

A New Real-time Lossless Data Compression Algorithm for ECG and PPG Signals

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

21-09-2021 / 23-09-2021

CITATION

BANERJEE, SOUMYENDU; Singh, Girish Kumar (2021): A New Real-time Lossless Data Compression Algorithm for ECG and PPG Signals. TechRxiv. Preprint. <https://doi.org/10.36227/techrxiv.16651732.v1>

DOI

[10.36227/techrxiv.16651732.v1](https://doi.org/10.36227/techrxiv.16651732.v1)

A New Real-time Lossless Data Compression Algorithm for ECG and PPG Signals

Soumyendu Banerjee, *Graduate Student Member IEEE*, and Girish Kumar Singh

Abstract—Objective: Data compression is a useful process in tele-monitoring applications, in which lesser number of bits are needed to represent the same data. In this work, a run-time lossless compression of single channel Electrocardiogram (ECG) and Photoplethysmogram (PPG) signal is proposed, maintaining all dominant features. **Methods:** The single channel data are first quantized using optimal quantization level, so that lesser number of bits are needed to represent it, maintaining low quantization error. Then second order delta encoding and run-length encoding (RLE) based data compression are proposed in this work. A new approach of using ‘buffer array’ along with RLE is also introduced, so that minimum bits are needed to store. **Results:** This algorithm was tested on various single lead ECG and PPG signals available in Physionet. An average compression ratio (CR) was achieved of 6.52, 3.82, and 2.49 for 547 PTBDB ECG records, 48 MITDB ECG records, and 53 MIMIC-II PPG records, respectively. This algorithm was also performed on single channel ECG, collected from 10 healthy volunteers using AD8232 ECG module, with 125 Hz sampling frequency and 10-bit data resolution, which resulted in average CR of 2.34. **Discussion:** This algorithm was also performed on a smartphone device that provided user-friendly operation. The low computational complications and standalone operation of data collection, compression, and transmission encouraged its implementation for run-time operation. **Significance:** A comparative study of proposed work with previously published works proved this fact that this algorithm provided better performance in the area of run-time patient health monitoring applications.

Index Terms—Buffer array, Lossless Compression, Raspberry Pi, Run length encoding, Second order delta encoding

I. INTRODUCTION

A. Overview and motivation

BIOELECTRIC signals are generated due to continuous functioning of various physiological processes in human body. Among these processes, cardiovascular system (CVS) is one of the most important biological systems, which is caused due to continuous contraction and relaxation of human heart. Electrocardiogram (ECG) and Photoplethysmogram (PPG) are the most important bioelectric signals that are generated in CVS. The ECG (both single and multichannel) is a

representation of full cardiac cycle, measured by placing electrodes at specific positions of human body [1]. The PPG is a single channel measure of blood volume in each cardiac cycle, measured using optical radiation, coming through skin (mainly in fingertip) [2]. Cardiovascular diseases are one of the most common factors of death worldwide, as mentioned by World Health Organization (WHO) [3]. Therefore, accurate measurement, storing, and transmission of these signals are very necessary. Now, long term data storage needs huge memory space, affecting the storage efficiency and transmission channel performance. Data compression is only way to overcome these problems. There are two types of compression processes, viz. lossy and lossless. In lossy compression, though compression ratio (CR) is higher, due to loss of unimportant information, reconstruction error arises; but in lossless compression, no data is lost, resulting in lower CR. In various medical applications, lossy data compression process is not acceptable as there may be loss of any important feature. Due to this fact, in this work, a completely lossless run-time compression process of ECG and PPG signals is proposed that can also be used after any lossy compression. In physiological signals, there are two types of redundancy, viz. interbeat redundancy, present within a beat, and intrabeat redundancy, present within each single beat [4]. Elimination of these redundancies results in good compression performance.

B. Related works on lossless ECG compression

There are many published papers on lossy compression implemented upon ECG signal on single lead [5] and multilead data [6]. Very few papers are available on lossless compression, and the number of works reported on lossless PPG compression, is even less. Quantization is a process to convert the fractional values of physiological data into quantized numbers. Vector quantization based lossless compression using set partitioning in hierarchical tree (SPIHT) model is able to provide high CR [7]. Linear prediction (LP) [8] is a very useful tool for lossless ECG compression, where future data are forecasted from past samples, in any time-series data. This prediction somehow helps in lossless encoding techniques like Huffman coding, variable length coding [9], entropy encoding (EC) etc. Huffman or EC is very popular lossless coding, in which specific codes are provided to repeating data/characters. Now, compression is performed by assigning smaller codes to high frequent characters and higher codes to those data, which have low frequency of appearance. In some works, golomb-rice

coding (GRC) [10] was used as primary compressing tool by providing optimal prefix-codes to ECG signals, mainly in hardware applications [11]. This process assisted in data compression and transmission in run-time using suitable packet formation [12], [13]. Multilead ECG lossless compression is also proposed using fast Levinson-Durbin recursion techniques [14] in recent days. Using LP, error modeling, and GRC sufficiently improved the CR [15]. ASCII character coding-based compression was introduced by Mukhopadhyay et al. in [16] that caused low reconstruction error, but good CR.

C. Related works on PPG compression

PPG compression research is not a vast aspect till date. Among the published works, mainly lossy compression is exploited. Fourier transform is one of the lossy algorithms that converts any time domain signal into frequency domain. Selection of dominant coefficients results in good CR, but generates reconstruction error [18]. In tele-monitoring applications, Mukhopadhyay et al. proposed ASCII character coding, to compress PPG data [19]. In recent works, steganography [20] along with PPG compression using ASCII coding and SVD was proposed in [21]. This paper achieved high CR, but at the cost of comparatively higher PRD. Though PRD was not so high, all these works were lossy. In [22], author proposed run-time PPG compression using Huffman coding, with low PRD, which can be assumed as quasi-lossless. To the best of the authors' knowledge, no published work is available on developing exact lossless compression in PPG signal.

D. Contributions and paper organization

Therefore, in this work, a new lossless compression process for biological signals (which are periodic in nature), is proposed. The compression result was tested using various ECG and PPG signals in run-time operation. Hence, the main contributions of this proposed work are;

- 1) Proposal of a new lossless compression procedure using second order delta encoding (DE) and run length encoding (RLE).
- 2) Introduction of the usage of buffer array in RLE strategy to improve CR.
- 3) Proposal of an improved approach of lossless compression for any biological signal (irrespective of sampling frequency and data resolution) that have intersample redundancy, assisting in medical science and clinical application.

In addition, the experimental results were tested on ECG and PPG signals to corroborate the fact that this algorithm can be utilized in any biological signal that has inter-sample redundancy. A run-time data collection using hardware module (Raspberry Pi, AD8232 ECG module and MCP3008 ADC chip) and performing compression /decompression algorithm on those signals, in python programming platform and transmission to cloud server using active internet facility, were accomplished. For user-friendliness, the algorithm was evaluated on a smart phone device too.

The rest of this article is organized as follows, Section II describes the hardware setup and proposed method of compression process, Section III shows the experimental result

performed on various ECG and PPG signals along with detailed discussion, and finally, Section IV concludes this article.

II. METHODOLOGY

A. Hardware setup and connection scheme

The run-time data collection, data processing and transmission were performed using Raspberry Pi-4 (R-Pi) [23] and AD8232 [24] single channel ECG module (commercially available). The AD8232 is a heart rate monitoring module that operates on 3.3 Volt DC voltage source. This device can provide single channel ECG output (analog) by placing three electrodes on human body (left arm, right arm, and right leg). The R-Pi is a single board computer device with 1.5 GHz and 64-bit quad-core Cortex-A72 processor and 4 gigabyte random access memory (RAM). The Rs-Pi executed the compression and reconstruction procedure after collecting the data. As a result, any other hardware device with adequate internal storage capacity and a data processing unit (e.g., Banana Pi, ASUS Tinker Board S etc.) can be used in place of the Rs-Pi. Since the proposed approach was developed in python, the code must be run with the appropriate software that supports these other devices. Now, AD8232 provides analog output, but R-Pi can deal with only digital signal; therefore, an analog to digital convertor (ADC) module, MCP3008 was utilized to convert the analog output into digital signal with 10 bits per sample resolution and was fed into R-Pi [25]. The connection scheme of various components is shown in Fig. 1(a) and hardware set-up is shown in Fig. 1(b). The wire description is explained in Table I, mentioning direction of data flow. In this Table, 'R{ ϵ }', 'A{ ϵ }' and 'M{ ϵ }', represent the ϵ -th pin of R-Pi, AD8232 and MCP3008 module, respectively. The wire numbers are made similar in both Table I and Fig. 1(a), for ease of understanding. To perform run-time data transmission, the compressed data were transmitted wirelessly and stored in cloud storage. For reconstruction, the same data were to be downloaded and decompressed. This proposed algorithm was also performed in smart phone devices. A snapshot of received compressed data and reconstructed ECG signal on smart phone screen, is shown in Fig. 1(c).

