



**AI Techniques for Efficient Healthcare Systems in ECG Wave Based
Cardiac Disease Detection by High Performance Modelling**

Jeba Sonia J

*Department of Data Science and Business Systems, School of Computing, College of
Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur,
Chennai- 603203, India jebas@srmist.edu.in*

D. J. Joel Devadass Daniel

*Assistant Professor, Department of Electronics and Communication Engineering, VelTech
Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu,
India*

drjoeldevadass@veltech.edu.in

Dr. R. Sabin Begum

*Assistant Professor, Department of Computer Applications, B. S. Abdur Rahman Crescent
Institute of Science and Technology, Chennai, Tamil Nadu, India*

sabin@crestent.education

Dr. Amanulla Khan Nasrulla Khan Pathan

*Assistant Professor, Department of Botany, Anjuman Islam Janjira Degree College of
Science, Murud Janjira, Maharashtra, India*

dramanullak@gmail.com

Dr. Veera Talukdar

Registrar, Kaziranga University, Jorhat, Assam, India

bhaskarveera95@gmail.com

Vivek Dadasaheb Solavande

*Assistant Professor, Department of Computer Science and Engineering, Bharati Vidyapeeth
Deemed to be University, Department of Engineering and Technology, Navi Mumbai, India*

viveksolavande@gmail.com

Article History	Abstract
Received: 13 July 2022 Revised: 20 September 2022 Accepted: 26 October 2022	Heart disease (HD) is extremely lethal by nature and claims a disproportionately large number of lives worldwide. Early and reliable detection techniques are necessary to prevent fatalities from HD. Clinical test results, electrocardiogram (ECG) signal, the heart sound signal, impedance cardiography (ICG), magnetic resonance imaging, and computer tomography (CT) can all be used to determine whether an individual has HD. This research propose novel technique in efficient healthcare system by ECG wave based cardiac disease detection using deep learning architecture with high performance modelling. Here the input is collected as ECG waves which has been processed and obtained as ECG wave fragments. This ECG fragment features has been extracted using deep belief kernel principal neural network. Based on this extracted features the patients 3D heart image has been collected and classified using deep quantum multilayer convolutional neural networks. Here the experimental analysis has been carried out in terms of accuracy, precision, recall, F-score, SNR,

CC License CC-BY-NC-SA 4.	RMSE. Proposed technique attained accuracy of 95%, precision of 81%, recall of 69%, F-1score of 73%, SNR of 59% and RMSE of 62%. Keywords: <i>heart disease, electrocardiogram, high performance modelling, deep learning, 3D heart image, cardiology</i>
-------------------------------------	---

1. Introduction

ECG is a key diagnostic tool for heart disease, leading cause of mortality worldwide. Diagnosing cardiac disease is more difficult than deciding whether or not the ECG is normal. Traditional ECG signal analysis and classification is labor- and time-intensive since it depends on the advice of skilled medical professionals [1]. Consequently, there is a rising demand for computer-based, fully automated ECG analysis. Feature extraction is challenging in ECG analysis. Systems combine the features and compare them with features taken from other cardiac diseases to identify the ECG. Some features, however, are challenging to extract because of noise interference [2]. It is impossible to create a system that can extract all the necessary properties because the ECG signals of various cardiovascular diseases have diverse qualities. Because of this, each diagnostic system has a low accuracy and limited scalability. Deep learning has recently been identified as a promising method for interpreting ECGs. Deep convolutional neural networks automatically extract features from the source signals or images. Deep neural network-based ECG interpretation appears increasingly attractive due to its great performance in automated classification. In low- and middle-income countries, CVD accounts for more than 75% of all fatalities [3]. In less developed nations like Bangladesh, India, and some African countries, where proper screening procedures for patients with HD symptoms are still debatable due to financial crisis, limited access to adequate and equitable health care equipment and facilities [4]. Additionally, the general public cannot afford to take advantage of the possibility to get a heart disease diagnostic with the current facilities. ECG is used to diagnose CVD. Main factors that influence success of these methods include feature selection, data extraction methods, classification algorithm types, and—most significantly—the use of imbalanced data for classification, which can decrease minority class's recognition accuracy [5]. Modern research also uses one lead-based ECG images rather than the more traditional 12-lead-based ECG images to demonstrate great accuracy in classification and detection tasks. Additionally, the healthcare institutions used a variety of ECG devices and presented results in irregular ECG image formats. Modern research has not been able to offer a generalised methodology for nonuniform ECG picture formats. Researchers can't get 12-lead-based ECG source images in the general domain. The primary goal of this study is to develop a brand-new automatic detection technique that is comparable to and adaptable for cardiac hospitals to use in processing 12-lead-based ECG pictures. It is crucial to use a deep neural network to automatically detect heart disorders utilizing 12-lead ECG picture processing. DNN is based on mathematical formulae and models and functions similarly to the human brain. Deep neural network mathematical functional principles aim to comprehend and recognise the pattern between various components. The basic building block of a DNN is a neuron, which is educated through repeating activities and learns through experience in a manner similar to that of the human brain. Goal of education as well as knowledge acquisition is to build a link between input and output. System can detect items it has been trained to recognise after training [6].

Contribution of this research is as follows:

1. To propose novel method in efficient healthcare system by ECG wave based cardiac disease detection using deep learning architecture with high performance modelling.
2. Here the input is collected as ECG waves which has been processed and obtained as ECG wave fragments.
3. This ECG fragment features has been extracted using deep belief kernel principal neural network.
4. Based on this extracted features the patients 3D heart image has been collected and classified using deep quantum multilayer convolutional neural networks

2. Background Study

To automatically identify CAD, researchers have tried using several ECG or PCG datasets. The majority of ECG signal research [7] concentrate on the traditional feature extraction and classification technique. Additionally, utilising the ECG data of 47 people, works [8] and [9] employed the CNN model as a method for feature learning and achieved enhanced performance. Work [10] reduced characteristics using linear discriminant analysis (LDA) and then built a classification model using machine learning techniques on the remaining features. Their categorization accuracy was 99.056% on average. Support vector machine (SVM) was employed in study [11] as a recognition system to identify heartbeat, and the experiment verified its accuracy and benefit. A new method for analysing cardiac arrhythmias based on hidden Markov models was described in work [12], which offered better performance compared to preliminary findings. Work [13] used random forest to study cardiac arrhythmia in ECG and reported precise findings. Author [14] created an integrated system using ECG data by combining a tiny, specifically tailored classifier with a general one. A classification method based on nonnegative matrix factorization was initially developed by [15] and has since been used for a variety of classification tasks [16]. The accuracy of work [17]'s nonnegative matrix factorization approach for automatically identifying electro cardio signal was 95.64%. Atrial fibrillation was automatically identified in [18] using an LSTM network. Accuracy of 98.5% and 99.77% are stated to be reached for 20 participants utilizing ten-fold cross validation and for three subjects utilizing blind fold validation. [19] uses the bidirectional LSTM-based DL technique to categorise the ECG signal. To retrieve the sub bands of main signal and then use them as inputs for deep network, the wavelet transform is introduced in the study. The proposed technique reported a 99.39% recognition performance. In [20], the ECG signal was located using a CNN and LSTM model combination.

