Evolving Systems
https://doi.org/10.1007/s12530-022-09429-1

ORIGINAL PAPER

A random deep neural system for heartbeat classification

Roguia Siouda[1] · **Mohamed Nemissi**[2] · **Hamid Seridi**[1]

Received: 6 May 2021 / Accepted: 7 February 2022

Abstract
This paper introduces a heartbeat classification system that combines three types of neural networks: random neural networks, deep autoencoders and RBF neural networks. The aim is to make use of the advantages of these neural networks in order to introduce a model with simpler architecture than the state-of-the-art deep models. Indeed, the advantages of the three combined networks, briefly, are these: (i) Autoencoders provide high level features without pre-processing; (ii) Random neural networks provide good generalisation and very fast training; (iii) RBF neural networks provide high coverage of the input space and allow using prior knowledge. On the other hand, two types of features are used: coded features (obtained from the autoencoder) and RR interval based-features. To evaluate the performance of the proposed system, we conduct experiments on the MIT-BIH arrhythmia dataset and we consider the recommendations of the association for the advancement of medical instrumentation, which defines five classes of interest. Furthermore, the experiments are based on an inter-patient paradigm and the obtained results are compared with some of the state-of-the-art methods.

Keywords ECG arrhythmia · Radial basis function neural network · Random vector functional link neural network · Sparse autoencoder · Multi-class classification

1 Introduction

The electrocardiogram (ECG) signal is widely used as one of the most important clinical tools for evaluating the cardiac state of patients. However, the non-linear, non-stationary nature of the signal and the noise hinder its automatic processing. The need to overcome these obstacles has led to the use of various methods and techniques for its analysis. Generally, ECG heartbeat classification systems include three main steps: data pre-processing, feature extraction and classification. The heartbeat classification methods can be divided into two main categories depending on how these three steps are performed. The first category includes traditional methods in which feature extraction and classification are performed in separate steps. Whereas, the second category includes methods based on deep learning approach, in which the feature extraction and classification are automatically performed in the same architecture.

In the first category, both the extracted features and the classifier model have an important impact on the overall performance. The most common features of a single ECG beat are the amplitudes, morphologies, durations of individual waves (P, Q, R, S, and T), intervals (RR, PP, PR, ST, ⋯) and segments (PR and ST). In recent years, various feature extraction techniques have been developed. This includes principal component analysis (PCA), linear discriminant analysis algorithms (LDA), wavelet transform (WT)..., etc (Martis et al. 2013). On the other hand, with regard to the classification step, artificial neural networks (ANNs) have been widely used for this task. This is due to their learning capabilities, approximation abilities and high predictive power (Soares et al. 2020; Angelov et al. 2017). Among ANNs, radial basis function neural networks (RBFNNs) Roguia and Mohamed (2019) have been successfully applied.

✉ Roguia Siouda
siouda.roguia@univ-guelma.dz

Mohamed Nemissi
nemissi.mohamed@univ-guelma.dz

Hamid Seridi
seridi.hamid@univ-guelma.dz

[1] LabSTIC Laboratory, Department of Computer Sciences, University of 8 Mai 1945, PB 401, Guelma, Algeria

[2] LabSTIC Laboratory, Department of Electronics and Telecommunications, University of 8 Mai 1945, PB 401, Guelma, Algeria

Published online: 19 February 2022

The second category includes deep learning-based methods, which yield to automatic extraction of effective features directly from the raw input data. These methods avoid time-consuming and complex feature extraction phases. The deep learning approaches have been successfully applied in various steps of the ECG analysis systems, such as dimensional reduction, feature extraction and classification. Among the most used deep models in heartbeat classification: convolutional neural network (CNN) (Ji et al. 2019; Mousavi and Afghah 2019; Zhou and Tan 2020) and sparse autoencoders (SAE) (Yang et al. 2015, 2018; Nurmaini et al. 2020; Siouda et al. 2020, 2021). However, these methods have two major issues. First, they have large architectures that include a lot of hidden layers and neurons. Second, they may ignore some important features like RR interval features.