B. Signal processing flow

Fig. 2 shows the basic structure of both ECG and PPG beat [26]. In this paper, sample to sample coherency of these signals was eliminated using DE and RLE, resulting in high CR with no data loss. The entire algorithm is shown in Fig. 3. At first, the single channel signal was quantized using quantization level ' n ' (provided by the user). Then, second order delta encoding was performed on this signal, followed by the proposed method of RLE using buffer arrays. The amount of compression depends upon user provided value β , discussed later. Finally, after header byte generation, the compressed data were either

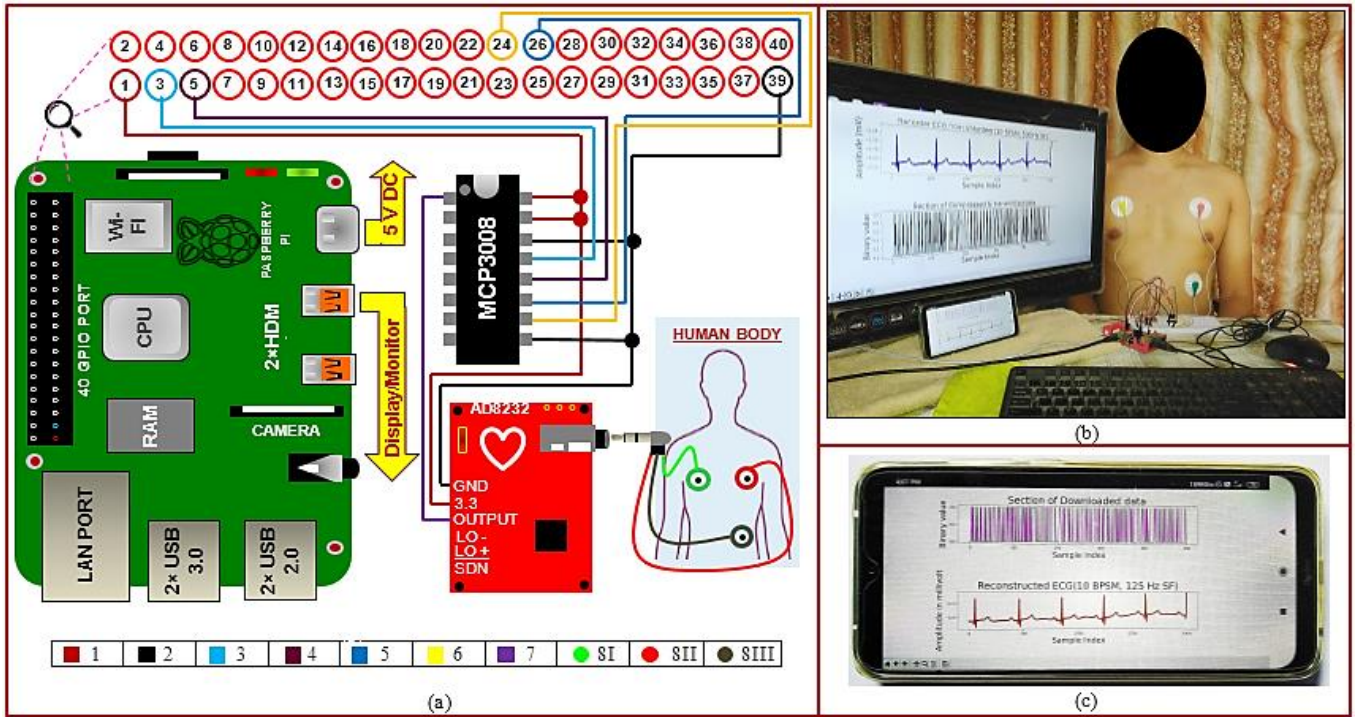


Fig. 1. Circuit diagram of various components used in proposed work, mentioning color codes of used wires at the bottom of this figure; (b) Hardware set up for run-time data collection, compression and transmission; (c) Snap shot of reconstructed signal, performed in smart phone.

stored or transmitted using active internet, to cloud storage.

C. Noise elimination from raw signal

Due to the contamination of various noise signals, denoising was a necessary process before data compression. For both ECG and PPG signal, a second order bandpass Butterworth filter with lower cut off frequency of 0.5 Hz, was designed to eliminate too low frequency signals, that create baseline artifact. The upper cut-off frequency was 100 and 3.4 Hz for ECG and PPG signals respectively [27], [28]. The ranges of cut off frequency were so chosen that the amplitude ratio (AR),

be limited within 10 %, as mentioned in the American Heart Association (AHA) rule [29]. After denoising the raw signal, lossless data compression was initiated as explained below.

D. Proposed algorithm of compression

In this work, to eliminate the intrabeat redundancy, second order delta encoding was performed upon single channel ECG/PPG signal. Initially, the signal was quantized by n -bit quantization level, using the following equation,

$$Y_i = \{(X_i - X_{\min}) / (X_{\max} - X_{\min})\} \times 2^{n-1} \quad (1)$$

where, X_i and Y_i represent the i -th sample in original signal, ' X ' and quantized array, ' Y ' of respective signal X ; X_{\max} and X_{\min} represent the maximum and minimum value of signal X . The aim of quantization process was to

TABLE I

CONNECTION SCHEME OF VARIOUS COMPONENTS, REFERS TO FIG. 1(A)

Wire	Data flow		Description
	In	Out	
1	R{1}*	3.3 v	3.3 V DC power supply to AD8232 and MCP3008 at A{3.3}*, M{15}*, M{16}
2	R{39}	GND	Ground supply to A{GND} and M{9}, M{14}
3	R{13}	M{13}	Clock input to ADC from R-Pi, controlled by software coding
4	R{5}	M{12}	Data input from the Raspberry Pi.to ADC
5	M{11}	R{26}	Data output pin from ADC to R-Pi
6	R{24}	M{10}	Chip select pin to ADC, controlled by R-Pi
7	A{OUTPUT}	M{1}	ECG output signal in analog form
8	I. RA	3.5 mm connector	3 ECG leads, obtained from human body to 3.5 mm connector pin of AD8232
	II. LA		
	III. LL		

* 'R{ ϵ }', 'A{ ϵ }' and 'M{ ϵ }', represent the ϵ -th pin of Raspberry Pi (R-Pi), AD8232 and MCP3008 module, respectively.

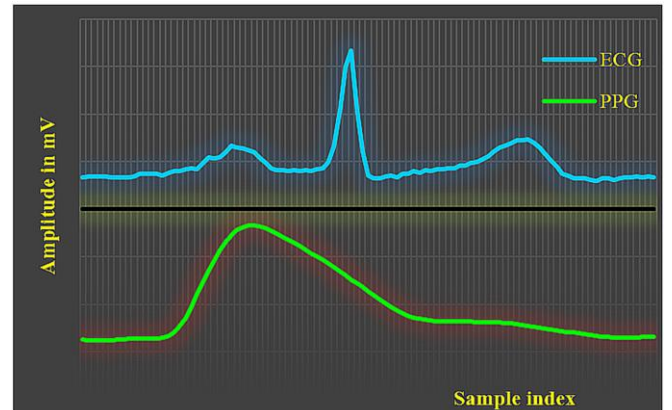


Fig. 2. Structure of single channel ECG and PPG signal.

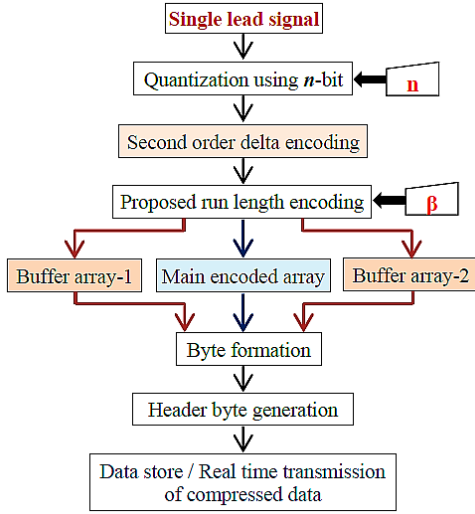


Fig. 3. Signal flow diagram of proposed work of lossless compression.

represent each sample in quantized number, so that the delta encoded array contains integer values. Finally, the second order DE was computed by taking sample-to-sample differentiation [30] using following equations,

$$\begin{aligned} \Delta_i^1 &= Y_{i+1} - Y_i ; \text{ where } i=1,2,\dots,N-1 \\ \Delta_j^2 &= \Delta_{j+1}^1 - \Delta_j^1 ; \text{ where } j=1,2,\dots,N-2 \end{aligned} \quad (2)$$

where, Δ^1 and Δ^2 represent the first and second order delta encoded array of signal Y , (with signal length ' N '), respectively. The delta encoding is a completely lossless coding technique. To reconstruct the original signal, cumulative addition of consecutive samples of delta encoded array is performed. For this purpose, the first sample of the original signal is necessary i.e., to reconstruct signal Y from Δ^1 , Y_1 -th sample is always necessary. Thus, after computing this, two samples were stored (viz. Y_1 and Δ^1), separately, to finally reconstruct Y . Now, after computing DE, it was observed that due to the presence of internal redundancy, presence of successive 'zero's ('0's) were almost common in array Δ^2 . Thus, to remove the coherency, this repetition was eliminated using the number of times, 'zero' was repeated i.e., run length encoding (RLE) [31]. An example of RLE is shown in Fig. 4, where the encoded array Θ contains the number of occurrences of consecutive '0's, in spite of keeping all of them, as present in main array Δ^2 .

Now, in run-time operation, it is always necessary to convert this array in binary form, hence each element is needed to be converted into equivalent binary number. Now, the number of allocated bits to each element (which is denoted as bits per sample or 'bpsm' in the rest of this

Δ^2	-1	0	-3	0	0	0	0	-1	2	0	0	0	0	0	0	0	1	-16	1	8	0	0	0	0	1
Θ	-1	0	-3	0	4	-1	2	0	6	1	-16	1	8	0	3	1									

Fig. 4. Example of general procedure of run length encoding (RLE).

paper) should be the same and it should be equal to the number of bits, needed to express maximum-absolute value (MAV) along with its sign, in that array. In Fig. 4, in Δ^2 array, MAV=16. So, 6 bits are needed to store each element (5 bits to store amplitude and 1 bit to store sign of respective element). Hence, in this example, Δ^2 and Θ array needs 138 (23×6) and 96 (16×6) bits, respectively as the number of elements in these two arrays are 23 and 16 respectively; hence the CR = $(138/96) = 1.4375$. Now, to increase this CR, in this work, implementation of two separate buffer arrays was introduced, as followed.