3. Proposed ECG Wave based Cardiac Disease Detection

This section discuss novel technique in efficient healthcare system by ECG wave based cardiac disease detection using deep learning architecture with high performance modelling. Here the input is collected as ECG waves which has been processed and obtained as ECG wave fragments. This ECG fragment features has been extracted using deep belief kernel principal neural network. Based on this extracted feature the patient's 3D heart image has been collected and classified using deep quantum multilayer convolutional neural networks. The proposed architecture is given in figure-1.

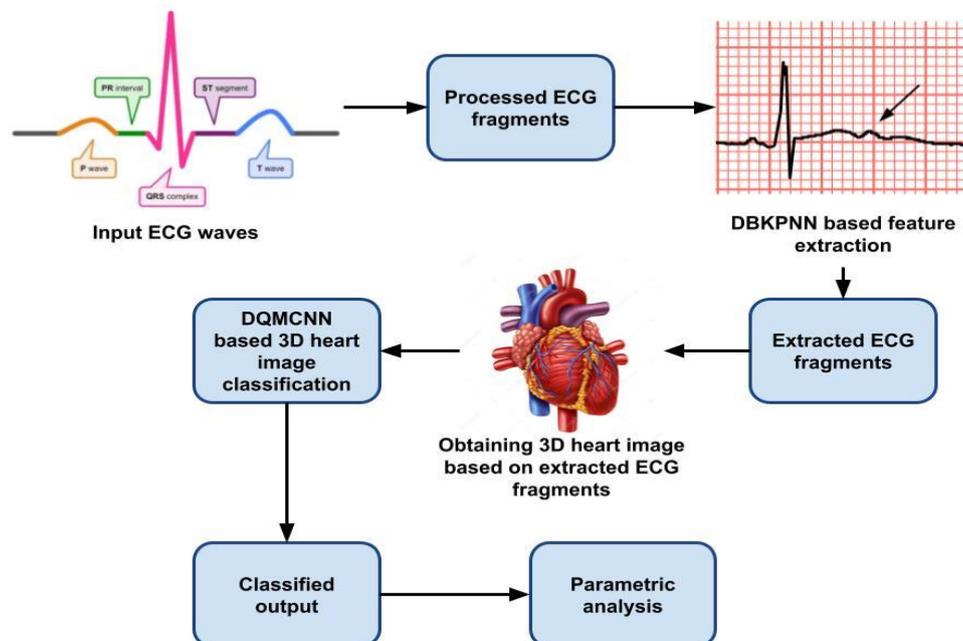


Figure 1. Overall proposed architecture

Before entering data into method for training, data preparation is a crucial step. Data can be arranged into useful information during this stage. Since ECG images are larger than 800 KB and take a long time to train the algorithm, all photos needed to be modified. To complete more training steps in less time, ECG images were reduced in size to around 300 KB. This study gathered the information necessary for each ECG image's 12 leads to be labelled before training. The authors of this work used the LabelImg tool to label the classes in the dataset, which required them to label 48 items from each of the four classes that our trained model will need to recognise.

3.1 Deep Belief Kernel Principal Neural Network Based ECG Feature Extraction

Stacks of RBMs make up the generative visual model known as DBN. DBN can capture a hierarchical representation of incoming data due to its deep structure. The joint distribution is defined as eq(1) given a visible unit of x and l hidden layers.

$$\min_{\theta_L, \theta_{DBN}} \mathbb{E}_{y,x} [\mathcal{L}(\theta_L; \mathbf{y}, h(\mathbf{x}))] + \rho \mathbb{E}_x [-\log p(\mathbf{x}; \theta_{DBN})] \quad (1)$$

Training a DBN's layers is equivalent to training an RBM because each layer is built as an RBM. Through DBN training, a network is initialised to do classification. Every step calls for an optimization challenge to be solved. Pre-training step solves following optimization issue at each layer k by eq(2) given the training dataset $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(|D|)}, y^{(|D|)})\}$ with input x and label y .

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (2)$$

Keep in mind that layer-wise updating requires us to fix all of issues from bottom hidden layer to top visible layer. We use eq(3) to solve following optimization problem at the fine-tuning stage

$$\min_{\phi} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}(\phi; y^{(i)}, h(x^{(i)})) \right] \quad (3)$$

where h stands for final hidden features at layer l , $L()$ is a loss function, and ϕ are classifier's parameters. Here, we'll write $h(x^{(i)}) = h(x_l^{(i)})$ for simplicity. To begin, we use a simplistic model of aggregating training and fine-tuning goals. This method implements two-phase training method simultaneously; however, in order to balance the relative importance of the two objectives, we must add one more hyperparameter ρ . The definition of model (DBN+loss) is eq (4)

$$\min_{\theta_L, \theta_{DBN}} \mathbb{E}_{y,x} [\mathcal{L}(\theta_L; \mathbf{y}, h(\mathbf{x}))] + \rho \mathbb{E}_x [-\log p(\mathbf{x}; \theta_{DBN})] \quad (4)$$

and empirically based on training samples D by eq. (5),

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\sum_h p(h | x^{(i)}) \mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (5)$$

where θ_L, θ_{DBN} are underlying parameters. Note that $\theta_L = \phi$ from (1) and $\theta_{DBN} = (\bar{\theta}_k)_{k=1}$. On the basis of the conditional distribution $p(h|x)$ derived by DBN, we first create an anticipated loss model. Hidden space is classified utilizing this paradigm. It should be more reliable and, as a result, produce better accuracy on data that hasn't been observed because it minimises the projected loss. Equation eq(6) designates mathematical method that minimises anticipated loss function

$$\min_{\theta_L, \theta_{DBN}} \mathbb{E}_{y,h|x} [\mathcal{L}(\theta_L; \mathbf{y}, h(\theta_{DBN}; \mathbf{x}))] \quad (6)$$

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\sum_h p(h | x^{(i)}) \mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (7)$$

With notation $h(\theta_{DBN}; x^{(i)}) = h(x^{(i)})$ we explicitly show dependency of h on θ_{DBN} . By adding a constraint that places constraints on DBN-related parameters with regard to their ideal values, we modify the predicted loss model. This model offers two advantages. In the beginning, the model limits the parameters that were fitted in an unsupervised way to maintain a good representation of the input. The restriction also regularises the model parameters by avoiding their explosion during updating. The mathematical shape of the model can be read by eq(8) given training samples D .