In this work, we propose a heartbeat classification system based on combining both categories, i.e., conventional methods and deep learning approaches. The proposed system consists of a deep SAE as a feature extractor and a hybrid neural model as a classifier. The latter combines random neural networks and RBFNN. In addition, two types of features are used, i.e., coded features (obtained using SAE) and RR-interval based-features. To solve the problem of imbalanced data, which occurs in most medical problems, we use the synthetic minority over-sampling technique (SMOTE) Chawla et al. (2002). For the evaluation, we use the well-known MIT-BIH arrhythmia dataset and we apply the recommendations of the association for the advancement of medical instrumentation (AAMI), which outlines five classes: normal (N), ventricular (V), supraventricular (S), fusion of normal and ventricular (F), and unknown beats (Q). The AAMI further recommends the adoption of the inter-patient paradigm, i.e., the training and test beats should be taken from different patients De Chazal et al. (2004). Accordingly, we perform inter-patient tests and compare our results with some of the state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 gives some backgrounds concerning the sparse autoencoder and RVFLN. Section 4 details the principles of the proposed model and discusses its implementation. Section 5 reports and analyses the experimental results conducted on MIT-BIH dataset. Finally, Sect. 6 concludes this paper.

2 Related work

The main aim of this paper is to combine the advantages of conventional methods and deep learning techniques to introduce an efficient deep system with simpler architecture than existing models. More precisely, we aim to introduce a system based on automatic features combined with only some important handcrafted features, namely RR interval features. In this context, Shi et al. (2020) combine automatic features and seven RR interval features. They proposed a deep structure based on CNN and long short-term memory (LSTM) network. A heartbeat is first divided into three regions. Then, the CNN is used to automatically extract features from the three inputs and the obtained sequence goes through an LSTM network. The output of LSTM is concatenated with the RR interval features, and a fully connected layer is used for outputting the final result. The deep neural network consists of nine layers: the convolution layer, pooling layer, two concatenated layers, the LSTM layer, flattening layer, and three fully-connected layers. Each layer has many units. Wang et al. (2020a) combine automatic features with four RR intervals. The automatic features are extracted by one-dimensional (1D) CNN and concatenated with the RR intervals features. The obtained set of features is used as the input of the multi-layer perceptron (MLP). The MLP hidden layer includes 64 neurons. The CNN architecture contains three convolutional blocks and three pooling layers. Each convolutional block includes a convolution layer, a ReLU activation function, and a batch normalization layer. These two methods performed well; however, they have many layers and a large number of neurons in each layer, and they use many RR-interval features. The optimization of these networks became more difficult as the layer size increased. In this work, we propose a simpler architecture based on a stacked autoencoder with only one hidden layer and a hybrid neural classifier that also consists of one hidden layer. In the proposed system, we combine automatic features with only two RR intervals. To the best of our knowledge, extracted features from ECG beats using autoencoders haven't been combined with RR interval features in previous works. The stack autoencoders have been widely used in extracting features from ECG Beats (Yang et al. 2015, 2018; Nurmaini et al. 2020; Siouda et al. 2021). Note that all these methods are based on autoencoders with multiple hidden layers.

On the other hand, we propose, as a classifier, a hybrid RBF-random neural network in order to achieve better generalization capability and faster learning speed. Therefore, the entire system consists of an AE to generate automatic features and a hybrid neural classifier to give the final decision. Indeed, the AE has been successfully combined with random neural networks in several applications (Zhang et al. 2019; Katuwal and Suganthan 2019; Nayak et al. 2020; Sun et al. 2017; Tang et al. 2015). In this work, we use AE jointly with a hybrid RBF-RVFLNN and apply them to arrhythmia classification. The motivation behind the proposition of this system is to achieve high generalization capability with a simple structure in the inter-patient paradigm, in which the training set and test set are from distinct patients.

3 Theoretical background

3.1 Sparse autoencoder (SAE)

An autoencoder is a neural network trained with an unsupervised learning algorithm to provide target values equal to the inputs ($y(i) = x(i)$). It is designed to learn a function $G(x) \approx x$; thus, the output $\hat{x}$ is similar to the input x.Consequently, interesting information can be found using a limited number of hidden neurons. An autoencoder can be considered as a specific type of DL, which is used to automatically learn potential features from unlabelled samples. Fig. 1 gives an example of an autoencoder with one hidden layer.

