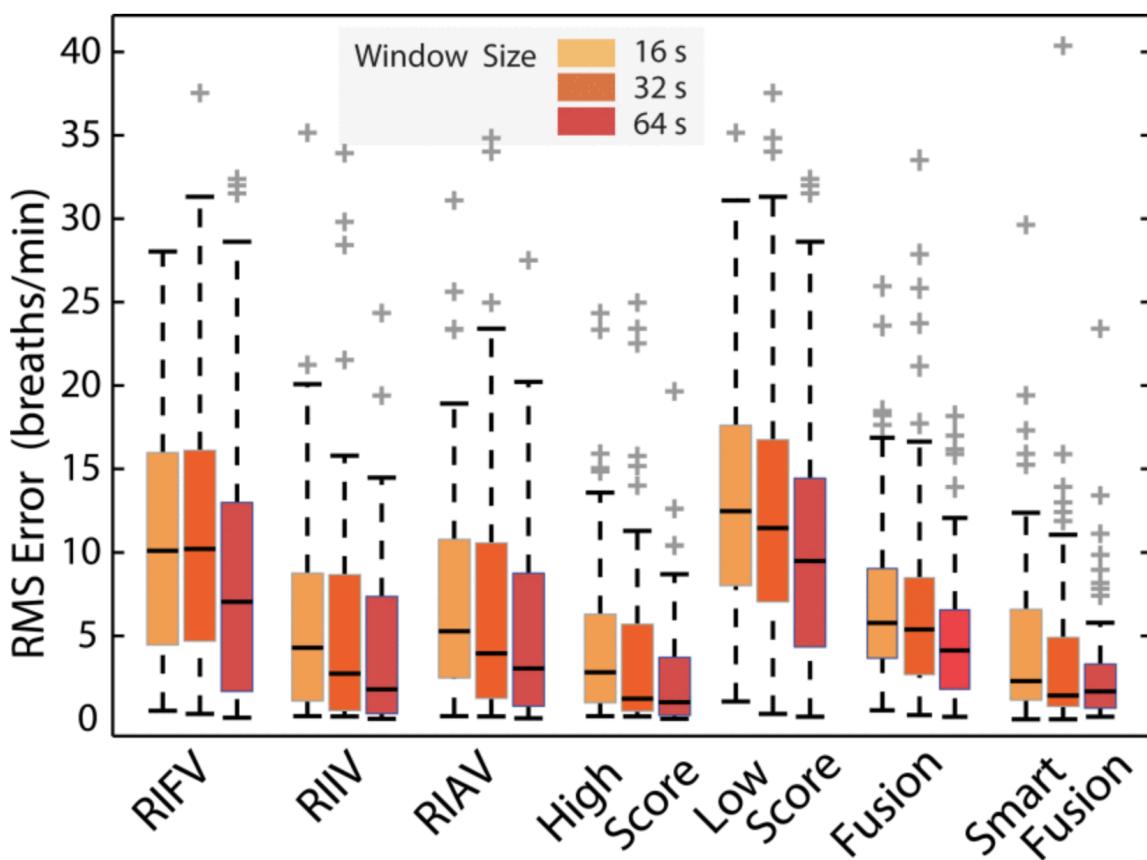


Figure 2.3: RMS metrics results [13]

## 2.2 Deep learning approach for respiratory rate estimation using PPG

An end-to-end deep learning approach based on convolutional neural network architecture is currently proposed for RR estimation using PPG in this paper [3]. The researchers fed the deep learning model with raw PPG as input, let the deep learning model learn the rules through training, and then estimate RR as an output. During this whole training process, the researchers have decided to provide the data-hungry deep learning model with real-world data combined with synthetic and augmented data to improve the results and train the model intensively. However, systematic combinations of real-world, synthetic, and augmented data are used to see which combination provides the optimum results.

Moreover, these systematic implementations of respiratory rate estimation are compared with the classical approach of estimating RR in this paper to introduce and decide whether the deep learning approach is suitable for this problem statement. In the classical respiratory rate estimation method fusion of independent RR estimates from multiple respiratory signals is implemented. Firstly, the raw PPG is down-sampled and filtered so all the low-frequency PPG components are removed. Following that, a peak detection method is implemented to determine the fiducial peak for each beat with empirical parameters tuned for the PPG signal. The feature-based techniques were used to extract the respiratory signals which are:

1. Base Wander: It is extracted by calculating the mean amplitude between the peaks and the preceding troughs
2. Amplitude modulation: It is extracted by calculating the difference between the amplitudes of peaks and proceeding troughs
3. Frequency modulation: It is extracted as the inter peaks intervals

After the extraction of the above-mentioned bio-signals a respiratory peak detection algorithm is used in this classical approach to estimate the respiratory rate from each of these independent surrogate signals. Following the implementation of the peak detection algorithm, the standard deviation of RR estimation from each of the surrogate bio-signals is implemented and calculated. After the calculation of the Standard deviation of each independent bio-signal, RR estimates were set to be invalid if SD is larger than a predetermined threshold. The

thinking behind the implementation of this independent standard deviation is that the RR outputs from different respiratory signals should not vary too much. Once all the variations are examined and implemented, a quality metric ranging from 1 to 0 is calculated for each RR estimate from each independent bio-signals, the final RR is calculated as the weighted mean of the RR from all three respiratory signals. This classical method is implemented on a public dataset containing 60 PPG segments in each record. Mean absolute error is the metric used to evaluate the performance of the classical method.

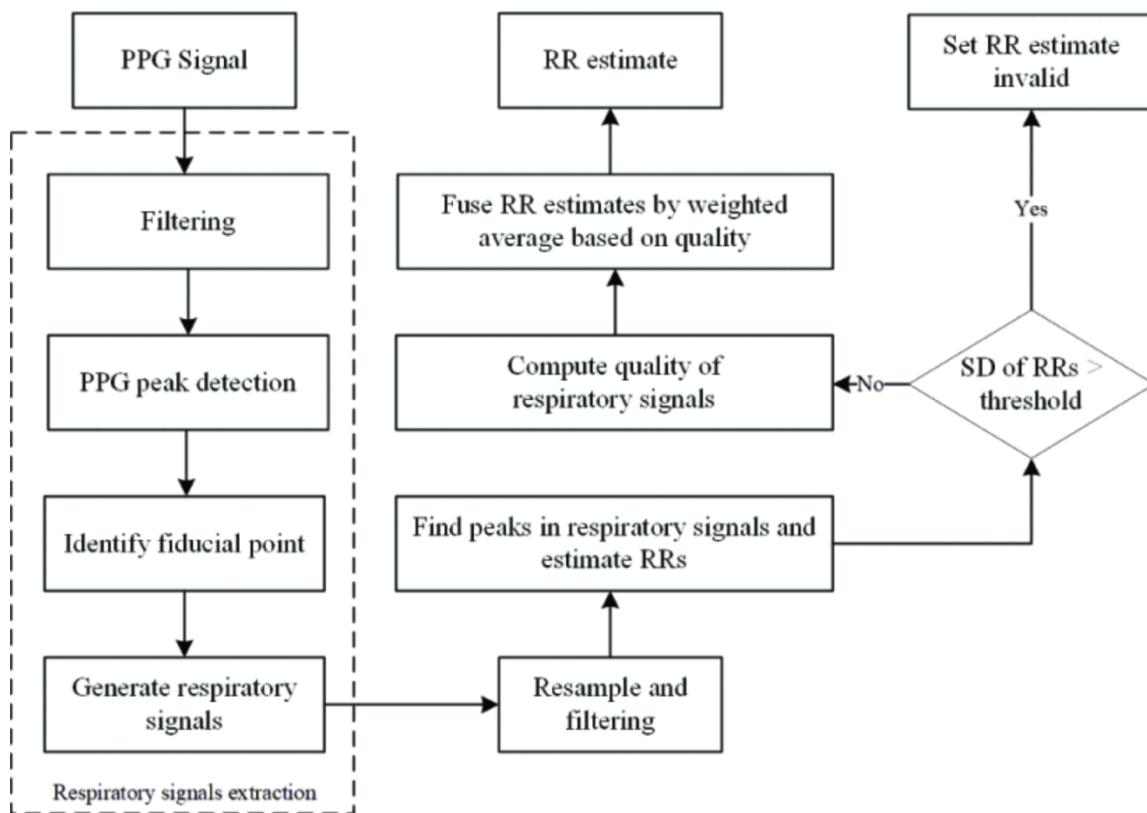


Figure 2.4: Modulation of PPG in the paper [3]

However, in this study continuous time series PPG signals are input in a convolutional neural network (CNN) architecture. A popular variant of CNN architecture called the residual network is being implemented by researchers for the measurement of RR using the PPG signals. Shortcut connections are inserted into convolution layers by ResNet which makes the CNN network turn into its counterpart residual version. Since ResNet are easier to optimize, has lower computational complexity, and can gain an increased performance boost with the increase of its depth so they are best to approach for real-time RR rate estimation from the PPG signals. As the ResNet model preserves the PPG signal's integrity and reduces the model complexity along with its computational cost so in this study the researchers resampled one-dimensional raw PPG signals to 30 Hz before inserting it into the ResNet model. The architecture of the deep learning model is shown in 2.5.