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\sum_h p(h | x^{(i)}) \mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (8)$$

$$\text{s.t. } |\theta_{DBN} - \theta_{DBN}^*| \leq \delta$$

where θ^*_{DBN} are optimal DBN parameters and δ is a hyperparameter. For the method to produce the DBN fitted parameters, a pre-training phase is required. With parameters that are fitted using both supervised and unsupervised methods, this model regularises them. Therefore, even if we require an

additional training in addition to two-phase trainings, it can still attain improved accuracy. The model reads by eq(9) given training samples D.

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\sum_h p(h | x^{(i)}) \mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (9)$$

s.t. $|\theta_{DBN} - \theta_{DBN-OPT}^*| \leq \delta$

where δ is a hyperparameter and $\theta_{DBN-OPT}^*$ are the optimal DBN parameter values following two phases of training. Equation (10) provides mathematical method (FFNDBN) based on training samples D.

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (10)$$

s.t. $|\theta_{DBN} - \theta_{DBN}^*| \leq \delta.$

This model, which combines (11) and training samples D, reads

$$\min_{\theta_L, \theta_{DBN}} \frac{1}{|D|} \sum_{i=1}^{|D|} \left[\mathcal{L}(\theta_L; y^{(i)}, h(\theta_{DBN}; x^{(i)})) \right] \quad (11)$$

s.t. $|\theta_{DBN} - \theta_{DBN-OPT}^*| \leq \delta.$

Hidden layer's jth output layer is defined by equation (12):

$$h_j = \sum_{i=1}^{n_H} w_{ji} x_i \quad j = 1, 2, \dots, n_2 \quad (12)$$

where w_{ji} is the hidden weight, n_2 is number of nodes in hidden layer, and n_1 is number of nodes in input layer. Neural network model's input vector is $x = [u(k), u(k-1), u(k-2), \dots]^T$. where $u(k)$ is output of the neural network controller. The eq. (13) that follows gives the neural network model's output:

$$yr(k+1) = \lambda_s \left(\sum_{j=1}^{n_2} w_{1j} s(h_j) \right) \quad (13)$$

Where w_{1j} is the output weight and λ is a scaling factor. The following equation (14) gives the output's compact form:

$$yr(k+1) = \lambda s(h_1) = \lambda s[w_1^T S(Wx)] \quad (14)$$

With eq. (15)

$$\begin{aligned} x &= [x_i]^T, i = 1, \dots, n_1 \\ W &= [w_{ij}], i = 1, \dots, n_1, j = 1, \dots, n_2 \\ S(Wx) &= [s(h_j)]^T, j = 1, \dots, n_2 \\ w_1 &= [w_{1j}]^T, j = 1, \dots, n_2 \end{aligned} \quad (15)$$

eq(16) provides the identification error $e(k)$

$$e(k) = y(k) - yr(k) \quad (16)$$

The following equation (17) provides the function cost:

$$E = \frac{1}{2} (e(k))^2 \quad (17)$$

N is total number of observations. Following equation (18) updates output weights:

$$w_{1j}(k+1) = w_{1j}(k) + \Delta w_{1j}(k) \quad (18)$$

where $\Delta w_{ji}(k), j = 1, \dots, n$ is given by minimizing cost function described as eq. (19):

$$\Delta w_{1j} = -\eta(k) \frac{\partial E(k)}{\partial w_{1j}} = -\eta(k) \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial h_1} \frac{\partial h_1}{\partial w_{1j}} = \lambda \eta(k) e(k) S(h_1) S(Wx) \quad (19)$$

$\eta(k)$ is variable learning rate for weights of NN method, $0 \leq \eta(k) \leq 1$, given by eq. (20)

$$\eta(k) = \frac{1}{\lambda^2 y^2(k_1) [Sf(W_x) S(W_x) + w_1^T S(W_x) S(W_x) m_1 e^{rxx}]} \quad (20)$$

$s'(h_1)$ is derivative of $s'(h_1)$ described as eq. (21):

$$s'(h_1) = s(h_1)(1 - s(h_1)) = \frac{e^{-h_1}}{(1+e^{-h_1})^2} \approx \frac{1}{4} + \frac{1}{2} h_1 + O(h_1^3) \quad (21)$$

The following equation (23) updates the hidden weights:

$$w_{ji}(k+1) = w_{ji}(k) + \Delta w_{ji}(k) \quad (23)$$

where Δw_{ji} is determined by the equation (24) below:

$$\begin{aligned} \Delta w_{ji} &= -\eta(k) \frac{\partial E(k)}{\partial w_{ji}} \\ &= -\eta(k) \frac{\partial E(k)}{\partial e(k)} \frac{\partial e(k)}{\partial h_1} \frac{\partial h_1}{\partial h_j} \frac{\partial h_j}{\partial w_{jk}} \end{aligned} \quad (24)$$

with $S'(Wx) = \text{diag} [s'(h_j)]^T, j = 1, \dots, n_2$. It is explained how the Lyapunov function affects stability of NN method. Let's define a discrete Lyapunov function as eq(25), for example

$$V(k) = E(k) = \frac{1}{2}(e(k))^2 \quad (25)$$

As we recall, eq(26) provides NN controller's input vector

$$z = [r(k), r(k-1), r(k-2), \dots]^T \quad (26)$$

where $r(k)$ is desired value. Non-linear mapped data in ζ is represented by the input data $\{z_k\}_{k=1}^l, \phi$. The projected features C 's covariance matrix is $l \times l$ defined as eq. (27)

$$C = \frac{1}{l} \sum_{j=1}^l \phi(z_j) \phi(z_j)^T \quad (27)$$

Its eigenvalues and eigenvectors are given by eq. (28)

$$Cp = \lambda_k \sum_{i=1}^l \alpha_i \phi(z_d)^T \phi(z_i) \quad (28)$$

From Equation (28), Equation (29) may be

$$\frac{1}{l} \sum_{j=1}^l \phi(z_j) \left(\phi(z_j)^T p_k \right) = \lambda_k p_k \quad (29)$$

$$p_k = \sum_{j=1}^l \alpha_j \phi(z_j)$$

with $\alpha_j, j = 1, \dots, n$ as expansion coefficients. Equation (29) can be rewritten as eq. (30)

$$\frac{1}{l} \sum_{j=1}^l \phi(z_d)^T \phi(z_j) \left(\phi(z_j)^T \sum_{i=1}^l \alpha_i \phi(z_i) \right) \quad (30)$$