In Fig. 4, if the number of occurrences of '0's becomes higher than MAV of that signal, then bpsm of Θ array goes higher than the number of bits to store the MAV. In addition, reduction of MAV, results in lower bpsm, which ultimately reflects as a smaller number of bits, needed to express the entire encoded array. Keeping consideration of these facts, in this work, two buffer arrays were proposed, viz. 1) buffer array-1, expressed by Ψ , used to store the number of occurrences of '0's, and 2) buffer array-2, expressed by ξ , used to decrease bpsm. Two parameters were introduced as α , and β , which acted as a relationship links between main encoded array Θ and the two buffer arrays Ψ , and ξ . Now to reduce MAV, 2 arrays (row vectors) were generated as T_{smp} and T_{bit} , explained below.

The i -th ($i=1, 2, 3, \dots, \mathbf{J}$, where \mathbf{J} is the number of bits needed to store the actual MAV of Δ^2 array with sign e.g., for the previous example in Fig. 4, $\mathbf{J} = 6$) element of T_{smp} contains number of samples present in Δ^2 array that can be expressed by $(2+i)$ or higher binary bits. Let, in the array shown in Fig. 4, $T_{\text{smp}} = [23 \ 2 \ 2 \ 1]$ (because, total 23 samples can be expressed in 3 or higher bits, hence $i = 1$ -th element of T_{smp} is 23, similarly, 2 samples can be stored in 4 or higher bits, hence the second element is 2 and so on). In this work, those samples, which needed higher number of bits to express, were stored in ξ array and expressed in \mathbf{J} bpsm and those samples were replaced from Δ^2 array by an identification value, ' α ', so that Θ array can be stored in \mathbf{K} bpsm where $\mathbf{K} \leq \mathbf{J}$. Now, to find out the optimal value of \mathbf{K} , T_{bit} array was used, where i -th element of this array $T_{\text{bit}}(i)$ was computed as,

$$\begin{aligned} T_{\text{bit}}(i) &= T_{\Delta^2}(i) + T_{\xi}(i) + T_{\Psi}(i) \\ T_{\Delta^2}(i) &= \left\{ T_{\text{smp}}(1) \right\} \times (\mathbf{K}) \\ T_{\xi}(i) &= \left\{ T_{\text{smp}}(i+1) \times (\mathbf{J}) \right\} \\ T_{\Psi}(i) &= \left\{ T_{\text{smp}}(i+1) \times (\beta) \right\} \end{aligned} \quad (3)$$

where, $i = 1, 2, 3, \dots, \mathbf{J}-1$; T_{Δ^2} , T_{ξ} , and T_{Ψ} represent bpsm of Δ^2 , ξ , and Ψ array, respectively with $\mathbf{K} = 2+i$. e.g., for $T_{\text{smp}} = [23 \ 2 \ 2 \ 1]$, respective $T_{\text{bit}} = [87 \ 110 \ 133 \ 138]$. Hence, for $i = 1$ i.e., $\mathbf{K} = 3$, lowest number of bits are needed to

TABLE II
COMPRESSION ALGORITHM

Algorithm 1. Proposed RLE based compression algorithm

Δ^2 = Delta encoded (twice) array with length ' M '
 Θ = Empty array representing RLE encoded array
 Ψ = Empty array representing buffer array-1
 ξ = Empty array representing buffer array-2
 α = Identification value ; β = manually provided by user
 $i = 1 ; j = 1 ; c = 0 ; r = 0 ; \mu = 2^{\beta} - 1$

START:

1. Check $\Delta^2_i = 0$, if yes go to step 2, else step 16
2. $c = c + 1$, go to step 15
3. Check if $c = 1$, go to step 4, else step 5
4. $\Theta = [\Theta \ 0]$, go to step 15
5. Check if $c = 2$, go to step 6, else step 7
6. $\Theta = [\Theta \ 0 \ 0]$, go to step 15
7. $r = \text{remainder of } c / \mu$
8. $q = \text{quotient of } c / \mu$
9. Check $r \neq 0$, if yes then $q = q + 1$, else $q = q$
10. Check $j > q$, if yes go to step 21, else step 11
11. $\Theta = [\Theta \ \alpha]$
12. Check $j \neq q$, if yes go to 13, step, else step 14
13. $\Psi = [\Psi \ \mu]$, $j = j + 1$, go to step 10
14. $\Psi = [\Psi \ q]$, $j = j + 1$, go to step 10
15. $c = 0$, $i = i + 1$, if $i \leq M$ go to step 1, else step 22
16. Check $c > 0$, if yes go to step 3, else step 17
17. Check $\Delta^2_i \geq \alpha$, go to step 18, else step 21
18. $\Theta = [\Theta \ \alpha]$
19. $\Psi = [\Psi \ 0]$
20. $\xi = [\xi \ \Delta^2_i]$, go to step 15
21. $\Theta = [\Theta \ \Delta^2_i]$, go to step 15
22. End

store the entire compressed data. Therefore, the array Δ^2 can be expressed in K ($=3$) bpsm with $\alpha = 2^{K-1} = 4$, as a replacement of those samples, which were stored in ξ array. To conclude, those samples that needed more than K ($=3$) bits to express in array Δ^2 (i.e., -16 and 8, in the example shown in Fig. 4), were replaced by α and stored in ξ array sequentially, with updating the Ψ array, as explained below.

Buffer array 2: the main function of buffer array Ψ was to store the number of occurrences of consecutive '0's in array Δ^2 . This array acted as an identifier of sample replacement from Δ^2 to ξ array, mentioned above and used to encode the main array Δ^2 into Θ array. Now, the Θ array was stored in K bpsm, and Ψ array was stored in β bpsm, provided manually. A simple algorithm depicted in Table II shows the process of generation of these three encoded arrays (i.e., Θ , Ψ , ξ) from delta encoded array Δ^2 .

This algorithm, shown in Table II starts with searching for consecutive zeros in the delta encoded array Δ^2 and every time it starts counting those repetitions in an element 'c'. Now, 1) if more than two consecutive zeros are counted, the compiler divides the number, stored in 'c' by $(2^{\beta} - 1)$. The remainder and quotient are stored in two scalars denoted as 'r' and 'q'. Now, the entire c number of consecutive '0's is replaced by inserting α in the array Θ , repeated by ' $q + \lambda$ ' ($\lambda = 1$ if $r \neq 0$, otherwise, $\lambda = 0$) times and the Ψ array is updated by padding $(2^{\beta} - 1)$, repeated by ' $q - \gamma$ ' ($\gamma = 0$ if $r \neq 0$, otherwise, $\gamma = 1$) times followed by padding 'r' once (padding of 'r' is performed only if $r \neq 0$).

2) if any non-zero element is searched by algorithm in array Δ^2 , and if it is less than α , that element is directly padded in Θ array otherwise,

3) if that non-zero element is not less than α , that element was replaced by padding α , once in Θ array, '0' is padded once in Ψ array and that non-zero element is stored in ξ array.

A better realization of this proposed algorithm can be explained using the same array given in Fig. 4. After encoding that array using the proposed algorithm, the result is shown in Fig. 5 with $\beta = 2$. For this array, $\alpha = (2^{3-1}) = 4$ (discussed earlier). Now, while the algorithm finds any non-zero element in Δ^2 , with absolute value less than α (i.e., $-3 \leq \Delta^2_i \leq 3$), are directly stored in Θ , sequentially. While four consecutive '0's were detected, it was divided by $2^{\beta} - 1$ ($=+3$), which resulted in remainder ' $r = 1$ ' and quotient ' $q = 1$ '. As $r = 1$, $\lambda = 1$ and $\gamma = 0$. Therefore, four zeros in Δ^2 array, was replaced by α , repeated by $q + 1$ time in array Θ and array Ψ was updated by inserting $2^{\beta} - 1$ ($= 3$) by q times, followed by padding r ($= 1$), once. When any non-zero number was detected higher than α (in this example, these are -16 and 8), it was replaced by padding α in Θ array, '0' in Ψ array, and finally that sample was stored in ξ array. In this way, the main array can be represented using three different arrays, viz. Θ , Ψ and ξ , which can be stored in 3, 2 and 6 bpsm, respectively. Therefore, total $(15 \times 3) + (7 \times 2) + (2 \times 6) = 71$ bits are needed to store hence the new CR is $(138 / 66) = 1.943$; thus, increment of CR is approximately $(1.943 / 1.4375) = 1.352$.

While decoding the main array, first, the second order delta encoded array was reconstructed in an empty array expressed by Φ , using three encoded arrays (i.e., Θ , Ψ , and ξ). The reconstruction algorithm is shown in Table III. The compiler starts searching for each element from Θ consecutively. Now, if an element is not equal to α , then that sample is padded directly in Φ , otherwise the compiler searches for respective element in Ψ , incrementing the program counter by 1 for this array. If that element is a non-zero number, then respective number of '0's is padded in Φ ; if the element in Ψ is equal to zero, then respective sample in ξ array is padded into Φ , incrementing the program counter by 1 for this array. In this way, the second order delta encoded array is generated, from which the original signal was computed as follows,

Δ^2	-1	0	-3	0	0	0	0	-1	2	0	0	0	0	0	0	1	-16	1	8	0	0	0	1
Θ	-1	0	-3	4	4	-1	2	4	4	1	4	1	4	4	1								
Ψ	3	1	3	3	0	0	3																
ξ	-16	8																					

Fig. 5. Example of run length encoding proposed in this work using two buffer arrays viz. Ψ and ξ , with main encoded array, Θ .