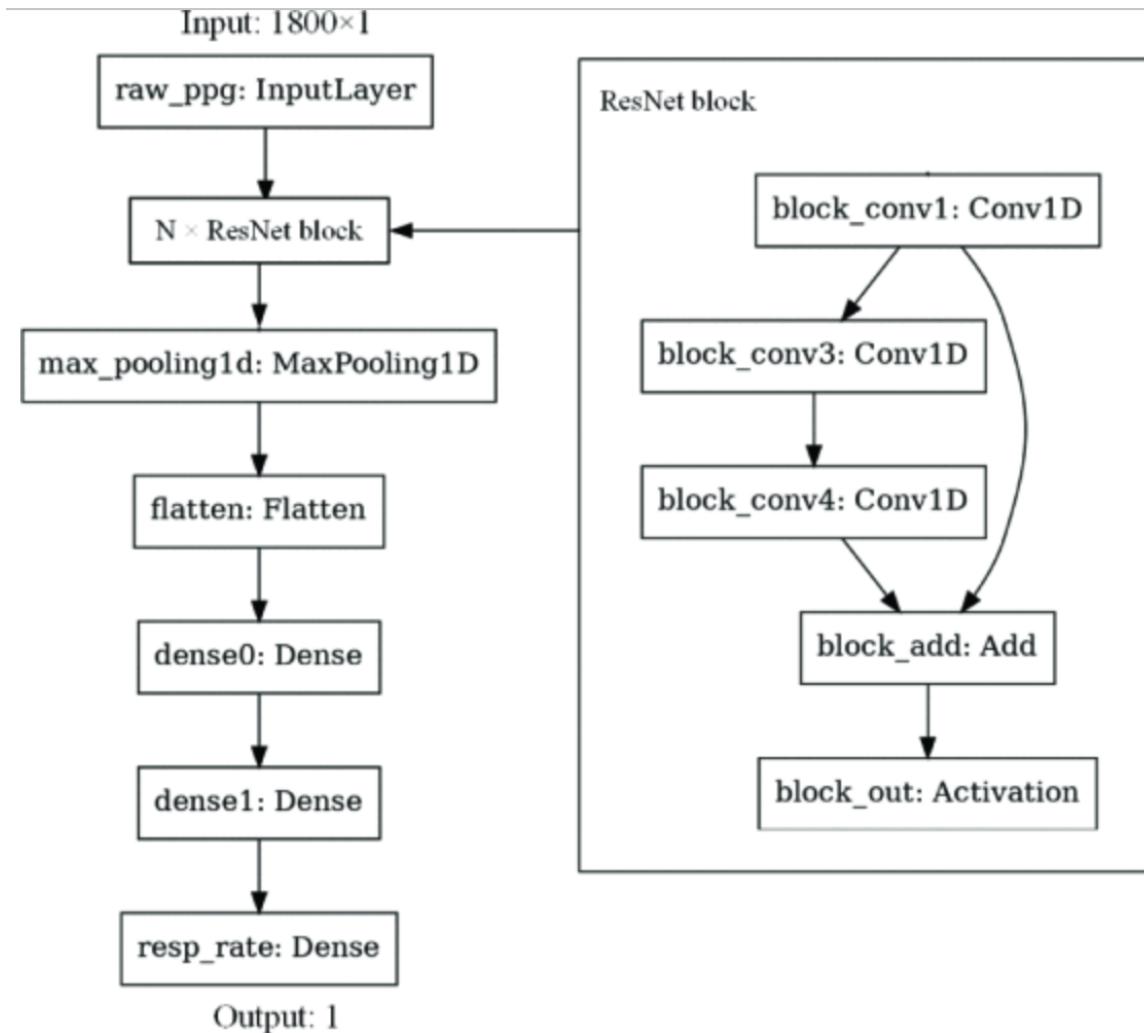


Figure 2.5: ResNet Architecture [3]

In this study, once the down-sampled PPG is sent into the deep learning model, the initial layers of the CNN filter the raw signal. The middle layers then use the PPG to learn and construct features throughout the middle layers. To reduce the non-linearity of the model ReLU activation function is used throughout the model except for the output layers. The output layer then outputs the continuous value for RR. The loss function which is used in this dense layer is called mean absolute error. Hyperparameters are tuned likewise for the optimal performance of the algorithm given the output. ResNet block, Filter size, kernel size, stride for convolutional layer, etc. are the hyperparameters that need tuning. The deep neural network is trained on real, synthetic, and hybrid data for the output results. The final performance of the algorithm was found by calculating the mean of the deep learning approach which gives RR estimation at 2.5 bpm. Capnibase and BIDMC datasets were used in this study along with synthetic data.

## 2.3 Estimation of respiratory rate from pulse oximeters

In this paper, a method for robust estimation of RR is implemented after taking PPG signals from the pulse oximeter [25]. The researchers published this paper in 2017. At that time there were estimation methods for RR from the pulse oximeters, but they were not very robust which led to the lack of use of pulse oximeters in and outside of the clinical environment. The researchers in their research have decided to use multiple autoregressive models of a different order, to determine the dominant frequency from the 3 respiratory-induced bio-signals which are frequency, amplitude, and intensity. The performance of these autoregressive models was compared with classical established models such as digital filters, Fast Fourier transformation, wavelet decomposition, and joint time-frequency analysis.

In this study, the researchers used the incremental merge segmentation algorithm which was proposed in [17] to extract the 3 vital respiratory-induced variations which are amplitude, frequency, and intensity. These bio-signals were then resampled to 4Hz using linear interpolation resulting in a time series of peaks and troughs. The purpose of this resampling is to implement autoregressive modeling straightaway. Furthermore, each resampled time series is normalized by using zero mean unit variance transformation to have all the 3 bio-signals with the same dynamic range. 2.6 represents the extraction method of the 3 respiratory induced variations.

Autoregressive modeling which is used in this study for robust estimation of the RR required a respiratory signal which was extracted by applying a lowpass or bandpass filter that reduces the component at the cardiac frequency in the PPG signal. Here the researchers determine the frequency by pole angles as the poles are generated by resonant frequencies corresponding to AR models. The pole with the greatest magnitude identifies the respiratory pole and the accompanying range of respiratory frequencies. The proposed methods in this study estimate the RR by combining the spectral estimates of the 3 induced physiological signals from the PPG as autoregressive models are used to calculate the power spectral density using multiple AR models. The model works on the assumption that an AR process governs each input time series that in response assumes the current values of the time series.

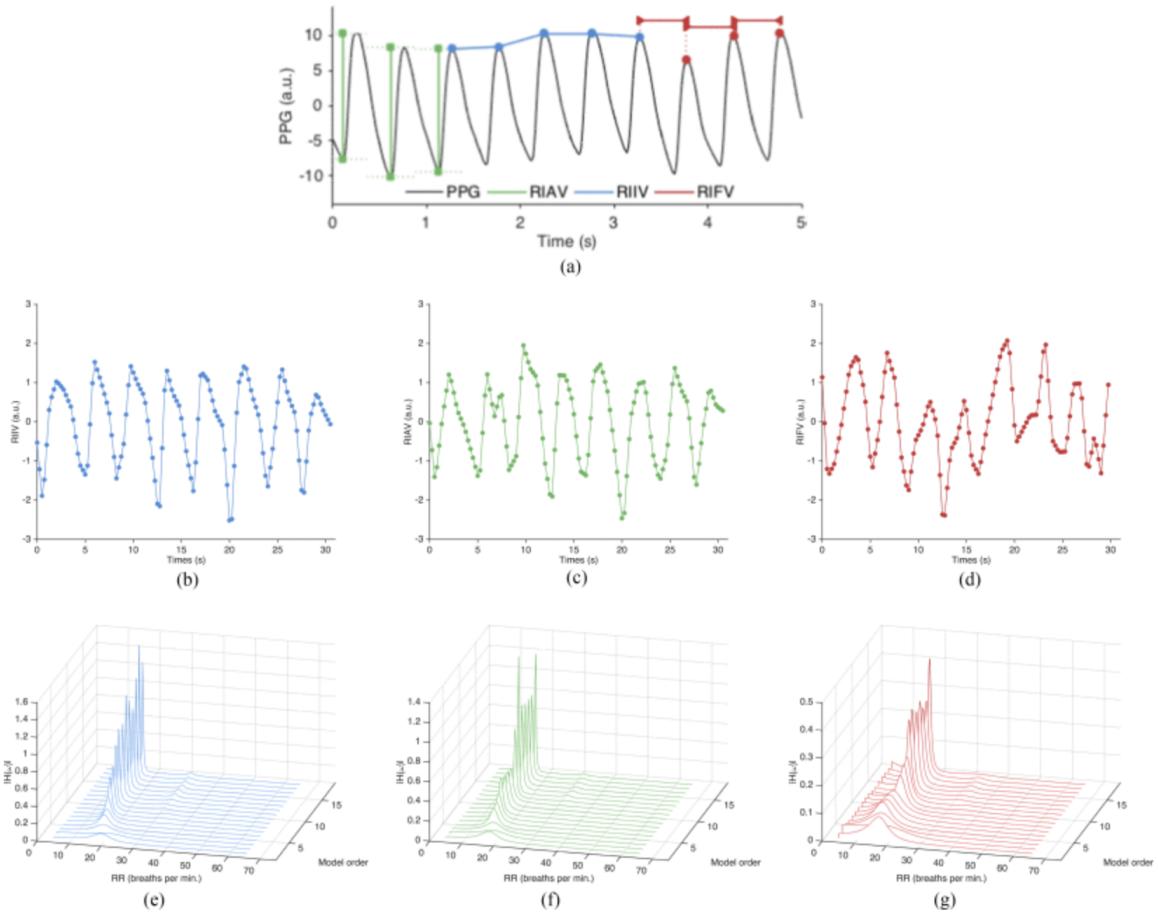


Figure 2.6: Physiosignals extraction method [25]

This study was conducted using 2 datasets which are Capnibase and BIDMC. The Capnibase dataset contains PPG recordings extracted from capnometry data from 59 children and 35 adults. The segment of patients in the Capnibase dataset contains data collected during elective surgery and routine anesthesia. The Capnibase dataset is divided into train and test sets where the test set contains 42 recordings of 8-minute duration collected from 29 children and 3 adult patients. The remaining patients were used as the training set to train the autoregressive algorithm. The second dataset which is used is the BIDMC dataset. The BIDMC dataset is extracted from the MIMIC-II data verse. This dataset consists of recordings of 53 adult patients in the form of PPG recordings and respiratory signals acquired through a conventional impedance pneumography (IP). This data is basically cured from a larger cohort of the data acquired from the patients who were admitted to medical and surgical intensive care units at the Beth Israel Deaconess Medical Center (BIDMC). This dataset consists of 53 recordings of 8-minute duration. In this study, 32s and 64s of windows were used to train and test the autoregressive algorithm. The proposed algorithm achieved fantastic results containing accuracy in the form of mean absolute error of 1.5 and 4.0 respectively for 32-s and 64-s windows.

## 2.4 Implementing respiration rate estimation methods using PPG from the Capnibase dataset

In this study, breathing rate estimation methods are explored with PPG signals using Capnibase database [14]. The researchers proposed using a combination of several different methods. The researchers used empirical mode decomposition along with the principal component analysis and wavelet analysis using 3 respiratory-induced signals which are:

1. Intensity
2. Amplitude
3. Frequency

The researchers then compared the performance of their study with six different previously implemented methods tested on the Capno-base dataset. The approach was tested on 32-s and 64-s windows. Moreover, to simulate the real-life scenario only segments of respiratory

rate annotations that are in the range of 0-0.5 Hz are used. The below-mentioned figure 2.7 explains how the algorithm is implemented in this approach.