Kernel function $kr(z_i, z_j)$ is described as eq. (31)

$$kr(z_i, z_j) = \phi(z_i)^T \phi(z_j) \quad (31)$$

$\phi(z_d)^T$ Equation (32) becomes

$$\frac{1}{l} \sum_{i=1}^l kr(z_d, z_i) \sum_{j=1}^l \alpha_j kr(z_i, z_j) = \lambda_k \sum_{i=1}^l \alpha_i kr(z_d, z_i) \quad (32)$$

Equation (33) is

$$\frac{1}{l} \sum_{i=1}^l kr(z_d, z_i) \sum_{j=1}^l \alpha_j kr(z_i, z_j) = \lambda_k \sum_{i=1}^l \alpha_i kr(z_d, z_i) \quad (33)$$

with $kr(z_d, z_i) = \phi(z_d)^T \phi(z_i)$

Equation (34) can be used to determine the kernel primary components that result

$$x_r(k) = \phi(z)^T p_k = \sum_{i=1}^l \alpha_i kr(z, z_i) \quad (34)$$

Input vector of NN controller is made up of signal's reduced space as determined by Equation (35) Similarly, the following equation provides the neural controller's output:

$$\begin{aligned} u(k) &= \lambda_c s \left(\sum_{j=1}^{n_4} v_{1j} s(h_{c_j}) \right) \\ &= \lambda_c s \left(\sum_{j=1}^{n_4} v_{1j} s \left(\sum_{i=1}^{n_3} v_{ji} x_{1i} \right) \right) \end{aligned} \quad (35)$$

where λ_c is a scaling factor, v_{1j} is output weight, and n_4 is hidden layer's node count. The following equation (36) gives compact form of system's control input:

$$\begin{aligned} u(k) &= \lambda_c s(h_c) = \lambda_c s[v_1^T S(Vx_1)] \\ x_1 &= [x_{11}]^T, i = 1, \dots, n_3 \\ V &= [v_j], i = 1, \dots, n_3, j = 1, \dots, n_4 \\ S(Vx_1) &= [s(h_j)]^T, j = 1, \dots, n_4 \\ v_1 &= [v_1]^T, j = 1, \dots, n_4 \end{aligned} \quad (36)$$

It is possible to update the weights of neural controller by minimising cost function, which is described by equation (37):

$$E_c = \frac{1}{2} (e_c(k))^2 \quad (37)$$

N is the total number of observations. eq(38) updates output weights

$$v_{1j}(k+1) = v_{1j}(k) + \Delta v_{1j}(k) \quad (38)$$

with $\Delta v_{1j}, j = 1 \dots n_4$, is incremental change of output weights by eq. (39):

$$\begin{aligned} \Delta v_{1j} &= -\eta_c(k) \frac{\partial E_c(k)}{\partial v_{1j}} \\ &= -\eta_c(k) \frac{\partial E_c}{\partial e_c(k)} \frac{\partial e_c(k)}{\partial y(k)} \frac{\partial y(k)}{\partial h_1} \frac{\partial h_1}{\partial(h_j)} \frac{\partial s(h_j)}{\partial h_j} \frac{\partial h_j}{\partial u(k)} \frac{\partial u(k)}{\partial h_{c1}} \frac{\partial h_{c1}}{\partial v_{1j}} \\ &= \eta_c(k) \lambda_c e_c(k) s'(h_1) w_{1j} S'(Wx) w_{ji} s'(h_{c1}) S(Vx_1) \end{aligned} \quad (39)$$

where $\eta_c(k)$ is learning rate for weights of NN controller, $0 \leq \eta_c(k) \leq 1$ represented by eq. (40)

$$\eta_c(k) = 1/(\lambda_c^2 s'^2(h_{c1})s'(h_1)w_{1j}w_{ji}S'(Wx) \times [S^T(Vx_1)S(Vx_1) + v_{1j}^T S'(Vx_1)S'(Vx_1)v_{1j}x_1x_1^T]) \quad (40)$$

The hidden weights are updated by eq (41) with regard to them

$$v_{ji}(k+1) = v_{ji}(k) + \Delta v_{ji}(k) \quad (41)$$

where Δv_{ji} is given by eq. (42)

$$\begin{aligned} \Delta v_{ji} &= -\eta_c(k) \frac{\partial E_c(k)}{\partial v_{ji}} \\ &= -\eta_c(k) \frac{\partial E_c}{\partial e_c} \frac{\partial e_c}{\partial y} \frac{\partial y}{\partial h_1} \frac{\partial y}{\partial s(h_j)} \frac{\partial h_1}{\partial h_j} \frac{\partial s(h_j)}{\partial h_j} \frac{\partial h_j}{\partial u} \frac{\partial u}{\partial h_{c1}} \frac{\partial h_{c1}}{\partial h_{c_j}} \frac{\partial h_{c_j}}{\partial v_{ji}} \\ &= \eta_c(k) \lambda_c e_c(k) s'(h_1) w_{1j} S'(Wx) w_{ji} s'(h_{c1}) v_{1j} S'(Vx_1) x_1^T \\ &\quad \text{with } S'(Vx_1) = \text{diag} [s'(h_j)]^T, j = 1, \dots, n_4 \end{aligned} \quad (42)$$

Let $\Psi = [\phi(z_1), \dots, \phi(z_l)]$, $1_l = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{l \times 1}$ and $\tilde{\Gamma} = \Psi^T \Psi \Gamma$ is matrix which is described as eq. (43)

$$\Gamma = \tilde{\Gamma} - 1_l \tilde{\Gamma} - \tilde{\Gamma} 1_l + 1_l \tilde{\Gamma} 1_l \quad (43)$$