TABLE III
DE-COMPRESSION ALGORITHM

Algorithm 2. Proposed RLE based reconstruction algorithm

Θ = RLE encoded array with length ‘L’
 Ψ = buffer array-1
 ξ = buffer array-2
 Φ = Empty array representing reconstructed array
 $i=1$; $k=1$; $m=1$; $n=1$;

START:

1. Check $\Theta_i = \mathbf{a}$, go to step 2 else step 9
2. Check $\Psi_k = 0$, if yes go to step 3, else step 4
3. $\Phi = [\Phi \ \xi_m]$, $m = m + 1$, $k = k + 1$, go to step 8
4. Check $n > \Psi_k$, if yes go to step 7, else step 5
5. $\Phi = [\Phi \ 0]$
6. $n = n + 1$, go to step 4
7. $n = 1$, go to step 8
8. $i = i + 1$, if $i \leq L$, go to step 1, else step 10
9. $\Phi = [\Phi \ \Theta_i]$, go to step 8
10. End

$$M_i = \sigma + \Phi_i, \quad \left\{ \begin{array}{l} \text{if } i = 1, \sigma = \Delta_1^1; \text{ else } \sigma = M_{i-1} \end{array} \right\}$$

$$N_i = \rho + M_i, \quad \left\{ \begin{array}{l} \text{if } i = 1, \rho = Y_1; \text{ else } \rho = N_{i-1} \end{array} \right\} \quad (4)$$

where M_i and N_i represent the i -th sample of first order delta encoded array (regenerated), ‘ M ’ and original signal (regenerated), ‘ N ’; Δ_1^1 and Y_1 are the extremas, stored during compression, which represent the first sample of Δ^1 array and original signal, respectively.

E. Selection of optimal value for β

It was observed that the CR varied with the variation of β value, for single channel signal. Hence, it was necessary to select the optimal value of β before initiating the compression. Now, the value of β for which, CR was maximum, represented as β_m , was dependent on SF, bpsm, and signal length. Therefore, to select the optimal value of β_m , a multilayer perceptron neural network (MLPNN) was trained using feedforward backpropagation algorithm and ‘sigmoidal’ activation function. Table IV shows the hyperparameters that were chosen for this network. The hyperparameter selection was needed in order to choose the best network model, which would increase NN training efficiency, while reducing the convergence time. Eight parameters were chosen for this reason, as shown in column 1 of Table IV, along with the ranges (or choices) in column 2. These ranges/choices were chosen heuristically in order to better suit the network model. The column 3 of Table IV shows the optimal values for these

TABLE IV
SELECTION OF HYPERPARAMETER

Hyperparameter	Parameter range	Optimized value
No. of hidden layer	1 to 5	3
No. of neurons	20 to 100	28
Learning rate	0.0001 to 0.1	0.002140
Activation function	‘tanh’, ‘sigmoid’, ‘relu’	‘sigmoid’
Optimizer	‘adam’, ‘rms’, ‘sgd’	‘sgd’
Drop-out rate	0.01 to 0.5	0.0247555712
No. of epochs	200 to 1000	874
Batch size	8, 16, 32, 64, 128, 256	64

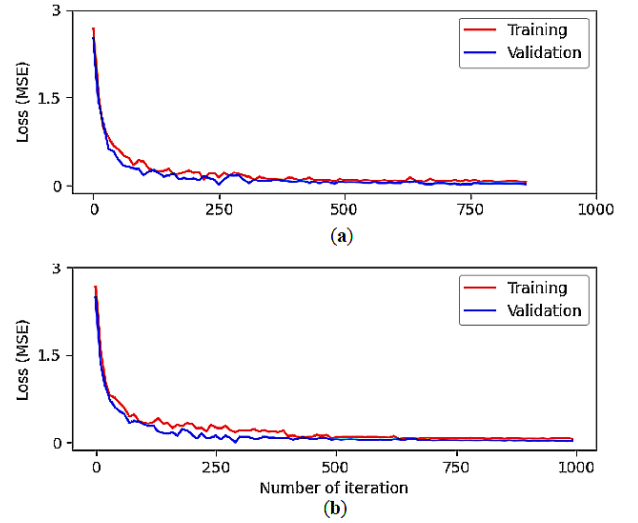


Fig. 6. Variation of training and validation loss with number of iteration (Nol) for MLPNN training process using all hyperparameters and; (a) number of neurons: 28, Nol: 874, optimized by network; (b) number of neurons: 30, Nol: 1000, provided manually.

parameters. Now, when training with these network parameters, there was a sudden increase in network loss, which was expressed as mean squared error (MSE), at higher range of training epochs. This might hamper the overall training of NN; As a result, the number of epochs and neurons were manually set to 1000 and 30, respectively, and it was discovered that the training phase was flawless, yielding maximum absolute error (MAE) of 1.09% and mean squared error (MSE) of 0.0325%. These two parameters were calculated between the predicted output of the trained network using the actual input dataset and output set used during training. Fig. 6 shows the training loss using both the optimum hyperparameters and the hyperparameters that have been manually modified. The input layer of neural network (NN) consisted of 1) SF, 2) bpsm, and 3) signal length and the output layer provided only the optimal value of β_m , denoted as β_m^{NN} , to achieve maximum CR, for the given specifications provided in the input layer. The training dataset were created using the optimal value of β_m , measured by hit and trial (HT) method, denoted as β_m^{HT} , tested on various ECG and PPG signals; and for testing, separate databases were used to check the accuracy. The details of database and accuracy of NN, are mentioned in Section III.

III. RESULT AND DISCUSSION

A. Data collection and noise elimination

Performance of the proposed compression and reconstruction algorithms was tested using various physiological signals like ECG and PPG, which are available in physionet [34]. In addition, this proposed method was tested upon single channel ECG data, which were collected from 10 healthy volunteers, with their informed consent, for run-time operation (i.e., data collection, compression, transmission, and reconstruction). The volunteers were aged between 18 and 57 (details of volunteers are shown in Table V). The data collection was performed in biomedical research laboratory at the department of Electrical

TABLE V
INFORMATION ABOUT VOLUNTEERS

Sl. No.	Volunteer information				
	Gender	Weight (kg)	Height (cm)	Age	Diagnosis
1	M	52	156	18	NIL
2	M	67	149	32	NIL
3	M	61	158	42	Low BP
4	F	54	157	27	NIL
5	M	68	152	35	NIL
6	M	74	154	57	High BP
7	F	54	157	24	NIL
8	F	78	154	26	NIL
9	M	51	172	26	NIL
10	M	61	168	25	NIL

Engineering in Indian Institute of Technology, Roorkee, using R-Pi and AD8232 ECG module, mentioned earlier. Now each signal was first denoised using second order band pass filter with passband frequency between 0.5 and 100 Hz for ECG signals and between 0.5 and 3.4 Hz for PPG signals, so that the AR be limited within 10%. A pictorial representation of denoising process is shown in Fig. 7.

Now, the quality measurement indices of the proposed work are mentioned below, followed by the details of databases and experimental result of each signal.

B. Quality measure indices

In this work, the amount of compression was expressed in form of ‘compression ratio’ (CR) [32], which is basically measure of number of bits needed to store the compressed data. The expression of CR is mentioned below,

$$CR = \frac{\text{bits needed to store original signal}}{\text{bits needed to store compressed data}} \quad (5)$$

The quality of quantization process is measured using percentage root mean squared difference (PRD), which is a measure of quantization error. This proposed work is completely lossless work. Therefore, in this section, PRD is mentioned as measure of quantization error before compression, and it is expressed as,

$$PRD = \left\{ \sqrt{\frac{\sum_{i=1}^L (U_i - V_i)^2}{\sum_{i=1}^L (U_i)^2}} \right\} \times 100 \% \quad (6)$$

where, U_i and V_i represent the i -th sample in the original signal, U and the signal obtained after quantization and de-quantization process, ‘ V ’, of same length ‘ L ’. It is evident that for good compression work CR should be as high as possible and quantization level should be higher to have low quantization error before compression process.

C. Compression result on run-time data

The AD8232 provides analog output voltage representing single channel ECG by placing three electrodes on human body. The three electrodes, which are colored in Red, Green, and Gray, should be placed on the chest, so that the ‘Einthoven’s triangle’ law [1] be fulfilled. A 3.5 mm ECG connector pin is provided in this module which are inherently connected to three ports mentioned above, providing ease of usage of this module. Now, to convert analog output of this module in digital signal, MCP3008 ADC chip was used that uses SPI bus protocol to communicate with R-Pi. The circuit diagram and connection

TABLE VI
AVERAGE COMPRESSION RATIO ON RUN-TIME DATA COLLECTED FROM 10 VOLUNTEERS (125 HZ SF, 10-BPSM)

CR	β	Signal length in samples					
		2000	5000	10000	20000	30000	40000
	2	1.79	1.93	1.98	2.20	2.28	2.34
	3	1.76	1.91	1.95	2.15	2.21	2.29
	4	1.74	1.89	1.92	2.13	2.18	2.22
	5	1.72	1.86	1.90	2.10	2.17	2.20
	6	1.70	1.84	1.87	2.08	2.11	2.19

scheme of proposed work was explained earlier. In this work, the single channel ECG was captured in 10 bpsm with 125 Hz SF from 10 volunteers. The signal length was varied from 2000 to 40000 samples, to compare the obtained result depending upon different values of β , provided manually. The average obtained result is shown in Table VI of these run-time data. It was observed that CR slightly changes with the variation of β . In addition, with increment of signal length, CR increased for same β input.