Firstly, empirical mode decomposition is applied in the study. In EMD, signals are decomposed into intrinsic signal values or mode functions (IMF) which are defined by a process called sifting. The sifting process stops when the sift tolerance is less than 0.2 as IMF is computed in the frequency bands. The segments of the PPG signals were subjected to the EMD, and IMFs were retrieved for each segment. The frequency of respiration was then calculated for each of the collected IMFs using the SPWV distribution. The last two frequencies were eliminated because they didn't provide any information about respiration after being sorted in descending order. The final 3 BR estimations, designated IMF1, IMF2, and IMF3, were chosen from the remainder. Finally, a new method of estimating respiratory rates known as IMF Fusion (IMFF) was carried out by averaging the two values that were closest to the 3 RR that was estimated. Furthermore, for the three IMFs chosen via the EMD algorithm, the PCA was calculated. The principal component analysis (PCA) transforms the collection of IMF observations into a set of principal components (PCs), which are linearly uncorrelated variables that are arranged so that the first PC preserves most of the variation in the IMF signals. After obtaining the PCs, the first and second PCs were applied to the SPWV to estimate the BR; the resulting estimates of the respiration rate were given the names EPCA1 and EPCA2. Finally, the 2 principal components of the estimated RR are averaged, and a new RR estimation known as PC Fusion (EPCAF) was carried out.

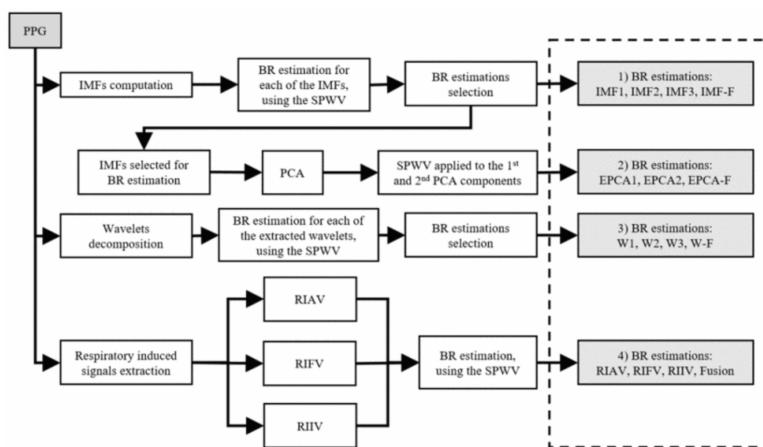


Figure 2.7: Architectural representation of implementation [14]

After EMD with PCA, wavelet decomposition is applied. This iterative method creates the wavelets decomposition tree by splitting the signal into several scales. In the current investigation, discrete wavelet decomposition using the Daubechies 1 wavelet was carried out on the PPG signal segments. Up to level 12, all information is taken from the segments. Following the application of the SPWV as previously explained in all the details that are extracted, the BR was estimated. The three highest respiration rates were designated as W1, W2, W3, and a novel RR estimation known as wavelet fusion (WF) was carried out by averaging the three estimated BRs' two closest values. After the implementation of wavelet decomposition respiratory-induced signals which are amplitude, intensity and frequency are used for the estimation of the RR.

By using Cohen's class time-frequency distribution and the Smooth-Pseudo Wigner Ville algorithm, a time-frequency analysis was carried out to extract the BR information. The quadratic time-frequency distribution's parameters were the same as those employed in principle components analysis. Using a feint sliding window of the 32s, time and frequency smoothing was performed. Signals were extracted by subtracting the mean value and resampled at 2 Hz using cubic spline interpolation before being subjected to the SPWV. The transform called Hilbert was calculated to obtain the signal analytical function. The entire spectral power of the signal was then computed for each frequency band, and the frequency band that carried the greatest amount of power was chosen as the one parallel to the RR.

The evaluation method used for this approach is the absolute error between the calculated and the reference RR. All the above-mentioned methods are applied and for each method, absolute error is calculated for 32-s and 64-s windows. The respiratory-induced signals outperformed the IMFs approach and wavelets algorithm in terms of performance. In all 64-s and 32-s signal segments, the RIAV signal outperformed competing techniques.

## 2.5 A machine learning approach to the non-invasive estimation of the Respiratory rate using a corrupted PPG signals dataset

In this study, A machine learning approach to the non-invasive estimation of the Respiratory rate using a corrupted PPG signals dataset [31]. The researchers worked on a corrupted PPG

dataset from a publicly available dataset. This dataset is first segmented into 32-s windows to be used for training and testing. The researchers then split the PPG signals into 80% training and 20% test splits along with 5-fold cross-fold validation. Before proceeding with any of the above-mentioned methodologies the PPG signals were filtered, and all the noise and motion artifacts were removed from the PPG signals.

Moreover, feature selection from the PPG dataset was done to extract meaningful features that can be used for this study, and then dimensionality reduction of the features was done to reduce the computation time and decrease the risk of overfitting.

As there are noise and motion artifacts in the PPG data which creates high noise components, the feature extraction can be hampered. The researchers implemented a low Butterworth infinite response filter with a cutoff frequency of 25 Hz to remove all the noise and motion from the PPG raw signals. As in the real-world acquisition of the PPG data, the data is very corrupted with motion and noise artifacts. Several methods can be used to remove the distortion from the PPG data however, variational mode decomposition was used to remove motion and noise from the data in this approach mentioned in the study.

VMD is strong and effective at removing noise and disturbances. In this study, five modes were identified from the physiological PPG signals based on empirical analysis. It was found that most of the motion artifact that taints the signal was present in the last mode. The initial 4 modes were used to reconstruct the PPG signal. The figure 2.8 mentioned shows the implementation of the removal of motion artifacts.

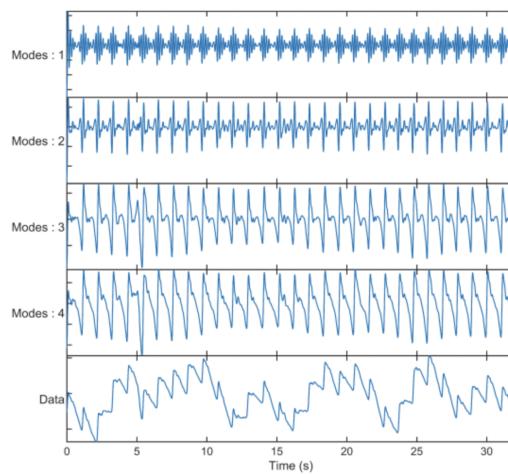


Figure 2.8: Modulation of the data [31]

The many feature types that were collected for this study are summarized in Figure 2.9. PPG waveforms are very detailed and have a lot of interesting properties. They have characteristics like the systolic peak, the waveform's foot, the pulse width, the peak-to-peak interval, etc.

Features	Definition
Systolic Peak	The amplitude of ('sys') from the PPG signal
Height of foot	The amplitude of ('amp foot') from the PPG signal
New Systolic peak	The amplitude of ('x') from the PPG signal
Systolic peak time	The time interval from the foot of the PPG signal to the systolic peak ('t1') pulse
Interval	The time interval from foot to next foot ('tpi')
Peak-to-Peak Interval	The time distance between two consecutive systolic peaks ('tpp')
t1/x	The ratio of systolic peak time to the systolic amplitude of the PPG waveform
t1/tpl	The ratio of systolic peak time to pulse interval of the PPG waveform
x/(tpi-1)	The ratio of 'x' to the difference between 'tpi' and 't1'
Rising area	Area from first foot to systolic peak('A1')
Decay area	Area from systolic peak to foot('A2')
A1/A2	The ratio from 'A1' to 'A2'
Width (25%)	The width of the PPG signal at 25% amplitude of 'x'
Width (50%)	The width of the PPG signal at 50% amplitude of 'x'
Width (75%)	The width of the PPG signal at 75% amplitude of 'x'

Figure 2.9: Features of the Data [31]

The time-domain features were recovered from the physiological PPG signal and its first and second derivatives, while statistical features were calculated using the preprocessed signal 2.10. The initial peak and first dip of the signal were its key features as seen in its derivatives. The time-domain characteristics' means, standard deviations, and variances were also calculated. This is due to the significance of these properties in capturing the distortion and modulation brought on by breathing on PPG.

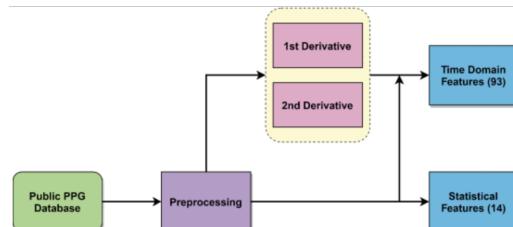


Figure 2.10: Threshold of the RR [31]

After feature extraction from the PPG dataset, feature selection is done to select the best features that correlate to the Respiratory rate estimation. By deciding to only use a portion of the estimated predictor variables to build a model, feature selection reduces the dimensionality of the data. Given limitations like feature relevance and subset size, feature selection algorithms (FSA) seek a subset of predictors that best model the tested responses. The optimal feature ranking strategy for this topic is discovered in this study following the use of 10 feature selection algorithms and numerous feature combinations. These are:

1. Fit a Gaussian Process Regression Model (FITRGP)
2. Least Absolute Shrinkage and Selection Operator (Lasso)

3. Relieff Feature Selection (RFS)
4. Feature Selection with Adaptive Structure Learning (FSASL)
5. Unsupervised Feature Selection with Ordinal Locality (UFSOL)
6. Laplacian Method (LM)
7. Unsupervised Dependence Feature Selection (UDFS)
8. Infinite Latent Feature Selection (ILFS)
9. Multi Cluster Feature Selection (mCFS)
10. Correlation Based Feature Selection (CFS)

Of the top-rank algorithms of all the 10 above-mentioned algorithms used in this study are:

1. Infinite Latent Feature Selection (ILFS)
2. Multi-Cluster Feature Selection (mCFS)
3. Correlation Based Feature Selection (CFS)

The machine learning models were trained, validated, and tested using 5-fold cross-validation. 80 percent of the 758 recordings were utilized for training, 20 percent were drawn from training samples for validation, and 20 percent were used for testing. The characteristics were then extracted. The researchers have used 3 different machine learning algorithms and 2 deep neural networks which are as follows:

1. Gaussian Process Regression (GPR)
2. Ensemble Trees
3. Support Vector Regression (SVR)
4. Artificial Neural Network (ANN)
5. Generalized Regression Neural Network (GRNN)

Machine learning algorithms were initially trained using the default settings. The hyperparameters of these machine learning and deep neural network algorithms can be tuned or optimized to improve their performance. This work, which was tweaked for 30 iterations, used Bayesian optimization.

In this study, the researchers have used mean absolute error and root mean squared error as the measurement metrics. Using hyperparameter optimization, the best ML model and feature selection procedure were adjusted to maximize performance. All other combinations were outperformed by Gaussian Process Regression (GPR) with Fit a Gaussian Process Regression Model (Fitrgp) feature selection technique, which shows RMSE, MAE, and two-standard deviation (2SD) of 2.63, 1.97, and 5.25 breaths per minute, respectively.