With $\tilde{\Gamma}_{ij} = \phi(z_i)^T 1_l \phi(z_j) = kr(z_i, z_j)$

3.2 Deep quantum multilayer convolutional neural networks

Consider a multilayer hidden unit with ns distinct levels or states. When this happens, its transfer function can be expressed as a superposition of ns sigmoidal functions, each displaced by θ_r . Output of this multilevel unit may be expressed as $(1/n_s) \sum_{r=1}^{n_s} \text{sgm}(v^T x - \theta^r)$ where x is the input feature vector and v is connected weight matrix between input and hidden units in hidden layers. A QNN has one hidden layer with nh nodes, each of which represents a multilevel unit, ni inputs, and ni outputs. The output units may be sigmoid or linear. Let $k=1, \dots, m$ be kth feature vector of data set X, and let $x_k = [x_{1,k}, x_{2,k}, \dots, x_{n,k}]^T$. Feature vector is then used as the input to the hidden unit according to equation (43):

$$h_{j,k}^- = \sum_{l=0}^{n_l} v_{jl} x_{l,k}$$

$$h_{j,k}^{\tilde{}} = \frac{1}{n_s} \sum_{r=1}^{n_s} h_{j,k}^r = \frac{1}{n_s} \sum_{r=1}^{n_s} \text{sgm}(\beta_h (h_{j,k}^- - \theta_j^r)) \quad (43)$$

$$y_{i,k}^- = \sum_{j=0}^{n_h} w_{ij} h_{j,k} \quad (44)$$

$$\text{loss}(\vec{\theta} + \gamma \vec{g}, z) = \text{loss}(\vec{\theta}, z) + \gamma \vec{g}^2 + \mathcal{O}(\gamma^2) \quad (45)$$

The first thought is to eq(46) since we want to transfer the loss to where it is at zero

$$\gamma = -\frac{\text{loss}(\vec{\theta}, z)}{\vec{g}^2}$$

$$\frac{1}{2}(|z, 1\rangle + i\mathcal{U}(\vec{\theta})|z, 1\rangle|0\rangle) + \frac{1}{2}(|z, 1\rangle - i\mathcal{U}(\vec{\theta})|z, 1\rangle|1\rangle). \quad (46)$$

Now measure the auxiliary qubit. The probability to get 0 is given by eq. (47)

$$\frac{1}{2} - \frac{1}{2} \text{Im}(\langle z, 1 | \mathcal{U}(\vec{\theta}) | z, 1 \rangle) \quad (47)$$

Evaluate output error by eq. (46).

$$e_{i,k}^0 = (y_{i,k} - y_{i,k}^*) y_{i,k} (1 - y_{i,k}^k) \quad (46)$$

Evaluate hidden layer error term by eq. (47)

$$e_{j,k}^h = \left(\frac{1}{n_s} \sum_{r=1}^{n_s} h_{j,k}^r (1 - h_{j,k}^r) \right) \sum_{p=1}^{n_o} e_{p,k}^0 w_{pj} \quad (47)$$

$$w_{ij,k} = w_{l,k-1} + a c_{i,k}^0 \frac{1}{m_k} \sum_{r=1}^{n_k} \text{sgm}(\beta_h (h_{i,k}^- - 0_i^r)) \quad (48)$$

$$v_{jl,k} = v_{jl,k-1} + a \beta_h \left(\frac{1}{n_k} \sum_{r=1}^{n_k} h_{i,k}^r (1 - h_{i,k}^r) \right) x_{lk}$$

$$\begin{aligned} \langle h_{i,c_m}^{\tilde{m}} \rangle &= \frac{1}{|c_m|} \sum_{x_k: x_k \in c_m} h_{j,k}^{\tilde{k}} \\ \langle v_{j,c_m}^r \rangle &= \frac{1}{|c_m|} \sum_{x_k: x_k \in c_m} v_{j,k}^r \end{aligned} \quad (49)$$

$|c_m| =$ Cardinality of m^{th} class

Step 2: Evaluate quantum interval adjustment $\Delta\theta_q^r$ for each level by eq. (50):

$$\Delta\theta_j^r = \alpha_e \frac{\beta_h}{n_s} \sum_{m=1}^{n_e} \sum_{x_k \cdot x_k \in c_m} (\langle h_{j,c_m}^{\sim} \rangle - h_{j,k}) (\langle v_{j,c_m}^s \rangle - v_{j,k}^s) \quad (50)$$

Where α_0 is learning rate.

Step 3: Update jump-positions by eq. (51):

$$\theta_j^r = \theta\theta_j^r + \Delta\theta_j^r \quad (51)$$

The samples with the labels "+1" and "1" can be separated from the rest of the sample space. Think about the eq(52) states

$$|+1\rangle = N_+ \sum_{z:l(z)=1} e^{i,qz} |z, 1\rangle \quad |-1\rangle = N_- \sum_{z:l(z)=-1} e^{i,pz} |z, 1\rangle \quad (52)$$

For sequence-to-sequence learning tasks, algorithms employ convolutional neural networks and multilayer-perceptrons with a number of hidden layers. CNN has input, output, and a number of hidden layers, just like a typical DNN. In all convolution layers, approach employs ReLU activation tool. Max pooling layer spatially resizes input and operates separately for each row and column. By reducing input size to half of the actual input, pooling layer in CNN lessens overfitting issue. In Figure 3, a flowchart of both techniques is briefly described. Both methods use an ECG signal's characteristics as the network's input and forecast the output as labels for the signal. ECG datasets will first undergo pre-processing. To accomplish this, the first network first determines the characteristics and labels of the datasets it has read. The suggested architecture for the CNN in Figure 4 demonstrates how first and last convolutional layers differ from middle three convolutional layers in algorithm.

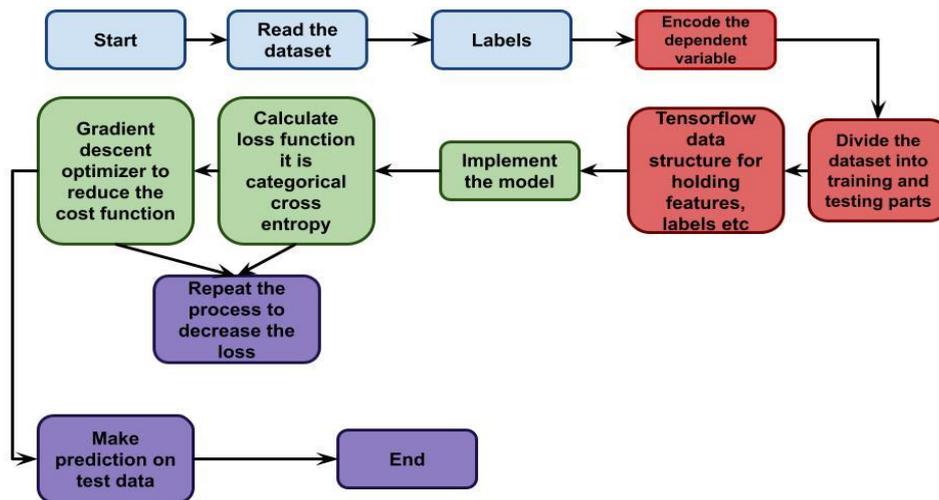


Figure 2. System process flowchart of Multilayer Perceptron (MLP) and Convolution NN

The final network structure used in this article is depicted in Figure 3 based on the comparison and analysis discussed above. 0 padding is used for each convolution layer to address the issue of pixels at the picture's corners being left out during each convolution operation, which results in loss of feature information for image edge. Finally, output network has 17,099.26 total parameters, which is a very low quantity when compared to DCNN.