D. Compression result on PTBDB ECG database

The proposed algorithm was tested on PTBDB ECG database available in physionet [34]. This Pdatabase contains 549 MECG records categorized among various annotation types. Each MECG signal contains 15 simultaneous leads, viz. 12 conventional leads (i.e., I, II, III, aVL, aVR, aVF, V1, V2, V3, V4, V5, and V6) and 3 Frank leads (V_x , V_y , and V_z), among which only lead II was selected for computation. Each signal is measured in 1 kHz SF and 16 bpsm. These signals were obtained from 290 patients with age between 17 to 87 (with mean age 57.2). Among them, 81 patients were female with mean age of 61.6 and 209 patients were male with average age of 55.5. As this proposed algorithm, started by quantizing the single lead signal with ‘ n ’-bit, each PTBDB record was first quantized using different values of ‘ n ’, viz. $n = 6, 8, 10, 12, 14$, and 16 and in each process, the quantization-error (after de-quantizing that signal with respective value of ‘ n ’) is shown in Fig. 7 in the form of PRD, separately. In addition, it was observed that the value of PRD changes with the signal length for same quantization level (Q_{lv}). Therefore, in Fig. 7, variation of quantization error with respect to varying signal length (with 2000, 5000, 10000, 15000, 20000, and 30000 samples, separately) is shown using different Q_{lv} mentioned above. It can be concluded that PRD varies in proportion with Q_{lv} and inversely proportion with signal length. Now, based on these six values of Q_{lv} , variation of average CR is shown in Fig. 8 with variable signal length and different values of β , implemented upon PTBDB records. Table VII shows the average CR and PRD on PTBDB records, using 16-bit

TABLE VII
MEAN COMPRESSION RESULT ON PTBDB DATABASE (1 KHZ SF, 16-BPSM)

CR	β	Signal length in samples					
		2000	5000	10000	150000	20000	30000
	2	6.09	6.15	6.38	6.41	6.49	6.52
	3	6.03	6.08	6.30	6.33	6.42	6.45
	4	5.97	6.02	6.24	6.26	6.35	6.38
	5	5.91	5.96	6.18	6.21	6.29	6.31
	6	5.86	5.91	6.13	6.14	6.22	6.25
PRD (10 ⁻⁶)		149	156	174	178	190	196

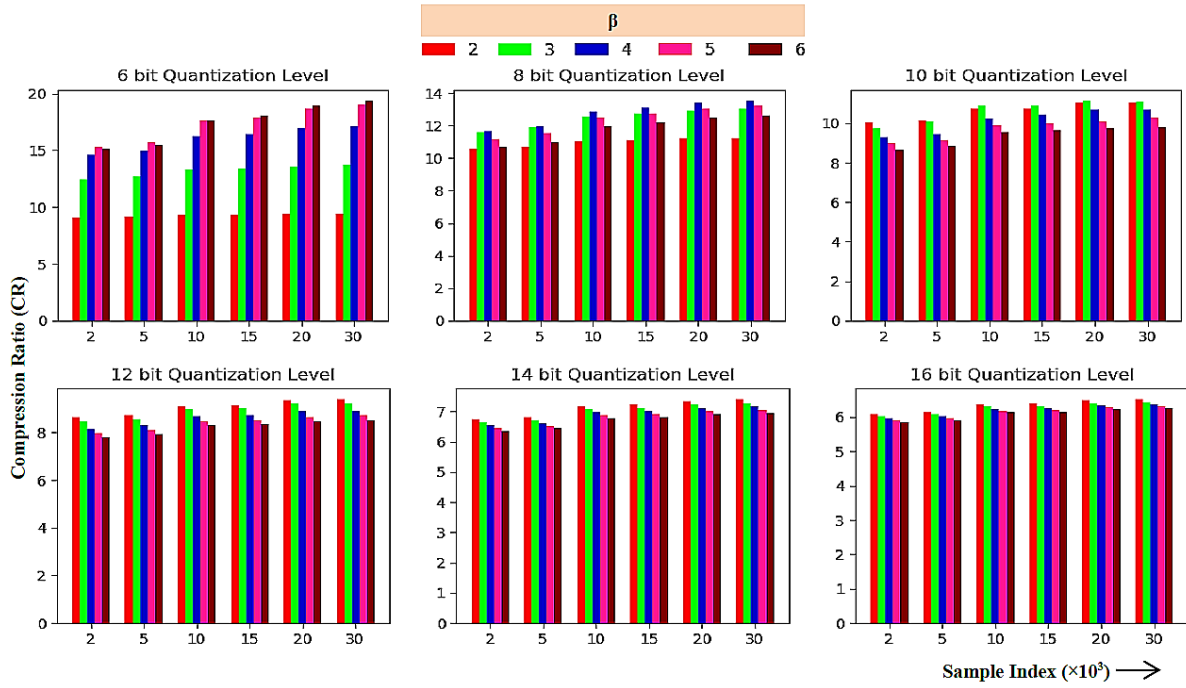


Fig. 7. Variation of average CR with respect to signal length, for varying quantization levels and different values of β , shown in different color shades, for PTBDB records.

quantization level with variable signal length and β . It can be observed that using $\beta = 2$, highest CR was achieved.

E. Compression result on MITDB ECG database

Separate analysis was performed using 48 MITDB records, which were sampled in 360 Hz frequency and 11-bit data resolution. Each record has 2 channel measurements between which, the lead II was used for result analysis. Each MITDB record was quantized using different Q_{IVL} with ‘ n ’= 6, 8, 10, and 11. As the performance result varies with increment of signal length, compression process was performed using 2000, 5000, 10000, 20000, 30000, and 40000 samples separately using each Q_{IVL} mentioned above. Variation of PRD of each quantization process is shown in Fig. 9. Finally, using each quantization level and signal length, separate result analysis was performed with different values of β , viz. 2, 3, 4, 5, and 6, which is shown in Fig. 9. From this figure, it is clear that CR decreases with the

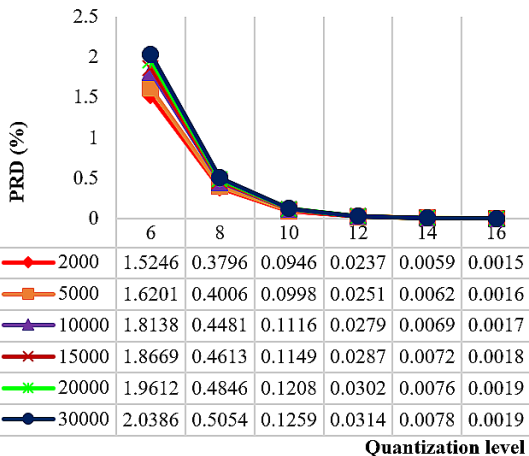


Fig. 8. Variation of average PRD with respect to Q_{IVL} , with varying signal length, shown in different color shades, for PTBDB ECG

increment of Q_{IVL} . It can be observed from Fig. 9, PRD is very low for Q_{IVL} with 10 and 11 and using these Q_{IVL} , highest CR was achieved using $\beta = 3$, shown in Fig. 10. Table VIII defines the variation of CR and PRD, tested upon 48 MITDB records, using variable signal length and β , with 11 bpsm and 360 Hz SF. From Table VIII, it can be concluded that CR is highest for $\beta = 3$ and both CR and PRD increase with the increment of signal length.

F. Compression result on MIMIC-II PPG database

The BIDMIC or MIMIC-II database contains 53 PPG records with 125 Hz sampling frequency. These signals were taken from 53 adult patients with mean age 64.81, among which 32 were female. The proposed algorithm was also tested on this entire dataset and the compression process was initiated by quantizing each signal using various Q_{IVL} , as performed for ECG signals. The PRD is shown in Fig. 11, using separate Q_{IVL} of value 6, 8, 10, and 12, each implanted on signal length 2000, 5000, 10000, 20000, 30000, and 40000 separately. Now, based on these quantization levels and signal length, variation of CR using different β is shown in Fig. 12. Finally, the CR using 12-bit data resolution and variable signal length and β value is shown in Table IX. From this table, it can also be concluded that the CR and increases with the increment of signal length and with decrement of Q_{IVL} .

TABLE VIII
MEAN COMPRESSION RESULT ON MIBDB DATABASE (360 Hz SF, 11-BPSM)

	β	Signal length in samples					
		2000	5000	10000	20000	30000	40000
CR	2	4.31	3.67	3.75	3.88	3.81	3.80
	3	4.29	3.70	3.77	3.91	3.83	3.82
	4	4.24	3.69	3.75	3.89	3.81	3.80
	5	4.20	3.68	3.73	3.88	3.79	3.78
	5	4.15	3.64	3.69	3.83	3.74	3.73
	6	4.15	3.64	3.69	3.83	3.74	3.73
PRD		0.068	0.052	0.053	0.06	0.059	0.06

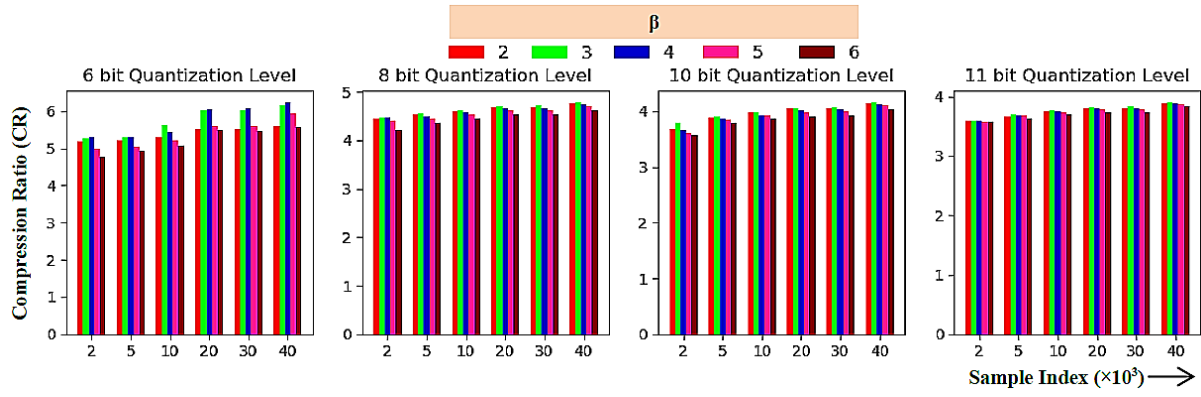


Fig. 9. Variation of average CR with respect to signal length, for varying quantization levels and different values of β , shown in different color shades, for MITDB records.