## Data Acquisition and proposed methodology

The impedance respiratory signal was used by two annotators to manually annotate the breaths in this dataset, which also includes signals and numbers that were taken from the much larger MIMIC II matched waveform Database [39]. The Beth Israel Deaconess Medical Center's severely ill patients provided the primary data for this study (Boston, MA, USA). Each recording's individual breaths were individually annotated by two people using the impedance respiratory signal. The dataset's 53 recordings, each 8 minutes long, include the following information:

1. Physiological signs such the electrocardiogram, impedance respiratory signal, and PPG (ECG). These are 125 Hz samples.
2. The heart rate (HR), respiratory rate (RR), and blood oxygen saturation level are examples of physiological parameters (SpO<sub>2</sub>). These are 1 Hz samples
3. Age, gender, and other fixed factors, as well as manual annotations of breaths

This dataset was initially mentioned in the cited study, where it was used to compare the effectiveness of several methods for determining respiratory rate from the photoplethysmogram (PPG) or pulse oximetry signal.

### 3.1 Data Files

The BIDMC MIMIC II [15] matched dataset is distributed into 3 formats:

1. PhysioNet uses the WFDB (WaveForm DataBase) format as its default format.
2. CSV format
3. Respiratory rate algorithms from the RRest Toolbox are compatible with the Matlab format.

In the CSV format each recording is given its own CSV file, where ## denotes the topic number:

- bidmc\_##\_Breaths.csv: Manual breath annotations
- bidmc\_##\_Fix.txt: Fixed variables
- bidmc\_##\_Signals.csv: Physiological signals
- bidmc\_##\_Numerics.csv: Physiological parameters

The WFDB format has 5 files for each recording where ## is the subject number:

- bidmc##.hea: Waveform header file
- bidmc##n.dat: Numerics data file
- bidmc##.breath: Manual breath annotations
- bidmc##.dat: Waveform data file
- bidmc##n.hea: Numerics header file

In the Matlab format the following subset of the dataset is included in the \*bidmc data.mat\* file as a single Matlab variable called \*data\*. The 53 recordings are each given the following information:

- ekg: ECG reading in lead II. Each signal is presented as a structure, with the signal values indicated by the \*v\* field and the sampling frequency indicated by the \*fs\* field.
- ppg: Photoplethysmogram signal
- ref.resp\_sig.imp: Impedance respiratory signal
- ref.breaths: two different annotators manually annotated breaths and delivered them. The signal sample numbers are represented by a vector of sample numbers.

- ref.params: Respiratory rate, heart rate, pulse rate, and blood oxygen saturation level are examples of physiological parameters. The respiratory rate, which is measured in breaths per minute, is determined by the monitor from the impedance signal.
- fix: a set of fixed variables consisting of the following: id (the subject ID and recording identifier from the MIMIC II matched waveform database), loc (the ward location), and source (the URLs from which the original data were downloaded)

## 3.2 Data preparation

In my study, MIMIC II matched dataset of CSV format was used to carry out the research [32]. Since the CSV format of the dataset was used, it contains files of 53 patients each containing files called Breaths.csv, Numerics.csv, and Signals.csv. As there are separate files of breaths, numeric, and signals data of each patient we need to collate all these separate patients' files into one single data set to be used for implementation.

To do that a file holder variable was created to store all the files of each patient. Each patient's data is matched with the corresponding values from the breaths, numeric, and signal data. Once that is implemented and all the files are collated, one single dataset set file in the CSV format is created and then stored in the system. After that, the file is loaded and made ready for data preprocessing and feature extraction.

The single dataset CSV files contain data of 53 patients consisting of 8 minutes of reading of each patient. The dataset is then checked for any anomalies and Nan values in the dataset. The Nan values are filled with the mean of the data. The file columns are renamed with standard and easy-to-understand names to be used in the implementation. The collated dataset file contains 2.7 million records and 35 features which is about 1.2GB of data. The final collated file is renamed as persons\_csvs and it contains records of all 53 patients' 8-minute PPG recordings along with other features.

## 3.3 Data preprocessing

Before proceeding with the extraction of target labels and features, the persons\_csvs file is collated with the 125Hz Singnals.csv file of each patient to bring in the data of PPG and ECG to each corresponding patient. This is mentioned in the following figures:

The structure of 1 Hz data is as:

Time (s)	HR	PULSE	SpO2
0	93	92	96
1	92	92	96
2	92	92	96
3	92	93	96
4	92	93	96

Figure 3.1: 1 Hz table

The structure of 125 Hz data is as:

Time (s)	RESP	PLETH	II	V	AVR
0.0	0.25806	0.59531	-0.058594	0.721569	0.859379
0.008	0.26393	0.59042	-0.029297	0.69608	0.69531
0.016	0.269790	0.58358	0.179690	0.7	0.45508
0.024	0.27566	0.57771	0.84375	0.32941	0.041016
0.032	0.2825	0.57283	1.3184	0.078431	-0.099609

Figure 3.2: 125 Hz table

After combining both 1 Hz and 125 Hz tables with left outer join:

Time (s)	RESP	PLETH	V	AVR	II	HR	PULSE	SpO2
0.0	0.25806	0.59531	0.721569	0.859379	-0.0585944	93	92	96
0.008	0.26393	0.59042	0.69608	0.69531	-0.029297	93	92	96
0.016	0.269790	0.58358	0.7	0.45508	0.17969	93	92	96
0.024	0.27566	0.57771	0.32941	0.041016	0.84375	93	92	96
0.032	0.2825	0.57283	0.078431	-0.099609	1.3184	93	92	96

Figure 3.3: 1 Hz and 125 Hz combined

Once the person\_csvs file is finalized and collated as shown above, now data is then used to create features. The RESP column is dropped from the dataset as it is the target label of the implementation. Once that is done minimum, maximum, skewness, and kurtosis are calculated.

The final state of the person\_csvs dataset after feature implementation is as follows in figure 3.4

Time (s)	RESP	PLETH	V	AVR	II	HR	PULSE	SpO2	RESP_Min
0.0	0.25806	0.59531	0.72157	0.85938	-0.05859	93	92	96	0.25806
0.008	0.26393	0.59042	0.69608	0.69531	-0.0293	93	92	96	0.25806
0.016	0.26979	0.58358	0.7	0.45508	0.17969	93	92	96	0.25806
0.024	0.27566	0.57771	0.32941	0.04102	0.84375	93	92	96	0.25806
0.032	0.2825	0.57283	0.07843	-0.09961	1.3184	93	92	96	0.25806

Figure 3.4: persons final dataset

Furthermore, there are some patients whose data is very irregular, unusable, and empty so the entire dataset is sent through an exception loop to find that patient and drop them from the dataset. A standard scaler is then implemented to scale the data to standard values for the implementation of the approaches mentioned in the study.

The below-mentioned figure 3.5 list all the patients that are dropped in the exception loop:

```
Ignoring person 09 due to error
Ignoring person 15 due to error
Ignoring person 30 due to error
Ignoring person 38 due to error
Ignoring person 39 due to error
Ignoring person 41 due to error
Ignoring person 47 due to error
Ignoring person 49 due to error
```

Figure 3.5: persons dropped for errors

Normalized respiratory rate data, Plethysmogram (PPG), and Electrocardiogram (ECG) data which are collected from the BIDMC hospital in the data set are used as the reference to estimate the respiratory rate in the implementation.

The 3 main signals data that are important for the implementation of this approach are visualized as follows:

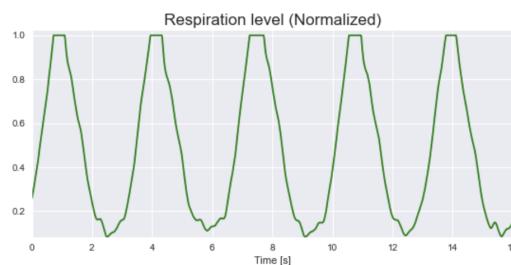


Figure 3.6: normalized RR

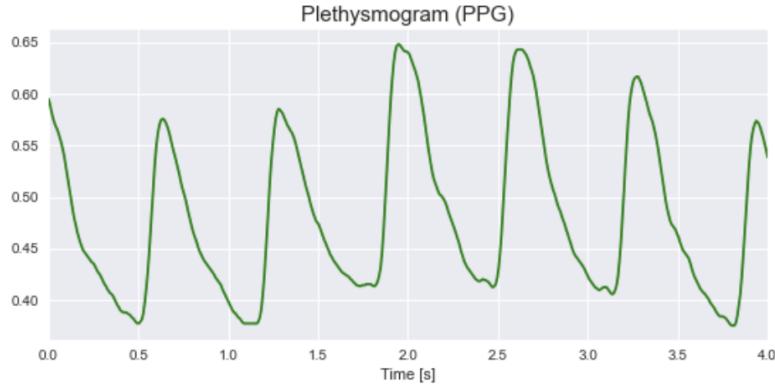


Figure 3.7: PPG signals

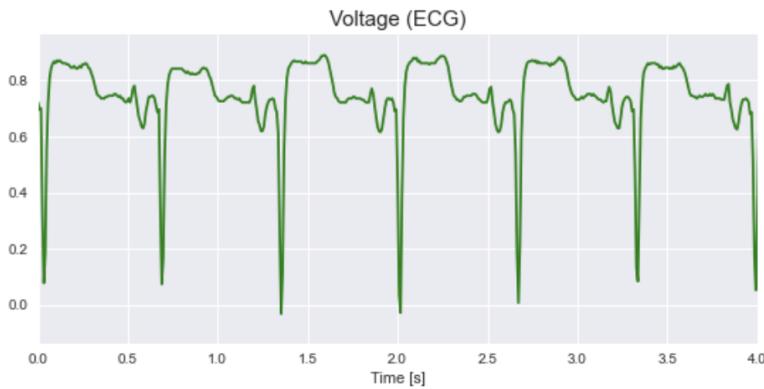


Figure 3.8: PPG signals

## 3.4 Methodology

We need to apply certain algorithms that can extract information from the signals and train on the various aspects of the data to complete the difficult task of estimating respiratory rate from PPG signals and other physiological signals such as ECG. By removing pertinent data from signals with a three-dimensional or two-dimensional data matrix, machine learning can be utilized to address this issue and help us assess the estimation of the respiratory rate. Deep neural networks (DNN), which can learn from images or signals like a human brain, and artificial neural networks, which behave similarly to human brains and are made up of many neurons, are the two branches of machine learning models that are further classified into these two categories.

With the number of hidden layers varying from one to many, as illustrated in the figure 3.9, DNN is made up of an input layer, a hidden layer, and an output layer. DNNs are

made up of millions of connected neurons and are dense like human brains. When DNN is trained, inputs from the input layer are supplied across the model in a feed-forward fashion. Applying the activation function to the weights and the associated linked neuron activates all the neurons.