The sparse autoencoder (SAE) aims to learn sparse features with the addition of a sparse penalty term inspired from sparse coding (Ng 2011). This penalty is added to the cost function to ensure that the learned features are not just a simple repetition of inputs. The sparse penalty aims to minimize the number of "active" hidden neurons. Consider that $a_j(x)$ is the activation of the *jth* hidden neuron, the average activation of this neuron is given by:

$$\rho_j = \frac{1}{n}\sum_{i=1}^{n}[a_j(x(i))] \tag{1}$$

where, n is the dimension of the feature space.

The sparsity can be achieved through the addition of a regularization term that indicates the difference between the mean activation value, $\widehat{\rho}_j$, and a sparsity target value, ρ. This can be done using the Kullback-Leibler discrepancy as follows:

$$\Omega_{spar} = KL(\rho \parallel \widehat{\rho}_j) = \rho log\frac{\rho}{\widehat{\rho}_j} + (1-\rho)log\frac{1-\rho}{1-\widehat{\rho}_j} \tag{2}$$

The cost function can be given as follows:

$$f = MSE(X - \widehat{X}) + a\Omega_{spar} + \beta\Omega_w \tag{3}$$

where, $MSE(X - \widehat{X})$ is the mean squared error and Ω_w is the sum squared of all network weights.

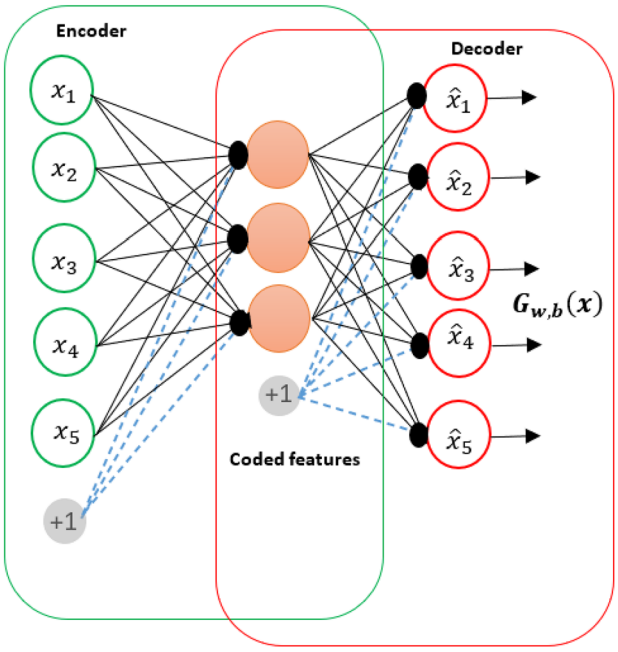


Fig. 1 An example of sparse autoencoder with five inputs and three hidden neurons

3.2 Random vector functional link network (RVFLN)

Random neural networks (RNNs) constitute powerful machine learning techniques due to their high computing and fast learning abilities in processing big data (Zhang and Suganthan 2016). The random vector functional link network (RVFLN) is one of the most important RNNs (Pao, Y- H, Takefuji, Y, 1992). In RVFLN, the input weights and biases are randomly initialized and the output weights are analytically calculated. In these networks, the direct link between the input and output layers is a simple yet efficient regulation technique that prevents over-adjustment. The structure of the RVFL neural network is shown in Fig. 2. The dashed lines represent the direct links from the input to the output layer.

Considering a dataset of training samples $\{(X_q, T_q), q = 1 : Q\}$, where, $X_q = [x_{q1}, x_{q2}, ..., x_{qN}]^T \in \mathcal{R}^N$ and $T_q = [t_{q1}, t_{q2}, ..., t_{qJ}]^T \in \mathcal{R}^J$.

An RVFL network with M enhancement neurons can be formulated as the following quadratic optimization problem.

$$\arg\min_{\mathbb{U}} \|\mathbb{S}\mathbb{U} - \mathbb{T}\|^2 \tag{4}$$

where, $\mathbb{U}$ is the output weight matrix, $\mathbb{T}$ is the matrix of targets and $\mathbb{S}$ is the matrix that concatenates the input and the outputs of the enhancement neurons.