G. Training of MLPNN to obtain optimal value of β

The performance of the algorithm was dependent on four parameters, viz. SF, Q_{IVL} , signal length and β . The SF and Q_{IVL} was dependent on the device, used for data collection. The length of signal, used for processing, was dependent on the user. The β_m , was varied with the change of SF, Q_{IVL} and signal length. In this work, β was chosen by hit and trial (HT) method between 2 and 6 and for each value, separate result analysis was made. Table X shows the optimal values of β_m^{HT} for ECG and PPG signals with different SF, Q_{IVL} , and signal length. The values of

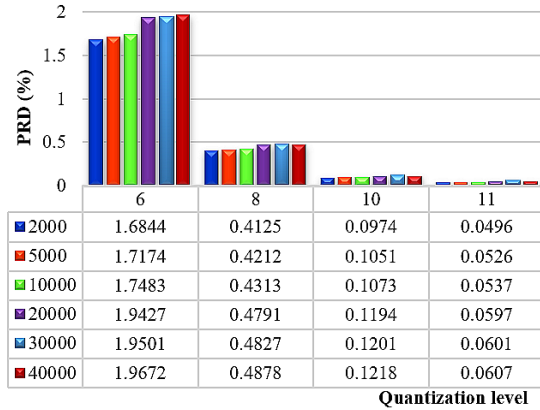


Fig. 10. Variation of mean PRD with respect to Q_{IVL} , with varying signal length, shown in different color shades, for MITDB ECG records.

β are shown in ‘bold’ numbers within this table for ease of understanding. It was observed, that for every signal, value of β increases with the decrement of Q_{IVL} and with increment of signal length. Using these signal specifications and β_m^{HT} values, the MLPNN was trained as mentioned earlier using 150 training datasets with 10-fold cross validation, with a final maximum absolute error as 1.09%. The training and validation dataset is defined in detail in Table XI. For the same SF, signal length, and Q_{IVL} , it was discovered that the value of β was almost fixed. Furthermore, as the signal length increased (above 10000), the value of β remained constant for each group (i.e., signal type, SF and Q_{IVL}). As a result, a nearly equal number of records were chosen for each signal type to ensure uniform training, with the signal duration not exceeding 12000. The training (using 150 records) was done with 10-fold cross validation, which divided the dataset into ten classes (15 records in each group). The rest group was used to analyze the output of trained NN on new input data after the NN was trained using successive 9 groups. The final CV accuracy of the 10 models were obtained as 0.823, 0.889, 0.967, 0.867, 0.833, 0.942, 0.965, 0.833, 0.933, and 0.965. Though the term accuracy is a qualitative concept, in NN prediction process, accuracy is calculated as the ratio of number of correct predictions to the total number of predictions [33]. As a result, greater accuracy equates to better training. The mean loss of these models is shown in Fig. 6. It is clear that the accuracy of third model was best hence, this model was finally selected as

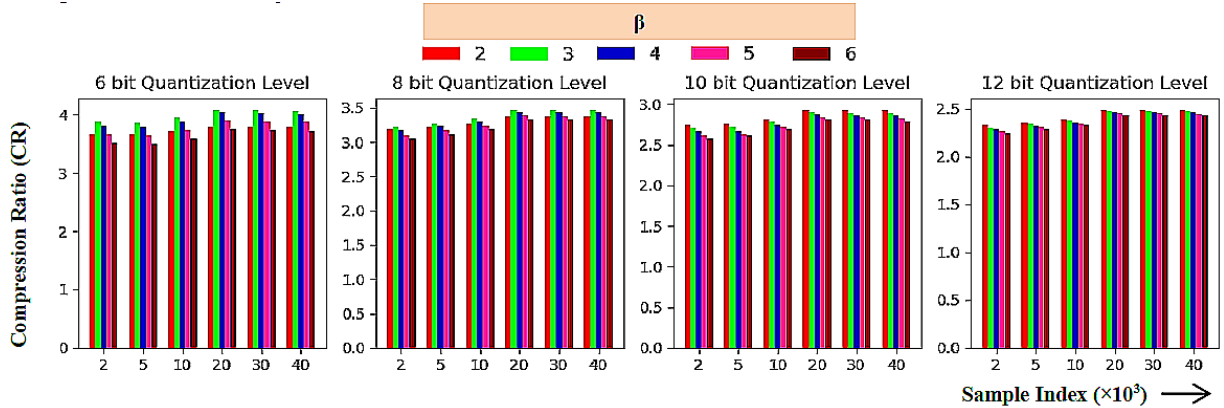


Fig. 11. Variation of average CR with respect to signal length, for varying quantization levels and different values of β , shown in different color shades, for MIMIC-II records.

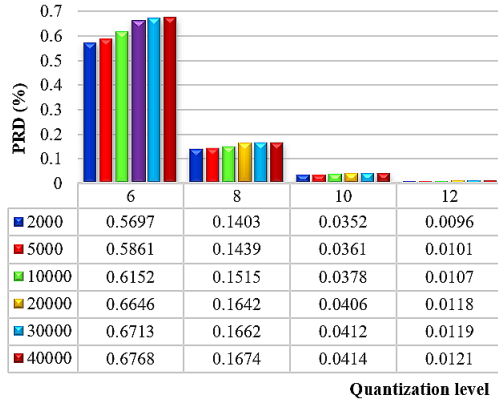


Fig. 12. Variation of average PRD with respect to quantization level, with varying signal length, shown in different color shades, for MIMIC-II PPG records.

TABLE IX
MEAN COMPRESSION RESULT ON MIMIC-II DATABASE (125 Hz SF, 12-BPSM)

CR	β	Signal length in samples					
		2000	5000	10000	20000	30000	40000
PRD (10^{-4})	2	2.33	2.36	2.39	2.49	2.49	2.49
	3	2.30	2.35	2.38	2.48	2.48	2.48
	4	2.28	2.32	2.36	2.47	2.46	2.46
	5	2.27	2.30	2.34	2.45	2.45	2.45
	6	2.24	2.28	2.33	2.43	2.43	2.43
			95	100	106	118	119

trained NN. Now, after completion of training, the blind validation accuracy was measured using 96 records, shown in Table XI. The overall accuracy, measured using these records was approximately 0.953. This CV technique was used to improve the training accuracy. Now, using this trained NN, the compression result was compared using both β_m^{HT} and β_m^{NN} , as mentioned below.

H. Testing of accuracy of MLPNN

To test whether the NN was able to provide β_m , one database, each of ECG and PPG signal, was used. The European ST-T database [34] contains 90 records with 2 channel ECG measurement, sampled in 12 bpsm with 250 Hz frequency. The records were obtained from 79 patients, among which 8 were female (aged 55–71) and 70 were male (aged 30–84), within which the lead-II was used for computation. The IEEE signal processing cup challenge (SPCC) 2015 database [35] contains six channel records of various physiological signals, among which the PPG signal (channel-2) was used for testing. The records were measured from various patients aged between 18 and 35, with 125 Hz sampling frequency. From these two

TABLE X
SELECTION OF β for maximum CR

ECG	SF	Signal length ($\times 10^3$)	Quantization level					
			6	8	10	12	14	16
1 kHz		5	5	4	3	2	2	2
		≥ 10	6	4	3	2	2	2
ECG	360 Hz	Signal length ($\times 10^3$)	Quantization level					
			6	8	10	11	-	-
			2	4	3	3	4	-
ECG & PPG	125 Hz	Signal length ($\times 10^3$)	Quantization level					
			6	8	10	12	-	-
			> 0	3	3	2	2	-

TABLE XI NUMBER OF RECORDS USED TO TRAIN AND VALIDATE MLPNN									
Signal type	SF	Signal length	(Q_m)					Total	
			6	8	10	11	12	Tr. ^b	VL. ^c
ECG	1 kHz	2000	5 3 ^a	5 3	5 3	-	5 3	20	12
		5000	5 3	5 3	5 3	-	5 3	20	12
		10000	5 3	5 3	5 3	-	5 3	20	12
		12000	3 2	3 2	3 2	-	3 2	12	8
ECG	360 Hz	2000	4 3	4 3	4 3	4 3	-	16	12
		5000	4 3	4 3	4 3	4 3	-	16	12
		10000	4 3	4 3	4 3	4 3	-	16	12
PPG	125 Hz	2000	4 2	4 2	4 2	-	3 2	15	8
		5000	4 2	4 2	4 2	-	3 2	15	8

^a5|3: Represents training and validation record 5 and 3, respectively; ^bTr.: Training; ^cVL.: Validation

databases, 15 records were arbitrarily selected from each (15 ECG and 15 PPG) for testing, hence the training and validation dataset were completely different from the test dataset. The experimental result is shown in Table XII. It can be concluded that the values of β_m are almost same for both ECG and PPG signals obtained from HT method (β_m^{HT}) and NN (β_m^{NN}), except for ECG signal with signal length 2000. The variation of CR was 3.24 and 3.22 measured by HT and NN method, respectively; hence very low. Thus, it can be concluded that the NN was almost able to provide, optimal value of β_m for both ECG and PPG signal.