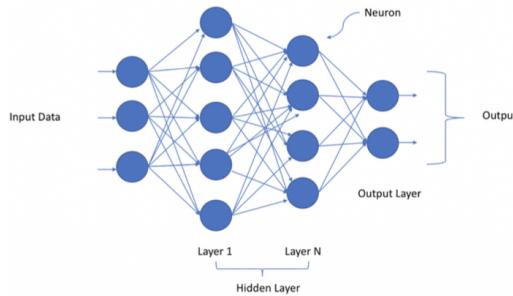


Figure 3.9: DNN architecture [16]

Up until the neural network reaches the output layer, this process is repeated. In this way, the model has been verified by computing the predicted output error and comparing it to the expected output, and the model has then through a backward process to improve the weights while taking other hyperparameters like lambda, learning rate, and momentum into account. By doing this, the total loss of the models is decreased to increase their accuracy when analyzing the data, and in the end, better weights are kept and may be used to assess the model.

Convolutional neural network (CNN) algorithms and time series algorithms like Long short-term memory (LSTM), which are specifically designed for big data and have the ability to find hidden information in signals and images subsequently learn from it, are a further subdivision of DNN. Shallow neural networks are used to construct deep learning models. They share many characteristics as a result, including layer-wise design, parameter adjustment, computing the gradient of the cost function using backpropagation, and utilizing comparable optimization approaches. The primary distinction between the two architectures is the existence of several learning convolutional layers that automatically extract features.

The machine learning algorithms that have been implemented in my study are as follows:

1. Linear Regression
2. Elastic net
3. Bayesian Ridge

4. K Neighbors Regression
5. Random Forest Regression
6. Ada Boost Regression
7. Extreme Gradient Boost Regression
8. Lasso Regression
9. Ridge Regression

The Artificial neural network and Deep neural network that is used in the study are as follows:

1. Neural Network
2. Long short-term memory (LSTM)

### **3.4.1 Linear Regression**

It is a machine learning algorithm based on supervised learning. It executes a regression operation. Regression uses independent variables to model a goal prediction value. It is mostly used to determine how variables and forecasting relate to one another. Regression models vary according to the number of independent variables they use and the type of relationship they consider between the dependent and independent variables. The dependent variable in regression has many different names. It can be referred to as a regressand, endogenous variable, criteria variable, or outcome variable. The independent variables may also be referred to as predictor variables, regressors, or exogenous variables.

The task of predicting a dependent variable's value  $y$  based on an independent variable is carried out using linear regression  $x$ . Therefore,  $x$  the input, and  $y$  the output is found to be linearly related by this regression technique output [37].

### **3.4.2 Elastic Net**

A variation of linear regression that uses the same fictitious prediction function is elastic-net regression. Overfitting is a problem for linear regression, as it cannot handle collinear data. when a dataset contains many features, even if some of them are irrelevant to the predictive

model. Due to the test set's incorrect forecast, the model becomes increasingly sophisticated or overfitting. A high variance model like this one does not generalize to the new data. To address these problems, we combine L-2 and L-1 norm regularization to gain from Ridge and Lasso simultaneously. The resulting model outperforms Lasso in terms of predictive ability. It carries out feature selection and simplifies the hypothesis [19].

### 3.4.3 Bayesian Ridge

The Bayesian technique is only a method for developing and estimating linear statistical models. When the dataset has too few or poorly dispersed data, Bayesian Regression might be quite helpful. In contrast to conventional regression techniques, where the output is only derived from a single value of each attribute, a Bayesian Regression model's output is derived from a probability distribution. The result, "y," is produced by a normal distribution where mean and variance are normalized. The goal of Bayesian Linear Regression is to identify the 'posterior' distribution for the model parameters rather than the model parameters themselves. It is anticipated that the model parameters will follow a distribution in addition to the output y [9].

### 3.4.4 KNN Regression

KNN regression is a non-parametric technique that, by averaging the data in the same neighborhood, intuitively approximates the relationship between independent variables and the continuous result. The neighborhood's size must be determined by the analyst, or it can be determined through cross-validation. The neighborhood size should be chosen to minimize the mean-squared error. The strategy is highly enticing, but as the dimension or the number of independent variables rises, it quickly becomes problematic [36].

### 3.4.5 Random Forest Regression

A supervised learning technique called Random Forest Regression leverages the ensemble learning approach for regression. The ensemble learning method combines predictions from various machine learning algorithms to provide predictions that are more accurate than those from a single model [27].

### 3.4.6 Ada Boost Regression

Boosting is an iterative method of using many algorithms simultaneously. In Ada boost regression the output of one algorithm goes into another and the output of the other algorithm goes into the next consecutive one until the end of the Ada boost collection of regression algorithms. All individual models are called weak learners while the entire Ada Boost model is called strong learners [35].

### 3.4.7 Lasso Regression

A regularization method is lasso regression. For a more accurate forecast, it is preferred over regression techniques. Shrinkage is used in this model. When data values shrink toward the mean, this is referred to as shrinkage. Simple, sparse models are encouraged by the lasso approach (i.e. models with fewer parameters). When a model exhibits a high degree of multicollinearity or when you wish to automate some steps in the model selection process, such as variable selection and parameter removal, this specific sort of regression is ideally suited. L1 regularization is employed by Lasso Regression. Because it does feature selection automatically, it is employed when there are more features [26].

### 3.4.8 Ridge Regression

Any data that exhibits multicollinearity can be analyzed using the model-tuning technique known as ridge regression. This technique carries out L2 regularization. Predicted values differ much from real values when the problem of multicollinearity arises, least-squares are unbiased, and variances are significant [23].

### 3.4.9 Extreme gradient boost regression

A class of ensemble machine learning methods known as gradient boosting can be applied to regression predictive modeling issues. Decision tree models are used to build ensembles. In order to repair the prediction mistakes caused by earlier models, trees are added one at a time to the ensemble and fitted. Boosting is a term used to describe this kind of ensemble machine learning model. Any subjective differentiable loss function and the gradient descent optimization procedure are used to fit the models. Gradient boosting gets its name from this

because as the model is fitted, it minimizes the loss gradient, much like a neural network. An effective open-source implementation of the gradient boosting technique is called Extreme Gradient Boosting, or XGBoost. As a result, XGBoost is a Python library, an open-source project, and an algorithm [7].

### 3.4.10 Neural Network

Artificial systems called neural networks were influenced by biological neural networks. These systems acquire task-specific knowledge by being exposed to a variety of datasets and examples. The concept is that the system creates distinguishing qualities from the data it receives without being pre-programmed with knowledge of these datasets. The computer models for threshold logic are the foundation of neural networks. Algorithms and mathematics are used to create threshold logic. Either the study of the brain or the use of neural networks in artificial intelligence are the foundations for neural networks. The work has improved the theory of finite automata.

A typical neural network consists of neurons, synapses connections between them weights, biases, propagation function, and a learning rule. Predecessor neurons with an activation threshold, activation function, and output function will provide input to neurons. It is determined by connections, weights, and biases in how a neuron sends output to another neuron. Propagation calculates the input, produces the output, and adds the weight and the function of the predecessor neurons. The change of the free parameters, such as weights and bias, is essentially what is meant by neural network learning. The network's variable weights and thresholds are altered by the learning rule [2].

### 3.4.11 Long-short-term memory (LSTM)

Recurrent neural networks (RNNs) are extended by LSTM networks, which were primarily developed to address RNN failure scenarios. When we talk about RNN, it is a network that operates on the current input while considering the prior output and temporarily storing it in memory i.e., short-term memory. The most well-liked uses of this technology are in the areas of non-Markovian control, speech processing, and musical composition. RNNs do, however, have shortcomings. In the beginning, it is unable to keep data for a longer period. Sometimes, to forecast the current output, a reference to specific data that was saved a long time ago is

needed. RNNs, however, are utterly unable to manage such "long-term dependencies".

Furthermore, there is no finer control over how much of the past should be "lost" and how much of the context should be carried forward. Exploding and vanishing gradients, which happen when a network is being trained via backtracking, is another problem with RNNs. Long Short-Term Memory (LSTM) was introduced as a result. The training model is left unmodified, and the vanishing gradient problem has been nearly eliminated. LSTMs are used to bridge long-time lags in some issues, and they can also deal with noise, distributed representations, and continuous values.

Moreover, unlike the hidden Markov model, which requires keeping a limited number of prior states, LSTMs do not have this requirement. We can choose from a wide range of LSTM parameters, including learning rates and input and output biases. Thus, there is no need for precise modifications. With LSTMs, updating each weight is simpler than with Back Propagation Through Time (BPTT), reducing complexity to  $O(1)$ .

The fundamental distinction between the LSTM and RNN architectures is that the LSTM's hidden layer is a gated unit or gated cell. It is made up of four layers that interact with one another to create both the cell state and the output of that cell. The following hidden layer receives these two items after that. LSTMs are made up of three logistic sigmoid gates and one tanh layer, in contrast to RNNs which only feature a single tanh layer. To restrict the amount of information that is passed through the cell, gates have been implemented. They choose which information will be needed by the following cell and which should be ignored. The output is typically between 0 and 1, with 0 denoting "reject all" and 1 denoting "include all." The figure 3.10 shows the architecture of the LSTM model [20].

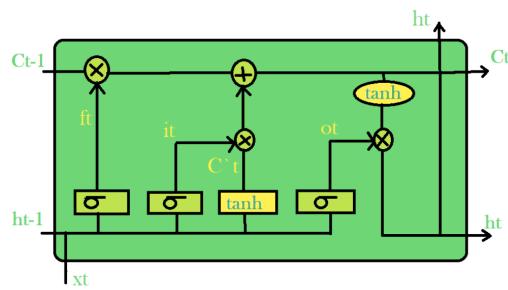


Figure 3.10: LSTM model [20]

## 3.5 Evaluation Metrics

Mean Absolute Error (MAE): Without considering their direction, MAE calculates the average size of mistakes in a set of forecasts. All individual differences are equally weighted in the test sample's average of the absolute disparities between prediction and observation. The formula for mean absolute error is shown in the figure 3.11

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

Figure 3.11: MAE equation [41]

Root mean squared error (RMSE) is a scoring formula for quadratic equations that also calculates the average error magnitude. It is the average of the squared discrepancies between predicted results and actual observations. The formula for root mean absolute error is shown in the figure 3.12

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

Figure 3.12: RMSE equation [41]

There are similarities between how mean absolute model prediction error (MAE) and root mean square error (RMSE) are expressed. Both measures are unaffected by the direction of mistakes and have a range of 0 to 1. Since these are negatively orientated scores, lower values are preferable.