The matrices $\mathbb{T}$, $\mathbb{S}$ and $\mathbb{U}$ are given by:

$$\mathbb{T} = \begin{bmatrix} T_1^T \\ \vdots \\ \vdots \\ T_Q^T \end{bmatrix}_{Q\times J} \tag{5}$$

$\mathbb{S} = [\mathbb{X}\mathbb{H}]$ where,

$$\mathbb{X} = \begin{bmatrix} x_{11} & \dots & x_{1N} \\ \vdots & \dots & \vdots \\ x_{Q1} & \dots & x_{QN} \end{bmatrix}_{Q\times N}$$

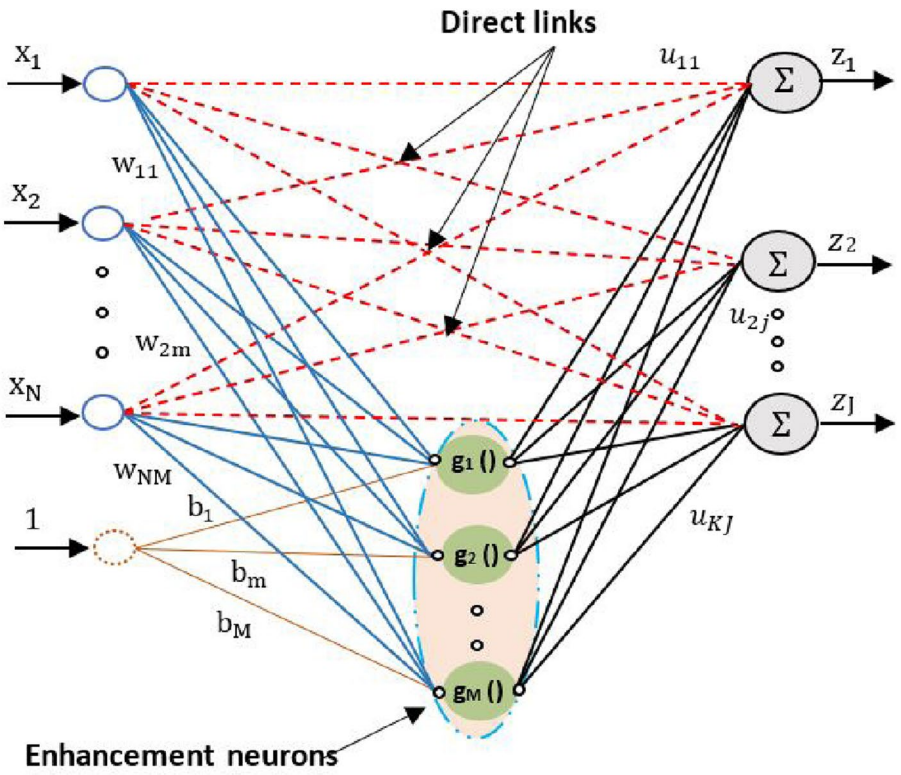


Fig. 2 An example of RVFLN with N inputs, M enhancement neurons and J output neurons. The input features are first transformed to the enhancement neurons through the input weights, which are randomly generated. All the original and improved features are connected to the outputs

$$\mathbb{H} = \begin{bmatrix} g(W_1.X_1 + b_1) & \dots & g(W_M.X_1 + b_M) \\ \vdots & \dots & \vdots \\ g(W_1.X_Q + b_1) & \dots & g(W_M.X_Q + b_M) \end{bmatrix}_{Q \times M} \tag{6}$$

Here, g is the activation function of the enhancement neurones, $W_m = [w_{1m}, w_{2m}, ..., w_{Nm}]$ and b_m are the input weights and the bias corresponding to the m^{th} hidden neuron.

$$\mathbb{U} = \begin{bmatrix} U_1^T \\ \vdots \\ \vdots \\ U_K^T \end{bmatrix}_{K \times J} \tag{7}$$

where, $U_k = [u_{k1}, u_{k2}, ..., u_{kJ}]^T$ is the vector weight connecting the *kth* hidden neuron (or input) to the output layer.

These output weights can be then analytically calculated using the minimum norm least square method by: Rao et al. (1972)

$$\mathbb{U} = \mathbb{S}^{\ddagger}\mathbb{T} \tag{8}$$

where, $\mathbb{S}^{\ddagger}$ is the Moore-Penrose generalized inverse of the matrix $\mathbb{S}$.