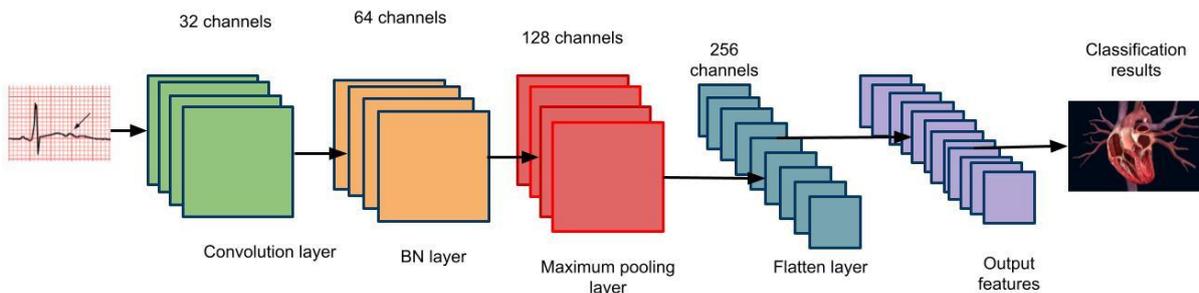


Figure 3. The MCNN structure adopted

The SGDM is selected as optimizer, and Nesterov momentum is used. Momentum parameter is set to 0.9, and learning rate is set to 0.1. Additionally, a learning rate reduction mechanism and an early stop mechanism are included. The patience value in this paper is set at 30 times. If loss function value from most recent training is not less than that from most recent training, learning rate will drop by 0.1 under present learning rate. 32 has been chosen as batch size. Proposed method is built using Keras, and training is finished using an NVIDIA GeForce 940MX graphics card.

4. Experimentation and Discussion

The results of numerous experiments used to evaluate effectiveness of suggested hybrid model are presented in this section. Proposed hybrid method was evaluated using a computer that had the following features: Windows 7 operating system, NumPy, SciPy, Pandas, Keras, and Matplotlib frameworks, as well as an Intel(R) Core (TM) i5-7500 CPU with a 32-bit operating system, 4 GB of RAM, and.

4.1 Dataset description

MIT/BIH arrhythmia database provided the data set for this investigation that we used. The BIH Arrhythmia Laboratory examined 48 entries in the data set. Two-channel ECG signals with a 30-min duration are included in each record and were chosen from the patients' 24-h recordings. The ECG signal has a 360Hz frequency. The database not only offers ECG signals but also timing data and heartbeat class data that have been cardiologist-verified. 44 records from the MIT/BIH arrhythmia database were chosen for use.

The four datasets that make up the Public Health Dataset, which was used for this study, are Cleveland, Hungary, Switzerland, and Long Beach V. The Public Health Dataset was established in 1988. It contains 76 properties, including the one that was predicted, but none of the published studies indicate utilising all of them; just a portion of 14 of them are mentioned. The "target" field makes reference to the patient's heart condition. One indicates an illness, whereas zero indicates none.

Table 1. Comparative analysis between proposed and existing technique

Parameter	Techniques	Accuracy	Precision	Recall	F_Score	SNR	RMSE
MIT/BIH arrhythmia	SVM	83	71	55	61	51	45
	LSTM	86	73	59	63	53	51
	AI_EHS_CDD	89	75	62	66	55	53
Public Health	SVM	91	77	63	69	57	55
	LSTM	93	79	66	71	58	59
	AI_EHS_CDD	95	81	69	73	59	62

The above table-1 shows comparative analysis between proposed and existing technique of heart disease dataset. Here parametric analysis is carried out in terms of accuracy, precision, recall, F1-score, SNR and RMSE. One metric for measuring classification model performance is accuracy. Informally, accuracy is percentage of predictions that our method correctly predicted. Accuracy is defined as follows in formal language: Accuracy is the quantity of accurate forecasts. sum of all projections. How frequently an algorithm successfully classifies a data point can be determined, for example, by looking at the accuracy of the algorithm. The percentage of projected data points that really occurred is known as accuracy. One measure of a machine learning model's effectiveness is precision, or the standard of a successful prediction the model makes. Ratio of overall number of true positives to total number of positive forecasts is known as precision. The percentage of Positive samples that were accurately labelled as Positive relative to all Positive samples is how the recall is calculated. How well the model can distinguish Positive samples is measured by recall. The recall increases as more positive samples are found. The F1-score combines precision and recall of a classifier into one metric by computing their harmonic means. It is mostly used to evaluate the performance of two classifiers. One of the methods for assessing the accuracy of forecasts is RMSE or

root mean square deviation. It displays the Euclidean distance between predicted values and measured true values.