Differences: Calculating the mean squared errors' square root has some intriguing ramifications for RMSE. The RMSE lends comparatively significant weight to large errors since the errors are squared before they are averaged. Therefore, when huge mistakes are particularly undesired, the RMSE should be more helpful. The three tables 3.13 below demonstrate situations in which RMSE rises as the variance connected to the frequency distribution of error magnitudes also rises while MAE remains constant.

If being off by 10 is more than twice as bad as being off by 5, for example, RMSE has the advantage of penalizing huge errors more severely. However, MAE is more suitable if being

off by 10 is merely twice as severe as being off by 5.

CASE 1: Evenly distributed errors			CASE 2: Small variance in errors			CASE 3: Large error outlier		
ID	Error	Error <sup>2</sup>	ID	Error	Error <sup>2</sup>	ID	Error	Error <sup>2</sup>
1	2	4	1	1	1	1	0	0
2	2	4	2	1	1	2	0	0
3	2	4	3	1	1	3	0	0
4	2	4	4	1	1	4	0	0
5	2	4	5	1	1	5	0	0
6	2	4	6	3	9	6	0	0
7	2	4	7	3	9	7	0	0
8	2	4	8	3	9	8	0	0
9	2	4	9	3	9	9	0	0
10	2	4	10	3	9	10	20	400

MAE	RMSE
2.000	2.000

MAE	RMSE
2.000	2.236

MAE	RMSE
2.000	6.325

MAE and RMSE for cases of increasing error variance

Figure 3.13: RMSE table [41]

MAE is unquestionably the winner when it comes to interpretation. In addition to describing an average error, RMSE has other consequences that are more challenging to decipher and comprehend.

In my study, I have used both measurement metrics to measure the performance of all the algorithms I have implemented but have taken Mean absolute error as the main metric for the measurement of the performance of the algorithms [41].

---

## Architecture

To estimate respiratory rate from the physiological signal gathered from the BIDMC hospital i.e., PPG. In my study, I have implemented 3 different approaches each of a similar domain. These are:

1. Estimation of RR through Supervised Machine learning algorithms in the Linear regression domain
2. Estimation of RR through the Recurrent Neural Network approach called Long-short term memory
3. Estimation of RR through a Dense neural network called Artificial neural network

### 4.1 Machine learning models for RR estimation

As mentioned in the previous chapter, I have used 9 machine-learning algorithms and for each of the algorithms, linear regression type is used in the supervised machine-learning domain. In order to implement all these 9 previously mentioned machine learning algorithms the RESP column in the persons\_csvs dataset file is dropped and labeled as a target label. The dataset is preprocessed as mentioned in the abovementioned chapter and is scaled to normalized values in order to be used in the machine learning algorithms. Since all of these algorithms belong to the linear regression domain, they need features as input and a target

variable output to learn while the model fits. The following hyperparameters were tuned to implement these 9 machine learning models:

1. Train-test split of 80% training and 20% test data
2. Standard scalarization on 27 features extracted from PPG, ECG, and normalized RR
3. Cross-fold validation with a random state of 12
4. 32-s window segmentation (68125 data points)
5. Mean absolute error, root mean squared error, and R2 score as measurement metrics

## 4.2 RNN model for respiratory rate estimation

As mentioned in the previous chapter I have used a Recurrent neural network approach for the estimation of RR. The RR approach that I have implemented in this paper is called LSTM. In order to feed the data to the data-intensive LSTM algorithm the data first need to be adjusted according to the need of the algorithm. Firstly, the ground truth label which is RESP is dropped from the table and stored as the label value to be used for training the algorithm. Here the data is augmented to create more features so there are 53 features now in the dataset. The features are augmented using the min, max, mean, skewness, and kurtosis. First order derivate is implemented on the dataset and threshold is set to limit the breaths amplitude. The following parameter and hyperparameter tuning are done to implement the LSTM:

1. Empty values are filled by mean
2. Features and Label array is reshaped
3. Cross-validation split is implemented
4. Train-test split is implemented with 40% testing and 60% training
5. A 32-s window is taken (69058 data points)
6. A breathing rate threshold of 5 is chosen following the previous research
7. Three Train split of the selected data is implemented to send into the LSTM algorithm one by one

8. Feature scaling is done with PCA to standardize the data
9. Two LSTM layers with 128 cells each
10. 20 Epochs
11. 128 Batch size
12. 0.0001 learning rate
13. 0.3 dropping probability
14. Gradient descent as optimizer and mean squared error as cost function

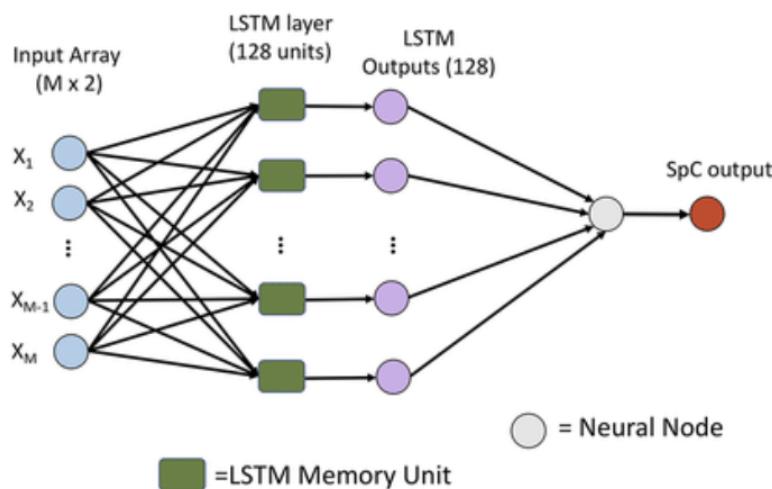


Figure 4.1: LSTM architecture

## 4.3 ANN model for respiratory rate estimation

As mentioned in the previous chapter, I have used an Artificial neural network for the estimation of the respiratory rate. The ANN that I have used in my study is a dense neural network. Just like the previously mentioned architectures, in this approach, I have also extracted the target label from the data. The remaining 27 features extracted from the biosignals such as PPG, ECG, and normalized RR is used as the features to be inserted in the DNN that I have implemented in my study. The following parameter and hyperparameter tuning are done to implement the DNN:

1. Train-test split of 80% training and 20% testing

2. Cross-validation split of 20 random states
3. 32-s window segmentation (68125 data points)
4. Input dimension of 27 as there are 27 features
5. Two hidden layers, one with 128 neurons and the second with 64 neurons
6. The hidden layer with reLU activation function
7. The output layer with linear activation function
8. Loss function as mean squared error and Optimizer as Nadam
9. Measurement metric as the mean absolute error

---

## Experimental results and analysis

The experiments and general results from the dataset proposed in the prior chapters are covered in this chapter. All the findings are discussed here in order to solve the issue of estimating respiratory rate, and numerous approaches and procedures from a wide range have been employed. The code for the respiratory rate estimation problem can be found at this [link](#). These analyses were carried out using the following programming languages, operating systems, and software applications:

1. Operating system: macOS Big Sur (1.4 GHz Quad-Core Intel Core i5)
2. Programming Languages and libraries: Python, Numpy, Pandas, matplotlib, Keras and Tensorflow v1
3. IDE: Jupyter Notebook and Kaggle notebook

### 5.1 Experimental analysis

The results that are discussed in this chapter are from 9 machine learning linear regression models, a Recurrent neural network model called LSTM and an Artificial neural network model called a dense neural network. The results analysis for each model are as follows:

### 5.1.1 Machine learning models analysis of results

Once the data is loaded from the MIMIC II dataset, preprocessed, and fine-tuned with standardization and other data cleaning and preprocessing methods it is almost ready for model fitting. However, before fitting the model, features are to be extracted and made ready for implementation.

Since I have implemented 9 machine learning models in the linear regression domain, all of these models need the same type of input as features and a target label to train the data. Once the data is ready after all the preprocessing and feature extraction approaches it is now ready to be sent into the model. There are 2.7 million records from 53 patients consisting of 8-minute readings from each patient. In my study, I have implemented a 32-s window for the implementation which leaves me with 68125 records to be used as input for the 9 machine learning models. All models are trained with metrics called RMSE, MAE, and R2 scores along with the time they took to learn and fit the model.

The following figure 5.1 shows the results of each of the models for the estimation of the respiratory rate.

	model	RMSE	MAE	R2 score	time
0	LinearRegression	0.327985	0.272190	0.372744	0.090463
1	BayesianRidge	0.327991	0.272236	0.372720	0.069433
2	AdaBoostR	0.321011	0.285015	0.399136	2.195367
3	XGBR	0.155937	0.102330	0.858213	2.530214
4	LassoR	0.414127	0.381903	-0.000010	0.025485
5	RidgeR	0.328017	0.272396	0.372622	0.022823
6	KNN-R	0.140933	0.060027	0.884185	13.760796
7	RF-R	0.108051	0.048205	0.931924	28.534862
8	ElasticNet	0.414127	0.381903	-0.000010	0.028881

Figure 5.1: ML algorithms output

The best-performing algorithm among all the 9 mentioned algorithms is Random Forest regression as the MAE is the lowest among all the mentioned algorithms i.e., 0.04, and the lowest RMSE is 0.108. However, Random Forest regression took the most computational time among all the mentioned machine learning algorithms.

The validation of the well-fit model i.e., the random forest can be seen in the below-

mentioned figure 5.2. In addition to performance indicators alone, it's beneficial to assess how effectively the model has been fitted. Here, we can observe how errors were distributed over the test set and the train set. It's okay if the frequency count has a little different scale because this is proportional to the size variances between the train set and the test set.