4 The proposed method

4.1 Main idea and motivations

Deep learning methods permit automatic extraction of features from raw data. These methods avoid time-consuming and complex phases of feature extraction encountered in conventional methods. However, deep models have two major problems. First, they have large architectures that include a lot of hidden layers and neurons. Second, they ignore some important key features like RR interval features. In this paper, we propose a system that tackles these two problems. This system has a simple architecture based on a stacked autoencoder with only one hidden layer and a hybrid neural classifier which also consists of one hidden layer. The proposed system combines automatic features and handcrafted features. The automatic features are extracted using SSAE. The used hand-crafted features are based on RR intervals, which are very effective and can be obtained easily.

As a classifier, we propose a hybrid RBF-RVFL neural network. The motivations behind combining RBF and RVFL neural networks are as follows:

First, RBFNN is based on local responses while RVFLN is based on global responses. Their combination can provide better decision boundaries using fewer hidden neurons and simpler structure.

Second, the output weights in both RVFLN and RBFNN can be directly calculated in one-step. In the proposed hybrid neural classifier, the direct and output weights of RVFLN are concatenated with the output weights of RBFNN. All These weights are then calculated in one step using the moore-penrose pseudo-inverse solution. This allows very fast learning.

Third, both RBFNN and RVFLN provide nonlinear mapping except the direct links in RVFLN, which provide linear mapping. The combination of these two neural networks allows them to map data in both linear and nonlinear manners.

Fourth, RVFLN is randomly initialized while RBFNN can be initialized using clustering methods. Their combination can then reduce the effect of the random initialisation and allow using prior knowledge. In the proposed model, the K-means method is used for initialising the RBF centres. This clustering method is used in a supervised manner, i.e., applied to each class separately, in order to get more accurate RBFs (Salah et al. 2019). The widths of RBFs are subsequently calculated using the P-nearest neighbours' method (Benoudjit et al. 2002).

4.2 Process

Fig. 3 illustrates the principal scheme of the proposed system. First, the R peaks are detected from the original ECG signal. Second, two types of features are calculated: the RR intervals and the coded features. The latter are obtained using the stacked sparse autoencoder. Third, all features are concatenated and the synthetic minority over-sampling technique (SMOTE) is used to alleviate the problem of imbalanced data. Finally, the hybrid RBF-RVFL neural network is used for the classification.

4.2.1 ECG beats segmentation

In order to extract ECG beats from a given ECG signal, two steps are performed:

(i) Find the R peaks of the ECG signal.
(ii) Divide the ECG signal into a series of heartbeats, in which each one consists of 260 samples : 99 samples before the R-peak, 160 samples after the R-peak and the R-peak itself (Yildirim et al. 2019; Murat et al. 2020).

4.2.2 RR interval features

The RR intervals are calculated based on the time between successive beats. In this work, two RR intervals are used as dynamic features:

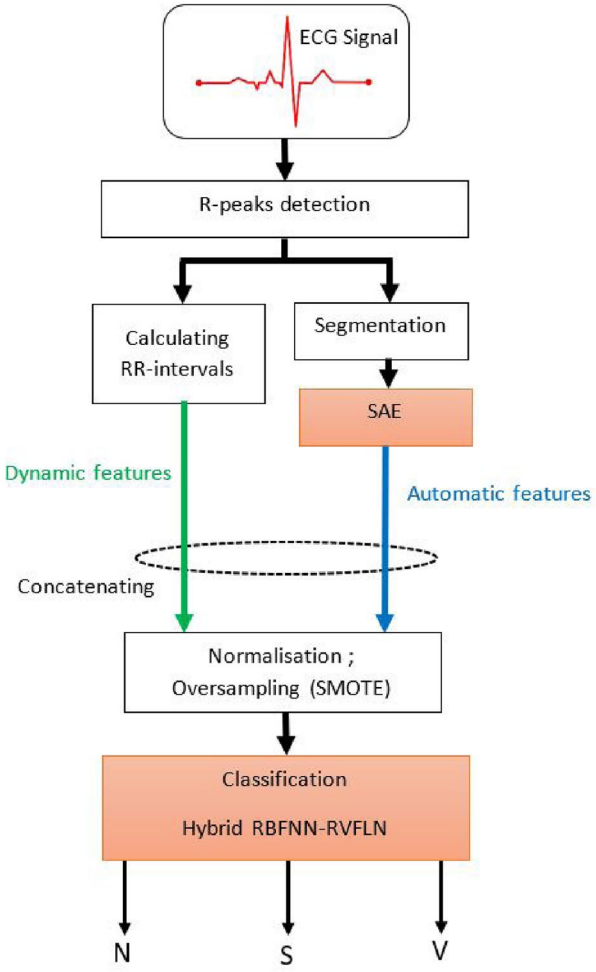


Fig. 3 A simple block diagram summarizing the stages of the proposed system

$$RRprevious(i) = [R(i) - R(i-1)] \times 360 \tag{9}$$

$$RRposterior(i) = [R(i+1) - R(i)] \times 360 \tag{10}$$

where, the *RRprevious* indicates the distance between the current heartbeat and the previous one; the *RRposterior* indicates the distance between the current heartbeat and the next one; $R(i)$ is the position of the *ith* R-peak; 360 is the frequency of the ECG signal.

4.2.3 Training

The training of the proposed system is performed in two steps: (i) Training the sparse autoencoder using unsupervised data; (ii) Training the hybrid neural network using supervised data.

The training formulas of the SAE are given in Section 3.1. However, there is a lack of methods for designing

the appropriate structure, i.e., the number of hidden layers and the number of features. To deal with this issue, we perform tests with several structures. For each structure, we use a validation set, taken from the training data, to evaluate the obtained coded features. The evaluation is performed by calculating the mean square error between the original and reconstructed heartbeats.

The training of the hybrid classifier is given in the next section.

4.3 Architecture and training of the hybrid neural classifier

The proposed classifier combines three neural networks: deep SSAE, RBFNN and RVFLN. The SSAE is used to extract coded features from ECG signals. These features and other RR interval based-features are used as input to the hybrid RBFNN-RVFLN classifier. In this classifier, the coded features are connected to the RBFNN while the RR intervals are connected to the RVFLN.

Fig. 4 illustrates an example of the proposed classifier with $(N + P)$ inputs, D RBF neurons, M enhancement neurons and J outputs. Every input sample is represented by a concatenation of two vectors: a vector with N coding features ($X = [x_1, x_2, ..., x_N]$) and a vector with P dynamic features ($XX = [xx_1, xx_2, ..., xx_P]$).

The different parameters of the proposed classifier are set as follows:

1. The weights linking the inputs (coding features) to the RBF neurons are equal to '1'
2. The centres of the RBFs are calculated using the K-means clustering method. We use this method in a supervised way, i.e., applied to each class separately, in order to get more accurate RBFs (Salah et al. 2019). The widths of RBFs are subsequently calculated using the P-nearest neighbours' method (Benoudjit et al. 2002). The width of the d^{th} hidden neuron is given by:

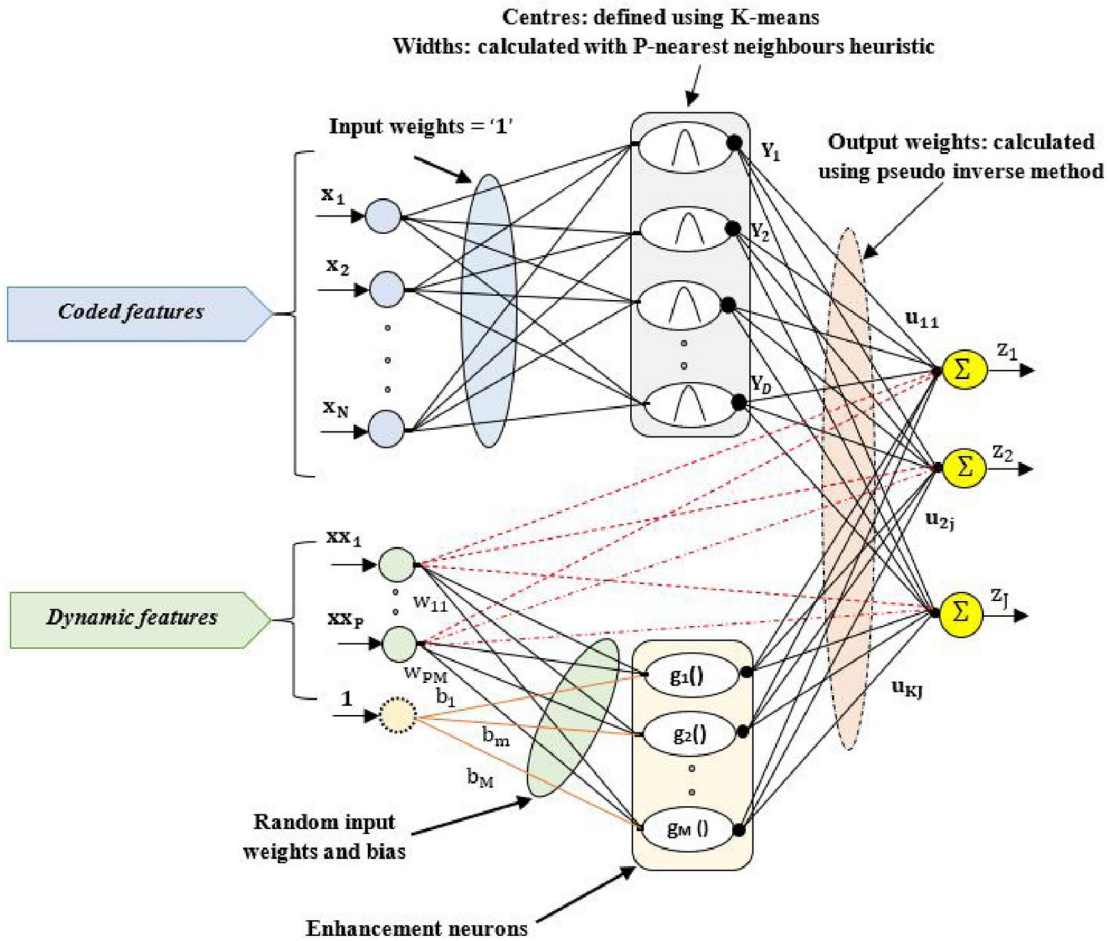


Fig. 4 The structure of proposed hybrid neural classifier (RBFNN-RVFLN)

$$\sigma_d = \frac{1}{P}\left(\sum_{n=1}^{P} \parallel V_n - V_d \parallel^2\right)^{\frac{1}{2}} \tag{11}$$

where, $V_n (n = 1...P)$ are the P-nearest neighbours' of the centre $V_d = [v_{d1}, v_{d2}, ..., v_{dN}]$.

The RBF neurons are linked with the vector of the coded features (X) . The output of d^{th} RBF neuron is then given by:

$$Y_d(X) = \exp\left(-\frac{\parallel X - V_d \parallel^2}{2\sigma_d^2}\right) \tag{12}$$

where, $V_d = (v_{1d}, v_{2d}, ..., v_{Nd})$ and σ_d are the center and the width corresponding to the d^{th} RBF.

3. The input weights linking the input to the enhancement neurons $([w_{im}], i = 1...P, m = 1...M)$ are randomly set.
4. The biases $(b_m, m = 1...M)$ are randomly set.
5. The enhancement neurons are linked with the vector of dynamic features (XX) . The outputs of the enhancement neurones are then given by the vector:

$$H = [g(W_1 * XX + b_1), g(W_2 * XX + b_2)...g(W_M * XX + b_M)]^T \tag{13}$$

where, g is the activation function, $W_m = [w_{1m}, w_{2m}, \ldots, w_{Pm}]$ and b_m are the vectors of input weights and the bias corresponding to the m^{th} enhancement neuron.

6. The output vector of the classifier is given by:
   $Z = S \times U$where, S is the vector that contatenate the vectors of RBF outputs (Y) , dynamic features (XX) and enhancement neurons (H) :
   $S = [YXXH]$
7. The matrix of the output weights $(\mathbb{U})$ is calculated in one-step using all training samples. Consider a training dataset with Q training samples, in which every sample is represented by a vector (X) with N coding features and a vector (XX) with P dynamic features. The feature vectors corresponding to the q^{th} sample are: $X_q = [x_{q1}, x_{q2}, ..., x_{qN}]$ and $XX_q = [xx_{q1}, xx_{q2}, ..., xx_{qP}]$.
   The qth target vector: $T_q = [t_{q1}, t_{q2}, ..., t_{qJ}]$.
   The matrix of the entire targets is given by the matrix $\mathbb{T}$ as follows:

$$\mathbb{T} = \begin{bmatrix} t_{11} & \ldots & t_{1J} \\ \vdots & \ldots & \vdots \\ t_{Q1} & \ldots & t_{QJ} \end{bmatrix}_{Q\times J} \tag{14}$$