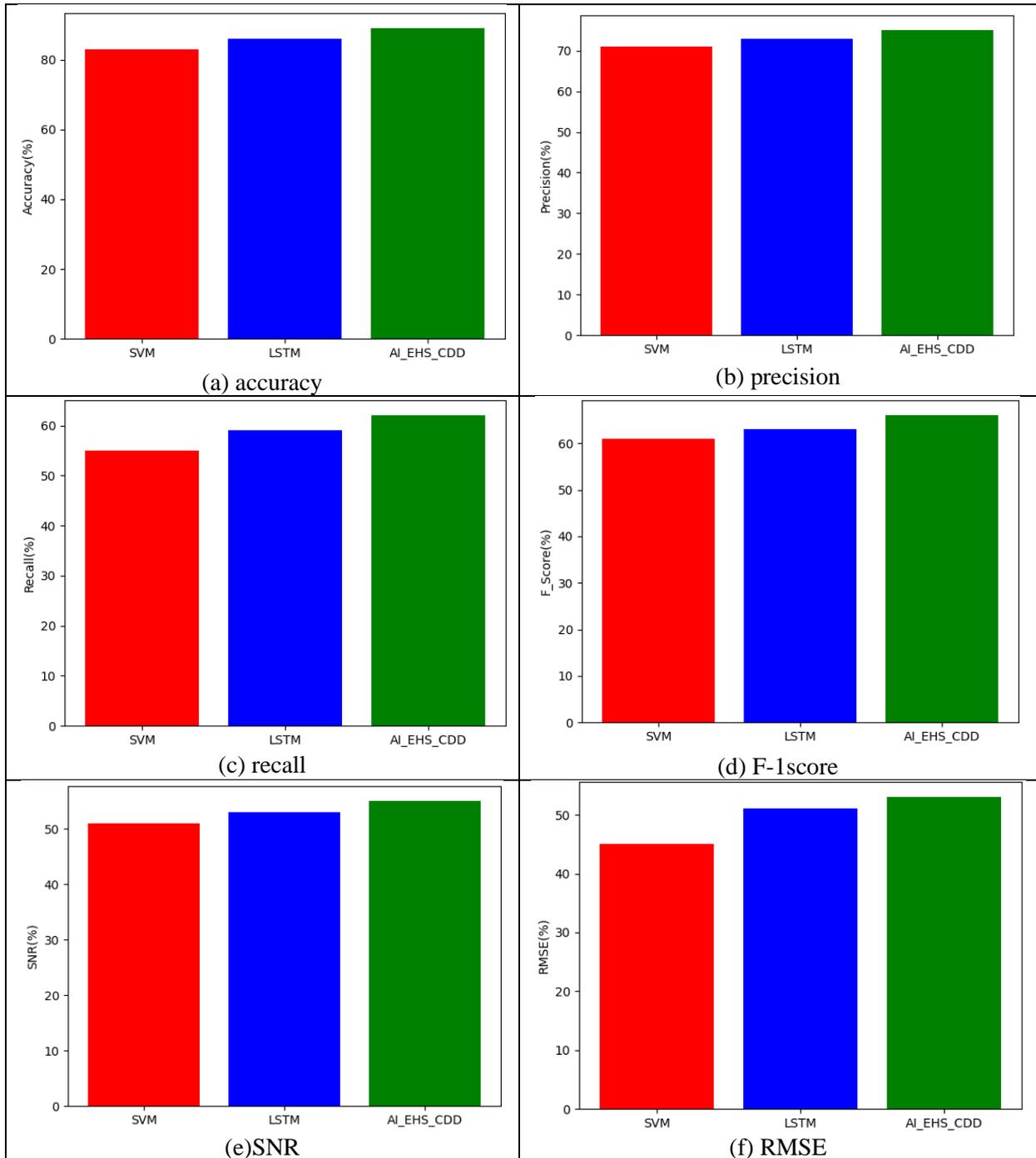


Figure 4. Comparative analysis between proposed and existing technique in terms of (a) accuracy, (b) precision, (c) recall, (d) F-1score, (e) SNR, (f) RMSE for MIT/BIH arrhythmia dataset

The above figure-4 shows comparative analysis between proposed and existing technique for MIT/BIH arrhythmia dataset. Here the proposed technique attained accuracy of 89%, precision of 75%, recall of 62%, F-1score of 66%, SNR of 55% and RMSE of 53%; while LSTM attained accuracy of 83%, precision of 71%, recall of 55%, F-1score of 61%, SNR of 51% and RMSE of 45%, SVM attained accuracy of 86%, precision of 73%, recall of 59%, F-1score of 63%, SNR of 53% and RMSE of 51%.