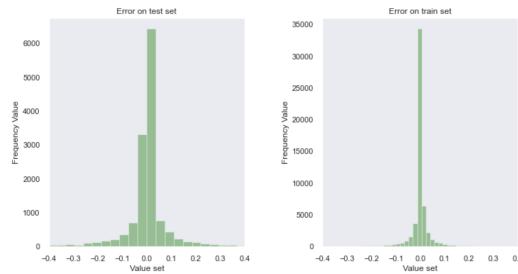


Figure 5.2: Errors in train and test set

One of the best things about the random forest library is that it enables us to provide an output that describes the level of feature relevance. The 5.3 and 5.4 show the feature importance of the features used in these machine-learning algorithms.

```
Ranking of Features according to their importance:
1. AVR_Mean (0.338486)
2. PLETH (0.117453)
3. V_Mean (0.064629)
4. V (0.047077)
5. II_Mean (0.045718)
6. AVR (0.044352)
7. II (0.035381)
8. AVR_Max (0.030928)
9. AVR_Min (0.027517)
10. II_Max (0.024671)
11. II_Kurt (0.018784)
12. PLETH_Min (0.018254)
13. PLETH_Max (0.017943)
14. PLETH_Skw (0.015920)
15. PULSE (0.015420)
16. II_Skw (0.014870)
17. V_Skw (0.014574)
18. AVR_Kurt (0.014511)
19. V_Max (0.014360)
20. PLETH_Mean (0.013111)
21. PLETH_Kurt (0.012502)
22. V_Min (0.012319)
23. HR (0.012028)
24. V_Kurt (0.009287)
25. II_Min (0.008061)
26. AVR_Skw (0.007251)
27. SpO2 (0.004594)
```

Figure 5.3: Features valuation

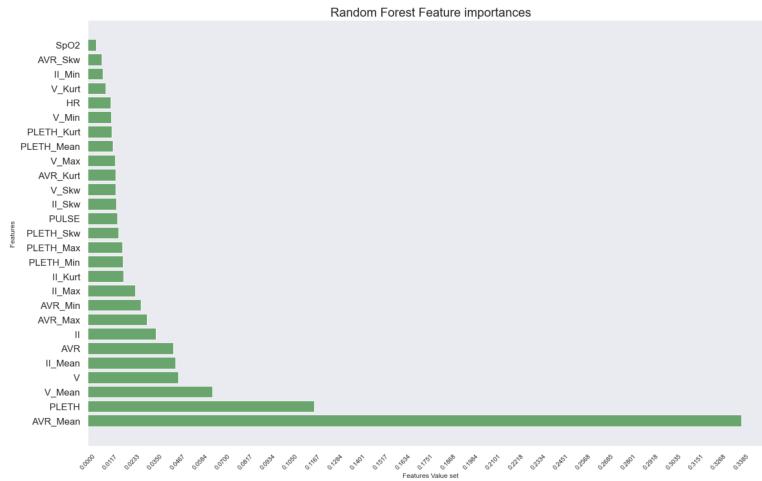


Figure 5.4: Feature importance

Additionally, it was discovered that even without the plethysmogram data, we were still able to predict respiratory rate with up to 80% accuracy.

We can deduce a few things. We can anticipate a person's respiratory rate with 90% accuracy using information from their ECG and plethysmogram. We can accurately estimate anyone's respiratory rate if we train using data from several people.

### 5.1.2 LSTM model analysis of results

The LSTM model is a type of RNN model therefore the data needs to be manipulated accordingly before it is sent to the model to be trained and fitted. Firstly, features are augmented to 52 features and the target label is extracted. 52 features are supplied as the input for an Exploratory Data Analysis on this data.

Below is an illustration 5.5 of a portion of the respiratory curve:

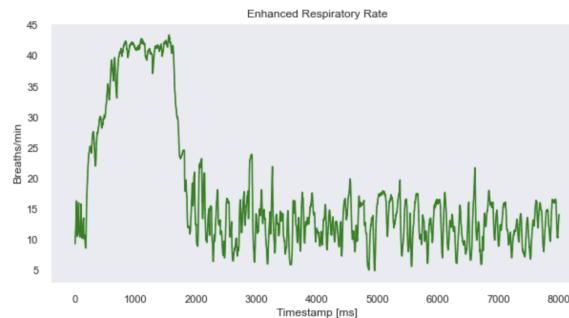


Figure 5.5: Enhanced RR

The full respiratory rate obtained from the data is displayed throughout time as follows 5.6:

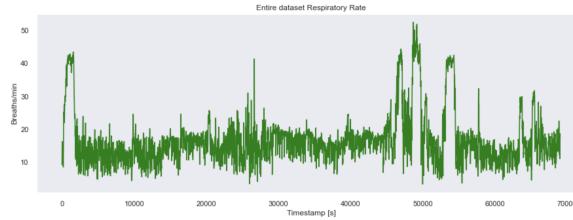


Figure 5.6: Entire dataset RR

We note that the dataset contains spikes brought on by erroneous measurements. The amount of variance in successive time steps is then measured using the absolute first-order derivative after the noise has been removed. This is how the absolute first-order derivative looks 5.7:

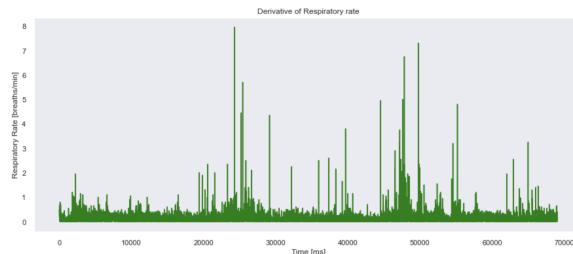


Figure 5.7: Derivative of RR

The rise in the respiratory rate at various timesteps is unreasonable, as can be seen in the figure 5.8. We established a threshold on the value change in order to deduct these amounts. If the rise exceeds this limit, the user may have substantially altered the operation or the sensor reading may have picked up noise. Below, with a threshold value of 5, is the first-order derivative that has been thresholded.

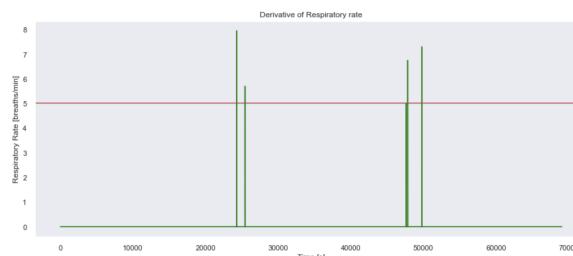


Figure 5.8: Thresholded Derivative of RR

I then divided our training dataset into three separate sets using the timestamps. I will train the model separately for every set. Below are the three sets [5.9](#)

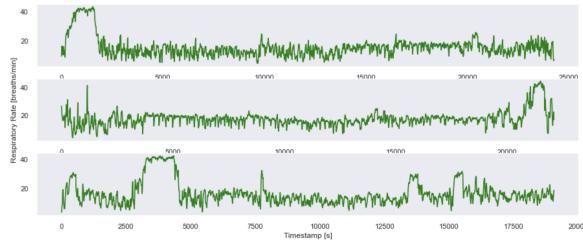


Figure 5.9: RR rate data

There are a total of 52 different ways to characterize the respiratory rate value. Dimensionality Reduction is used to lower the number of computations because we do not need to employ all 52 characteristics. To determine the variances of each of the 52 components, we first run PCA while keeping track of all 52 characteristics. The sum of the variances for each separate component makes up the total variance. The proportion of a principal component's variation to the total variance is known as the fraction of variance explained. We determine each component's contribution using the "explained variance ratio" parameter offered by Sklearn, and then we select the top k components that best describe the data.

For time series forecasting, we employ LSTMs. The data is fed in order. Assuming that the sequence length is L and that the current time is T, the features from time steps (T-1) to (T-L) will be the model's input, while the data at time T will be its output. Given that a result is a real number, the issue at hand is a regression problem. Thus, we employ the gradient descent optimizer for convergence and the MAE as the loss. The below-mentioned figure [5.11](#) shows the training and validation loss of the LSTM algorithm.

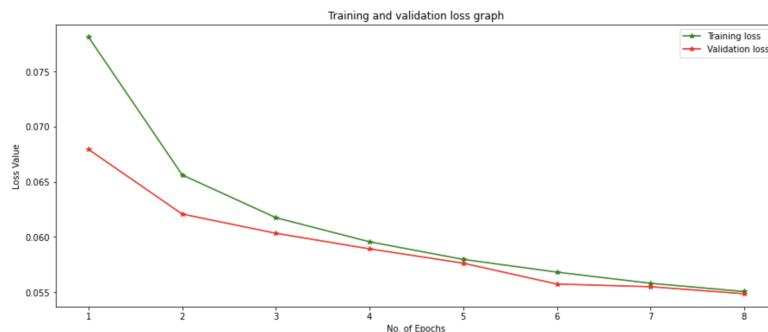


Figure 5.10: Training and Validation loss

The final MAE of the LSTM approach is 3.860. The training and validation loss shows that the algorithm is fitting properly and is being trained exceptionally. The below-mentioned figure 5.11 illustrates how the test data performed as compared to the target label.

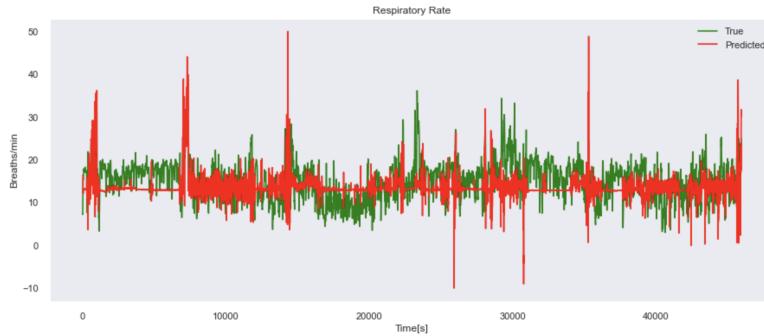


Figure 5.11: Training and Validation loss

### 5.1.3 DNN analysis of results

Just like the previously mentioned approaches, for the neural network approach the data is preprocessed, clean, and scaled and features were extracted to make a final dataset ready to be sent into the model and fitted. Here 27 features are used while 1 label is selected as the target value.

The Neural network is made up of 27 input neurons, and 2 hidden layers with 128 and 64 neurons respectively. The hidden layers have reLU as an activation function while the output neuron has a linear activation function. The model is trained with MAE loss function and Nadam optimizer. The following 2 figures i.e., 5.12 and 5.13 show:

1. Training and validation loss graph
2. Training and validation MAE graph

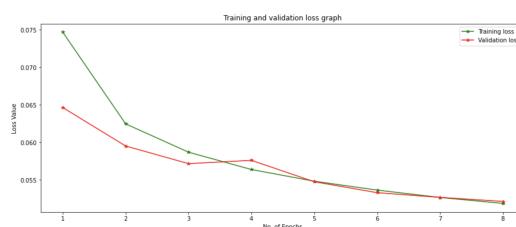


Figure 5.12: Training and Validation loss graph

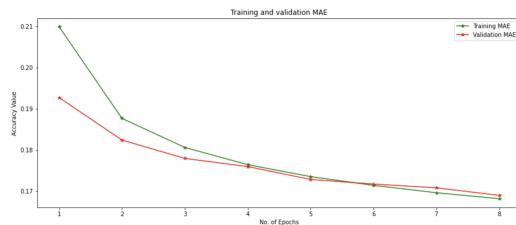


Figure 5.13: Training and Validation loss MAE

The mean absolute error using the neural network approach is 0.169. The training and validation loss and MAE graphs show the model is fitting properly and showing exceptional results.