I. Comparison with existing techniques

Table XIII describes a comparative study of previously published works on lossless ECG compression with the proposed work, tested on 48 MITDB database (except S.K. Mukhopadhyay et al., 2011 [16], which is a lossy work, implemented on PTBDB database). Miaou et al. [7] proposed the SPIHT based lossless compression upon wavelet coefficients of ECG data. A very good CR of 3.02 was achieved in their work. LP is a very useful tool for lossless compression used in various works by predicting future samples of any time domain series, from previous samples. Huffman encoding based two stage new encoding strategy, used in [8] along with prediction, resulted in CR of 2.43. A modified length coding using multistage adaptive region predictor [9], resulted in increased CR of 2.67. A new short-term LP based lossless compression was proposed by Li in [9], in which the CR was slightly decreased to 2.28. Hardware implementation of QRS detection and lossless compression was proposed by C. Deepu et al. [10-11]. This work was implemented of MITDB ECG database which resulted in CR of 2.15. T. Tsai [12] et al. introduced GRC along with adaptive LP topology for lossless compression. For run-time decoding purpose, they utilized

TABLE XII
COMPARISON OF β , OBTAINED FROM HT AND NN FOR MAXIMUM CR

Databases	Method	Signal Length			
		2000	5000	10000	15000
E-ST ^a (ECG)	HT ^c	3.24 3 ^e	3.33 2	3.41 2	3.67 2
	NN ^d	3.22 2	3.33 2	3.41 2	3.67 2
ISPC ^b (PPG)	HT	2.17 3	2.23 3	2.31 2	2.53 2
	NN	2.17 3	2.23 3	2.31 2	2.53 2

^aE-ST: European ST-T database

^bISPC: IEEE signal processing cup 2015 database

^cHT: hit and trail method

^dNN: neural network

^e3.24|3: represents CR| β_m i.e., maximum CR of 3.24, for $\beta_m = 3$

TABLE XIII
COMPARISON WITH EXISTING WORKS ON LOSSLESS ECG COMPRESSION

Sl. No.	Related works	Data-Base	No. of records	CR	PRD	Remarks
1	S. G. Miaou et al., 2005 [7]	MITDB	48	3.068	Nil	Set partitioning in hierarchical tree
2	S.-L. Chen et al., 2013 [8]	MITDB	48	2.43	Nil	Linear predictor (LP) - 2stage entropy encoder
3	K. Li et al., 2014 [9]	MITDB	48	2.67	Nil	Multi order adaptive prediction (AP)
4	C.J. Deepu et al., 2015 [10]	MITDB	48	2.28	Nil	Short term liner predictor
5	C.J. Deepu et al., 2016 [11]	MITDB	48	2.15	Nil	ECG-on-Chip joint QRS detection and lossless data compression
6	T. Tsai et al., 2018 [12]	MITDB	48	2.77	Nil	AP, Golomb - Rice coding (GRC)
7	C. Bailey et al., 2019 [13]	MITDB	48	2.43	Nil	Log2-Sub-Band' Encoding (L2SB)- Huffman coding
8	D. Rzepka et al., 2020 [14]	MITDB	48	2.92	Nil	Fast Levinson-Durbin recursion
9	M. Jia et al., 2020 [15]	MITDB	48	3.04	Nil	LP+ error-modeling, GRC
10	S.K. Mukhopadhyay et al., 2011[16]	PTBDB ^a	-	7.18	0.023	ASCII character coding
11	T. Tsai et. al, 2020 [17]	PTBDB	548	4.073	Nil	Adaptive LP+GRC
12	Proposed	MITDB	48	3.82+	Nil	Second order delta encoding, RLE, buffer array
		PTBDB	547	6.52+		

^a lossy compression work implemented on PTBDB records

suitable packet format providing CR of 2.77. In recent works, hardware implementation of Huffman encoding along with a new Log2-Sub-Band' Encoding (L2SB) provided a very good result of 2.43 [13]. A new fast Levinson-Durbin recursion technique given in [14] used in multichannel ECG, provided high CR of 2.92. In [15], authors proposed three different algorithms, viz. error modeling, linear signal predictor and GRC techniques, which enhanced the CR up to 3.04.

As compared to previous techniques, the proposed work introduced RLE and DE, for lossless compression of ECG signal. The computational complications were also very low in this work. A new technique of buffer array on RLE, was proposed, which provided high CR of 3.82 for MITDB records with 11 bpsm and 360 Hz SF. The algorithm was performed on R-Pi module with run-time ECG data collected using AD8232 ECG module. The algorithm was also performed on smart phone, which provided a better and easier way of usage.

In [16], author used ASCII character coding-based ECG compression implemented upon PTBDB database with CR of 7.18. This work was not completely lossless, as a very low PRD (generated due to compression) of 0.023 was obtained in their work. T. Tsai et. al [17] proposed a complete lossless compression using adaptive LP and GRC, implemented on PTBDB database which resulted in CR of 4.073. As compared to this work, implementation of the proposed algorithm on 549 PTBDB records provided complete lossless result with CR of higher than 6.52.

In the area of PPG compression, though there are few papers, but all are almost lossy work. Reddy [18] used Fourier transformation for reducing motion artifacts and data compression of PPG signal. ASCII character coding-based Huffman coded PPG compression was proposed in [19] which is still not lossless. Along with PPG compression, in [21], author proposed steganography to hide confidential information within the compressed data which resulted in PRD of higher than 7%. In recent work, R. Gupta proposed [22] Huffman encoding based run-time lossless PPG compression, which resulted in CR of 1.74 and PRD of 0.055. As compared to previous works, this proposed work introduced a complete lossless compression of PPG signal using second order DE and RLC with buffer array, which provided CR of 2.49 with 12 bpsm and 125 SF for MIMIC-II database.

J. Discussion

The algorithm proposed in this work presented a complete lossless compression work that can be implemented on various biological signals. All the previous works were either tested on single type of ECG (lead-II) or PPG signal. The proposed work was tested on 3 ECG and 2 PPG database; hence, it is evident that this algorithm can readily be applied to physiological signals with interbeat coherency, irrespective of sampling frequency and bpsm. The DE basically eliminated the redundancy and the buffer array based RLE, increasing the CR without generating the PRD. For run-time operation, these data needed to be quantized, and it was observed that the error was reduced by providing higher Q_{lv} . On the other hand, CR and quantization error, both increased with the increment of signal length. To analyze the run-time performance, hardware implementation of the proposed work along with data collection, compression/de-compression and run-time data transmission were proposed using R-Pi. This algorithm was also performed in smart phone device, and data reconstruction procedure after downloading the compressed data, is shown in Fig. 13, along with quantization error curve. The physiological signals are highly noisy in nature and with high noise contaminated signals, performance of the proposed algorithm might be weaker. To analyze this of proposed algorithm, 20 signals were arbitrarily selected, each from PTBDB, MITDB and MIMIC-II databases (total 60 signals) and white Gaussian noise was added to each signal with various noise power (represented by $SNR=[10\log_{10}\{(\text{signal power})/(\text{noise power})\}]$ in dB). The compression result is shown in Table XIV. It was observed that CR decreases with the increment of noise power (it is worthwhile to mention that for each signal, CR was measured for same β_m , for different noisy signal). Though increment of CR was proportional to signal length, rate of increment, gradually reduced with the number of samples increased. In addition, it was observed that CR varied with the varying input value of β . Now, the value of β was provided manually, and therefore the result analysis was so performed to analyze the CR with varying β . For 6, 8, 10, 12, 14 and 16-bit Q_{lv} , the average per sample computation time were 5.6, 9.7, 17.1, 29.4, and 55.2, 94.2 μ s for compression and 3.4, 5.1, 8.5, 12.5, 22.4, and 42.1 μ s for reconstruction, respectively. As this

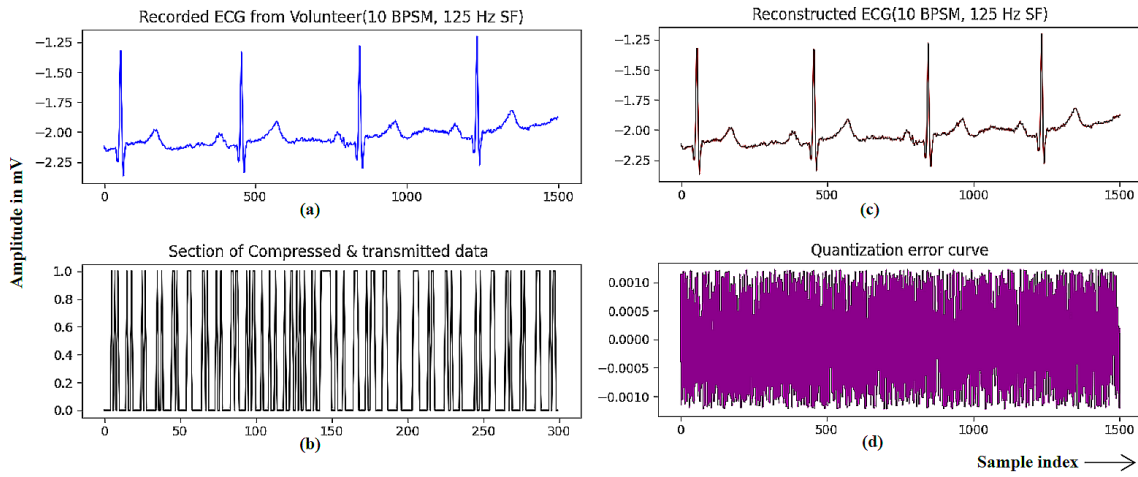


Fig. 13. Pictorial representation of (a) recorded ECG signal, (b) a small section of corresponding compression data, (c) reconstructed signal, and (d) Quantization error curve computed between signals shown in (a) and (c).

TABLE XIV
VARIATION OF MAXIMUM CR WITH VARIOUS LEVEL OF NOISE CONTAMINATION WITHIN VARIABLE SIGNAL LENGTH

SNR (in dB)	ECG: PTBDB (1 kHz SF, 16-bpsm)				ECG: MITDB (360 Hz SF, 11-bpsm)				PPG: MIMIC-II (125 Hz SF, 12-bpsm)			
	Signal length				Signal length				Signal length			
	2000	5000	10000	20000	2000	5000	10000	20000	2000	5000	10000	20000
25	6.11	6.20	6.41	6.53	3.87	3.92	4.01	4.07	2.31	2.32	2.37	2.45
20	3.95	4.19	4.41	4.88	2.48	2.47	2.61	2.90	1.56	1.61	1.68	1.86
15	4.11	4.24	4.46	4.89	2.71	2.55	3.01	3.31	1.64	1.74	1.79	2.02
10	4.23	4.33	4.48	5.01	2.73	2.89	3.04	3.33	1.66	1.75	1.77	2.05
00	4.77	6.13	6.37	6.42	2.84	2.93	3.57	3.59	1.71	2.13	2.17	2.23

work proposed a complete lossless compression, the reconstructed signal was exactly the same as original signal, resulting in no distortion of clinical features. To analyze this fact, for each signal, sample to sample error signal was computed between the original and reconstructed signal (in digital domain) which was a straight line with constant value of 'zero'; hence it ensured this fact that the various characteristics domains of ECG and PPG were intact and all the samples were in their exact value after reconstruction.