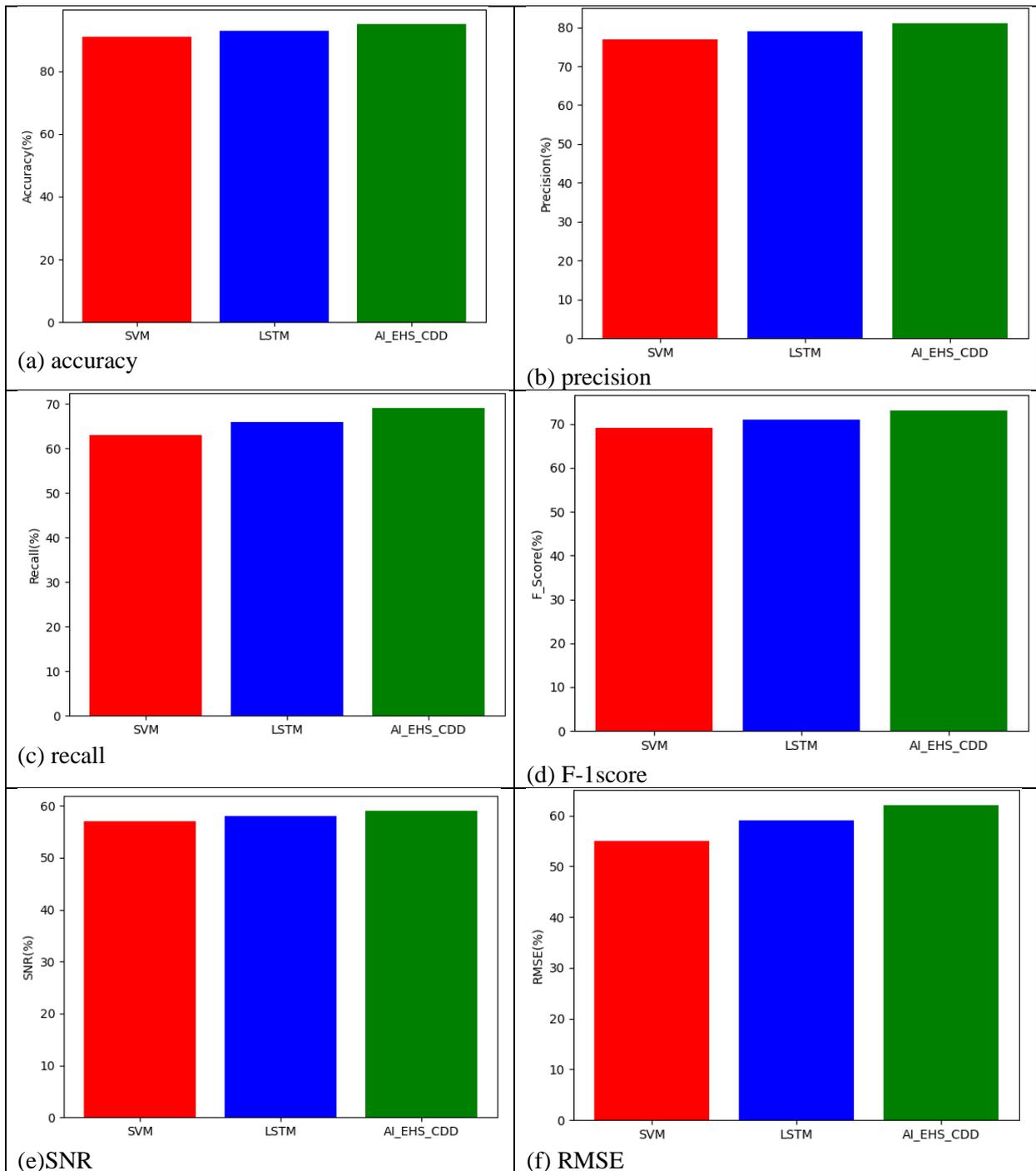


Figure 5. Comparative analysis between proposed and existing technique in terms of (a) accuracy, (b) precision, (c) recall, (d) F-1 score, (e) SNR, (f) RMSE for Public Health dataset

The above figure-4 shows comparative analysis between proposed and existing technique for Public Health dataset. For this dataset, the proposed technique attained accuracy of 95%, precision of 81%, recall of 69%, F-1 score of 73%, SNR of 59% and RMSE of 62%; while LSTM attained accuracy of 91%, precision of 77%, recall of 63%, F-1 score of 69%, SNR of 57% and RMSE of 55%, SVM attained accuracy of 93%, precision of 79%, recall of 66%, F-1 score of 71%, SNR of 58% and RMSE of 59%.

5. Conclusion

This research propose novel technique in efficient healthcare system by ECG wave based cardiac disease detection using deep learning architecture with high performance modelling. The processed ECG features has been extracted using deep belief kernel principal neural network. The experimental analysis has been carried out in terms of accuracy, precision, recall, F-1score, SNR and RMSE. the proposed technique attained accuracy of 95%, precision of 81%, recall of 69%, F-1score of 73%, SNR of 59% and RMSE of 62%. Detection method discussed in this article can also be used to identify other diseases. By using larger and more unbalanced datasets with DL methods, the robustness and other performance metrics can be determined.

References

- [1] Koo, J., & Klabjan, D. (2020, September). Improved classification based on deep belief networks. In International Conference on Artificial Neural Networks (pp. 541-552). Springer, Cham.
- [2] Errachdi, A., Slama, S., & Benrejeb, M. (2020). On the combination of kernel principal component analysis and neural networks for process indirect control. *Mathematical and Computer Modelling of Dynamical Systems*, 26(2), 144-168.
- [3] Abd Al-Majeed, G. H., Alisa, Z. T., & Naji, H. S. (2014). Data classification using quantum neural network. *J. Eng*, 20, 1-15.
- [4] Farhi, E., & Neven, H. (2018). Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002.
- [5] Wang, P., Lin, Z., Yan, X., Chen, Z., Ding, M., Song, Y., & Meng, L. (2022). A wearable ECG monitor for deep learning based real-time cardiovascular disease detection. arXiv preprint arXiv:2201.10083.
- [6] Nagavelli, U., Samanta, D., & Chakraborty, P. (2022). Machine Learning Technology-Based Heart Disease Detection Models. *Journal of Healthcare Engineering*, 2022.
- [7] Khan, A. H., Hussain, M., & Malik, M. K. (2021). Cardiac disorder classification by electrocardiogram sensing using deep neural network. *Complexity*, 2021.
- [8] Kusuma, S., & Udayan, J. D. (2020). Analysis on deep learning methods for ECG based cardiovascular disease prediction. *Scalable Computing: Practice and Experience*, 21(1), 127-136.
- [9] Zhang, X., Gu, K., Miao, S., Zhang, X., Yin, Y., Wan, C., ... & Liu, Y. (2020). Automated detection of cardiovascular disease by electrocardiogram signal analysis: a deep learning system. *Cardiovascular Diagnosis and Therapy*, 10(2), 227.
- [10] Yildirim, Ö., Pławiak, P., Tan, R. S., & Acharya, U. R. (2018). Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Computers in biology and medicine*, 102, 411-420.
- [11] Zhang, X., Jiang, M., Wu, W., & de Albuquerque, V. H. C. (2021). Hybrid feature fusion for classification optimization of short ECG segment in IoT based intelligent healthcare system. *Neural Computing and Applications*, 1-15.
- [12] Sadad, T., Bukhari, S. A. C., Munir, A., Ghani, A., El-Sherbeeney, A. M., & Rauf, H. T. (2022). Detection of Cardiovascular Disease Based on PPG Signals Using Machine Learning with Cloud Computing. *Computational Intelligence and Neuroscience*, 2022.
- [13] Nguyen, T. H., Nguyen, T. N., & Nguyen, T. T. (2020). A deep learning framework for heart disease classification in an IoTs-based system. In *A Handbook of Internet of Things in Biomedical and Cyber Physical System* (pp. 217-244). Springer, Cham.
- [14] Rath, A., Mishra, D., Panda, G., & Satapathy, S. C. (2021). An exhaustive review of machine and deep learning based diagnosis of heart diseases. *Multimedia Tools and Applications*, 1-59.
- [15] Siontis, K. C., Noseworthy, P. A., Attia, Z. I., & Friedman, P. A. (2021). Artificial intelligence-enhanced electrocardiography in cardiovascular disease management. *Nature Reviews Cardiology*, 18(7), 465-478.

- [16] Butun, E., Yildirim, O., Talo, M., Tan, R. S., & Acharya, U. R. (2020). 1D-CADCapsNet: One dimensional deep capsule networks for coronary artery disease detection using ECG signals. *Physica Medica*, 70, 39-48.
- [17] Somani, S., Russak, A. J., Richter, F., Zhao, S., Vaid, A., Chaudhry, F., ... & Glicksberg, B. S. (2021). Deep learning and the electrocardiogram: review of the current state-of-the-art. *EP Europace*, 23(8), 1179-1191.
- [18] Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments. *Future Generation Computer Systems*, 104, 187-200.
- [19] Dai, H., Hwang, H. G., & Tseng, V. S. (2021). Convolutional neural network based automatic screening tool for cardiovascular diseases using different intervals of ECG signals. *Computer Methods and Programs in Biomedicine*, 203, 106035.
- [20] Rincon, J. A., Guerra-Ojeda, S., Carrascosa, C., & Julian, V. (2020). An IoT and fog computing-based monitoring system for cardiovascular patients with automatic ECG classification using deep neural networks. *Sensors*, 20(24), 7353.