## 5.2 Comparison of results

The three approaches that are used in this study are the machine learning approach, the Recurrent neural network approach, and the Artificial neural network approach. Since MAE is the main metric used in my approach for the evaluation of the algorithms the following table shows the table 5.1 of the result of the 3 main approaches used in the study. From the results, we can evaluate that the machine learning approach outperformed both deep learning approaches. However, the purpose of my study is to find out the implementation of deep learning algorithms and their viability in the field of medical science. Therefore, the ANN algorithm produced an exceptional result with an MAE of 0.169 while the LSTM algorithm produced an MAE of 3.860. This is still a viable result given that LSTM is a data-intensive algorithm and if big data is given to this algorithm, it could improve the results drastically.

Model	MAE
Random Forest Regression	0.048
LSTM	3.860
DNN	0.169

Table 5.1: Results comparison

---

## Conclusion and Future Works

To estimate respiratory rate from physiological signals such as PPG and ECG there is continuous work being conducted in this domain. However, every related research in this domain leads to the implementation of a new and better approach. In my study, I have demonstrated and implemented the use of Machine learning algorithms, Recurrent neural network algorithms, and Artificial neural network algorithms. Each algorithm has its own different approach to training on the given data, but each algorithm demonstrated the capability of the use of AI in this domain of the medical field. Researchers are continuously working on different approaches to estimate the RR, however, the deep learning approach that I have used is a novel and new contribution to the existing contributions in the field.

### 6.1 future works

While researching and working on this domain of the medical field. I have gone through different approaches. One approach which can be used along with my implementation is the signal processing data directly into the algorithms that I have constructed. In other words, raw PPG signals can be directly fed into the algorithms that I have used in my study and would let the algorithms learn the rules and predict features themselves. This way the signals just need to be processed before being sent into the algorithms through methods such as Fast Fourier transformation and bandpass filters. This approach can be extremely useful as it can clean and preprocess the data even further which can lead to a further reduction in errors and

noise in the PPG signals. In short, this approach can be implemented in my study, and based on that the estimation of the respiratory rate can be made even more robust and accurate



---

## Code link

The mentioned link takes the user to the GitLab repository containing the code

[https://cseegit.essex.ac.uk/21-22-ce901-sl/21-22\\_CE901-SL\\_azim\\_safi-s/-/blob/master/Respiratory\\_rate\\_estimation\\_-\\_MSc\\_Project-Safi\\_Sohail\\_Azim.ipynb](https://cseegit.essex.ac.uk/21-22-ce901-sl/21-22_CE901-SL_azim_safi-s/-/blob/master/Respiratory_rate_estimation_-_MSc_Project-Safi_Sohail_Azim.ipynb)

---

## Bibliography

- [1] C. A. Alvarez, C. A. Clark, S. Zhang, E. A. Halm, J. J. Shannon, C. E. Girod, L. Cooper, and R. Amarasingham. Predicting out of intensive care unit cardiopulmonary arrest or death using electronic medical record data. *BMC medical informatics and decision making*, 13(1):1–11, 2013.
- [2] U. Anders and O. Korn. Model selection in neural networks. *Neural networks*, 12(2):309–323, 1999.
- [3] D. Bian, P. Mehta, and N. Selvaraj. Respiratory rate estimation using ppg: a deep learning approach. In *2020 42nd annual international conference of the IEEE engineering in Medicine & Biology Society (EMBC)*, pages 5948–5952. IEEE, 2020.
- [4] A. J. Buda, M. R. Pinsky, N. B. Ingels Jr, G. T. Daughters, E. B. Stinson, and E. L. Alderman. Effect of intrathoracic pressure on left ventricular performance. *New England Journal of Medicine*, 301(9):453–459, 1979.
- [5] P. H. Charlton, D. A. Birrenkott, T. Bonnici, M. A. Pimentel, A. E. Johnson, J. Alastruey, L. Tarassenko, P. J. Watkinson, R. Beale, and D. A. Clifton. Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review. *IEEE reviews in biomedical engineering*, 11:2–20, 2017.
- [6] P. H. Charlton, D. A. Birrenkott, T. Bonnici, M. A. Pimentel, A. E. Johnson, J. Alastruey, L. Tarassenko, P. J. Watkinson, R. Beale, and D. A. Clifton. Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review. *IEEE reviews in biomedical engineering*, 11:2–20, 2017.
- [7] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

- [8] L. Clifton, D. A. Clifton, M. A. Pimentel, P. J. Watkinson, and L. Tarassenko. Predictive monitoring of mobile patients by combining clinical observations with data from wearable sensors. *IEEE journal of biomedical and health informatics*, 18(3):722–730, 2013.
- [9] A. Efendi and Effrihan. A simulation study on bayesian ridge regression models for several collinearity levels. In *AIP conference proceedings*, volume 1913, page 020031. AIP Publishing LLC, 2017.
- [10] C. for Clinical Practice at NICE (UK) et al. Acutely ill patients in hospital: recognition of and response to acute illness in adults in hospital. *London: National Institute for Health and Clinical Excellence (UK)*, 2007.
- [11] A. Guyton. Renal function curve—a key to understanding the pathogenesis of hypertension. *Hypertension*, 10(1):1–6, 1987.
- [12] F. M. Hardinge, H. Rutter, C. Velardo, C. Toms, V. Williams, L. Tarassenko, and A. Farmer. Using a mobile health application to support self-management in copd-development of alert thresholds derived from variability in self-reported and measured clinical variables. In *A37. COST, TECHNOLOGY, AND COMPARATIVE EFFECTIVENESS*, pages A1396–A1396. American Thoracic Society, 2014.
- [13] W. Karlen, S. Raman, J. M. Ansermino, and G. A. Dumont. Multiparameter respiratory rate estimation from the photoplethysmogram. *IEEE Transactions on Biomedical Engineering*, 60(7):1946–1953, 2013.
- [14] R. Lazazzera and G. Carrault. Breathing rate estimation methods from ppg signals, on capnibase database. In *2020 Computing in Cardiology*, pages 1–4. IEEE, 2020.
- [15] J. Lee, D. J. Scott, M. Villarroel, G. D. Clifford, M. Saeed, and R. G. Mark. Open-access mimic-ii database for intensive care research. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 8315–8318. IEEE, 2011.
- [16] T. Lesort. Continual learning: Tackling catastrophic forgetting in deep neural networks with replay processes. *arXiv preprint arXiv:2007.00487*, 2020.
- [17] B. N. Li, M. C. Dong, and M. I. Vai. On an automatic delineator for arterial blood pressure waveforms. *Biomedical Signal Processing and Control*, 5(1):76–81, 2010.

- [18] J. Li, J. Jin, X. Chen, W. Sun, and P. Guo. Comparison of respiratory-induced variations in photoplethysmographic signals. *Physiological measurement*, 31(3):415, 2010.
- [19] Q. Li and N. Lin. The bayesian elastic net. *Bayesian analysis*, 5(1):151–170, 2010.
- [20] Y. Liu, C. Sun, L. Lin, and X. Wang. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*, 2016.
- [21] P. B. Lovett, J. M. Buchwald, K. Stürmann, and P. Bijur. The vexatious vital: neither clinical measurements by nurses nor an electronic monitor provides accurate measurements of respiratory rate in triage. *Annals of emergency medicine*, 45(1):68–76, 2005.
- [22] P. D. Mannheimer. The light–tissue interaction of pulse oximetry. *Anesthesia & Analgesia*, 105(6):S10–S17, 2007.
- [23] D. W. Marquardt and R. D. Snee. Ridge regression in practice. *The American Statistician*, 29(1):3–20, 1975.
- [24] S. I. Nikolenko. *Synthetic data for deep learning*, volume 174. Springer, 2021.
- [25] M. A. Pimentel, A. E. Johnson, P. H. Charlton, D. Birrenkott, P. J. Watkinson, L. Tarassenko, and D. A. Clifton. Toward a robust estimation of respiratory rate from pulse oximeters. *IEEE Transactions on Biomedical Engineering*, 64(8):1914–1923, 2016.
- [26] J. Ranstam and J. Cook. Lasso regression. *Journal of British Surgery*, 105(10):1348–1348, 2018.
- [27] M. R. Segal. Machine learning benchmarks and random forest regression. 2004.
- [28] S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati. A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology. In *2017 IEEE EMBS international conference on biomedical & health informatics (BHI)*, pages 141–144. IEEE, 2017.
- [29] K. H. Shelley. Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate. *Anesthesia & Analgesia*, 105(6):S31–S36, 2007.
- [30] K. H. Shelley. Photoplethysmography: beyond the calculation of arterial oxygen saturation and heart rate. *Anesthesia & Analgesia*, 105(6):S31–S36, 2007.

- [31] M. N. I. Shuzan, M. H. Chowdhury, M. S. Hossain, M. E. Chowdhury, M. B. I. Reaz, M. M. Uddin, A. Khandakar, Z. B. Mahbub, and S. H. M. Ali. A novel non-invasive estimation of respiration rate from motion corrupted photoplethysmograph signal using machine learning model. *IEEE Access*, 9:96775–96790, 2021.
- [32] M. N. I. Shuzan, M. H. Chowdhury, M. S. Hossain, M. E. Chowdhury, M. B. I. Reaz, M. M. Uddin, A. Khandakar, Z. B. Mahbub, and S. H. M. Ali. A novel non-invasive estimation of respiration rate from motion corrupted photoplethysmograph signal using machine learning model. *IEEE Access*, 9:96775–96790, 2021.
- [33] G. B. Smith, D. R. Prytherch, P. E. Schmidt, and P. I. Featherstone. Review and performance evaluation of aggregate weighted âtrack and triggerâsystems. *Resuscitation*, 77(2):170–179, 2008.
- [34] G. B. Smith, D. R. Prytherch, P. E. Schmidt, and P. I. Featherstone. Review and performance evaluation of aggregate weighted âtrack and triggerâsystems. *Resuscitation*, 77(2):170–179, 2008.
- [35] D. P. Solomatine and D. L. Shrestha. Adaboost. rt: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 1163–1168. IEEE, 2004.
- [36] Y. Song, J. Liang, J. Lu, and X. Zhao. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251:26–34, 2017.
- [37] X. Su, X. Yan, and C.-L. Tsai. Linear regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(3):275–294, 2012.
- [38] L. Tarassenko, D. A. Clifton, M. R. Pinsky, M. T. Hravnak, J. R. Woods, and P. J. Watkinson. Centile-based early warning scores derived from statistical distributions of vital signs. *Resuscitation*, 82(8):1013–1018, 2011.
- [39] S. Wang, M. B. McDermott, G. Chauhan, M. Ghassemi, M. C. Hughes, and T. Naumann. Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM conference on health, inference, and learning*, pages 222–235, 2020.

- [40] T. M. Wardlaw, E. W. Johansson, and M. J. Hodge. *Pneumonia: the forgotten killer of children*. Unicef, 2006.
- [41] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.