IV. CONCLUSION

In this work, a run-time lossless data compression of single channel ECG (lead-II) and PPG signal is proposed, based on delta and run length encoding. The main aim was to reduce the intrabeat redundancy, hence this algorithm can also be implemented on any other signals, which have intra sample coherency. The compression result was tested on universally accepted database (MITDB, PTBDB, and MIMIC-II) available on physionet. In addition, a run-time data collection was performed using AD8232 ECG module and R-Pi, and data transmission was also performed using active internet facility. The ECG module provides Lead I and thus it can be concluded that the proposed algorithm is independent from lead category; however, result analysis, and comparison were performed using standard ECG (Lead II) and PPG records. In this work, no feature extraction was performed or no data loss occurred; hence, the run-time data, those were collected from healthy volunteers, needed no approval of medical authority for quality and acceptability analysis. This algorithm was also performed in smart phone device, thus providing easy usage to doctors and physicians. The entire algorithm was performed in python

programming platform, and therefore tested on Rs-Pi, thus providing an easy and standalone operation. As physiological signals are not completely free from noises, noise sensitivity analysis by adding white Gaussian noise of various noise power, was performed but result comparison was unable to perform in this aspect due to the lack of sufficient information. However, the comparative analysis with the existing works indicated that this procedure provided a better user-friendly, easy, less computationally complicated, lossless data compression and run-time transmission process of both ECG and PPG signal, with high CR and no reconstruction loss.

V. REFERENCES

- [1] R. Gupta, M. Mitra and J. Bera, "ECG Acquisition and Automated Remote Processing", New Delhi: Springer India, 2014.
- [2] A. B. Hertzman, "Photoelectric plethysmography of the fingers and toes in man," *Experim. Biol. Med.*, vol. 37, no. 3, pp. 529-534, 1937.
- [3] World Health Organization, "Cardiovascular diseases (CVDs)," Media Centre, World Health Organization, Geneva, Switzerland, 2016.
- [4] S. Banerjee, R. Gupta and J. Saha, "Compression of multilead electrocardiogram using principal component analysis and machine learning approach," *2018 IEEE App. Sig. Proc. Conf.(ASPCON)*, Kolkata, India, 2018, pp. 24-28.
- [5] S. M. S. Jalaleddine, C. G. Hutchens, R. D. Strattan and W. A. Coberly, "ECG data compression techniques-a unified approach," *IEEE Trans. Biomed. Eng.*, vol. 37, no. 4, pp. 329-343, 1990.
- [6] S. Banerjee and G. K. Singh, "Quality aware compression of multilead electrocardiogram signal using 2-mode tucker decomposition and steganography," *Biomed. Sig. Proc. Cont.*, vol. 64, 2020.
- [7] S. G. Miaou and S. N. Chao, "Wavelet-based lossy-to-lossless ECG compression in a unified vector quantization framework," *IEEE Trans. Biomed. Eng.*, vol. 52, no. 3, pp. 539-543, 2005.
- [8] S. - Chen, G. - Luo and T. - Lin, "Efficient fuzzy-controlled and hybrid entropy coding strategy lossless ECG encoder VLSI design for wireless body sensor networks," *Elect. Lett.*, vol. 49, no. 17, pp. 1058-1060, 2013.

-
- [9] K. Li, Y. Pan, F. Chen, K. Cheng and R. Huan, "Real-time lossless ECG compression for low-power wearable medical devices based on adaptive region prediction," *Electr. Lett.*, vol. 50, no. 25, pp. 1904-1906, 2014.
- [10] C. J. Deepu and Y. Lian, "A joint QRS detection and data compression scheme for wearable sensors," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 1, pp. 165-175, 2015.
- [11] J. Deepu, X. Zhang, C. H. Heng and Y. Lian, "A 3-Lead ECG-on-Chip with QRS detection and lossless compression for wireless sensors," *IEEE Trans. Cir. Sys. II: Express Briefs*, vol. 63, no. 12, pp. 1151-1155, 2016.
- [12] T. Tsai, and W. Kuo, "An efficient ECG lossless compression system for embedded platforms with telemedicine applications," *IEEE Acc.*, vol. 6, pp. 42207-42215, 2018.
- [13] C. Bailey, C. Dai and J. Austin, "A 65-nm CMOS lossless bio-signal compression circuit with 250 femtojoule performance per bit," *IEEE Trans. Cir. Sys.*, vol. 13, no. 5, pp. 1087-1100, 2019.
- [14] D. Rzepka, "Low-complexity lossless multichannel ECG compression based on selective linear prediction," *Biomed. Sig. Proc. Cont.*, vol. 57, 2020.
- [15] M. Jia, F. Li, Y. Pu and Z. Chen, "A lossless electrocardiogram compression system based on dual-mode prediction and error modeling," *IEEE Acc.*, vol. 8, pp. 101153-101162, 2020.
- [16] S. K. Mukhopadhyay, S. Mitra, M. Mitra, "A lossless ECG data compression technique using ASCII character encoding," *Comp. Elect. Eng.*, vol. 37, no. 4, pp. 486-497, 2011.
- [17] T. Tsai, F. Tsai, "Efficient lossless compression scheme for multi-channel ECG signal processing," *Biomed. Sig. Proc. Cont.*, vol. 59, 2020, 101879.
- [18] K. A. Reddy, B. George and V. J. Kumar, "Use of Fourier series analysis for motion artifact reduction and data compression of photoplethysmographic signals," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 5, pp. 1706-1711, 2009.
- [19] S. K. Mukhopadhyay, M. O. Ahmad, M. N. S. Swamy, "ASCII-character-encoding based PPG compression for tele-monitoring system," *Biomed. Sig. Proc. Cont.*, vol. 31, pp. 470-482, 2017.
- [20] S. Banerjee, and G. K. Singh, "A new approach of ECG steganography and prediction using deep learning," *Biomed. Sig. Proc. Cont.*, vol. 64, 2021, 102151.
- [21] S. K. Mukhopadhyay, M. O. Ahmad and M. N. S. Swamy, "Compression of steganographed PPG signal with guaranteed reconstruction quality based on optimum truncation of singular values and ASCII character encoding," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 7, pp. 2081-2090, 2019
- [22] R. Gupta, "Lossless compression technique for real-time photoplethysmographic measurements," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 4, pp. 975-983, 2015.
- [23] F. Fotouhi-Ghazvini, M. Golzar, H. Rabbani, and F. Zakeri, "Mobile cardiac health-care monitoring and notification with real time tachycardia and bradycardia arrhythmia detection," *Journ. Medi. Sig. Sen.*, vol. 7, no. 4, p. 193-202, 2017.
- [24] H. Xia, I. Asif, and X. Zhao, "Cloud-ECG for real time ECG monitoring and analysis," *Comp. Meth. Prog. Biomed.*, vol. 110, no. 3, pp. 253-259, 2013.
- [25] S. Banerjee and G. K. Singh, "Monte Carlo Filter based Motion Artifact Removal from Electrocardiogram Signal for Real-time Telecardiology System," *IEEE Trans. Instrum. Meas.*, Early Access, 2021.
- [26] S. Banerjee, "A first derivative based R-peak detection and DWT based beat delineation approach of single lead electrocardiogram signal". *Proc. of IEEE Reg. 10 Symp. (TENSYP)*, Kolkata, India, 2019, pp. 565-570
- [27] P. R. Rijnbeek, J. A. Kors, and M. Witsenburg, "Minimum bandwidth requirements for recording of pediatric electrocardiograms", *Circulation*, vol. 104, no. 25, pp. 3087-3090, 2001.
- [28] M. Singha Roy, B. Roy, R. Gupta and K. D. Sharma, "On-Device Reliability Assessment and Prediction of Missing Photoplethysmographic Data Using Deep Neural Networks," *IEEE Trans. Biomed. Cir. Sys.*, vol. 14, no. 6, pp. 1323-1332, 2020.
- [29] J. W. Mason, E. W. Hancock, and L. S. Gettes, "Recommendations for the standardization and interpretation of the electrocardiogram", *Hea. Rhy.*, vol. 4, no. 3, pp. 413-419, 2007.
- [30] V. V. Cogo, J. Paulo and A. Bessani, "GenoDedup: Similarity-Based deduplication and delta-encoding for genome sequencing data," *IEEE Trans. Comp.*, vol. 70, no. 5, pp. 669-681, 2021
- [31] Run-length encoding decompression, by J. S. Brooks, R. Golla, A. Danysh, S. Chavan, P. Agrawal, A. E. Weaver, (2020, January 14), *Patent US 10,534,606 B2*.
- [32] M. B. Velasco, F. Cruz-Roldán, J. Godino-Llorente, J. Blanco-Velasco, C. Armien-Aparicio, F. López-Ferreras, "On the use of PRD and CR parameters for ECG compression", *Med. Eng. & Phys.*, vol. 27, no. 9, pp. 798-802, 2005.
- [33] J.R. Parker, "Rank and response combination from confusion matrix data," *Inform. Fus.*, vol. 2, no. 2, pp. 113-120, 2001.
- [34] Physionet.org." [Online]. Available: www.physionet.org // <https://physionet.org>
- [35] Z. Zhang, Z. Pi, B. Liu, "TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise", *IEEE Trans. Biomed. Eng.*, vol. 62, no. 2, pp. 522-531, 2015.