Let $\mathbb{S}$ be the matrix concatenating the dynamic inputs, the RBF outputs and the enhancement outputs. $\mathbb{S}$ is then given by:

$$\mathbb{S} = [\mathbb{Y}\mathbb{X}\mathbb{X}\mathbb{H}]_{Q\times(D+P+M)} \text{where ,}$$

$$\mathbb{X}\mathbb{X} = \begin{bmatrix} xx_{11} & \ldots & xx_{1P} \\ \vdots & \ldots & \vdots \\ xx_{Q1} & \ldots & xx_{QP} \end{bmatrix}_{Q\times P},$$

$$\mathbb{H} = \begin{bmatrix} g(W_1.XX_1 + b_1) & \ldots & g(W_M.XX_1 + b_M) \\ \vdots & \ldots & \vdots \\ g(W_1.XX_Q + b_1) & \ldots & g(W_M.XX_Q + b_M) \end{bmatrix}_{Q\times M} \text{and}$$

$$\mathbb{Y} = \begin{bmatrix} y_1(X_1) & \ldots & y_D(X_1) \\ \vdots & \ldots & \vdots \\ y_1(X_Q) & \ldots & y_D(X_Q) \end{bmatrix}_{Q\times D} \tag{15}$$

The matrix of the outputs corresponding to all training samples is given by:

$$\mathbb{Z} = \mathbb{S} \times \mathbb{U} \tag{16}$$

Finally, the output weights $(\mathbb{U})$ are simply calculated in one-step by:

$$\mathbb{U} = \mathbb{S}^{\ddagger}\mathbb{T} \tag{17}$$

where, $\mathbb{S}^{\ddagger}$ is the moore-penrose generalized inverse of $\mathbb{S}$.

5 Experiment results

5.1 Dataset description

To evaluate the performance of the proposed classifier, we used the physionet MIT-BIH arrhythmia dataset (PhysioNet 2001; Moody and Mark 2001). This dataset includes ECG signals collected at a sampling rate of 360 Hz for 48 independent patients. The signal length is 650000 samples for 30 minutes. There are two ECG leads in each record: lead II and lead V1. We use lead II as it is commonly used in literature. This dataset is recommended by the american association of medical instrumentation (AAMI) AAMI (1998), as it contains the five major types of arrhythmias described in Table 1.

The experimental analysis is performed on the MIT-BIH dataset according to the inter-patient protocol proposed by De Chazal et al. (2004), which has been widely adopted in the literature (De Chazal et al. 2004; Luo et al. 2017). Under this protocol, the dataset (44 records according to the AAMI) is divided into two groups: DS1 = 101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203,

Table 1 Types of heartbeats in the MIT-BIH dataset recommended by AAMI

AAMI heartbeats	MIT-BIH heartbeats
Normal (N)	Normal beat (N); Left and right bundle branch block beats (L, R); Atrial escape beat (e); Nodal (junctional) escape beat (j)
Supraventricular ectopic beat (S)	Atrial premature beat (A); Aberrated atrial premature beat (a); Nodal (junctional) premature beat (J); Supraventricular premature beat (S)
Ventricular ectopic beat (V)	Premature ventricular contraction (V); Ventricular escape beat (E)
Fusion (F)	Fusion of ventricular and normal beat (F)
Unknown beat (Q)	Paced beat (/); Fusion of paced and normal beat (f); Unclassifiable beat (U)

Table 2 The number of samples in each class

Number of heartbeats		
Heartbeat class	DS1	DS2
N	45824	44218
S	943	1836
V	3788	3219
F	414	388
Q	8	7
Total	50977	49668

205, 207,208, 209, 215, 220, 223,230 and DS2 = 100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219,221, 222, 228, 231, 232, 233, 234. DS1 is used for the training, while DS2 for the test. With this separation, it is not possible to have heartbeats of the same patient in both training and test sets. The numbers of heartbeats in DS1 and DS2 data sets are listed in Table 2 by class.

5.2 Metric evaluation

The measures recommended by AAMI for the classification of arrhythmia are: overall accuracy (Acc), specificity (Spe), sensitivity (Se), positive predictive (+P) and false positive rate (FPR). In this study, we use the most commonly used measures, i.e.: overall accuracy (Acc), specificity (Spe), sensitivity (Se), positive predictive (+P) and F1-score. These measures are indicated as follows:

Accuracy is the most common performance measure, expressed as the ratio of correct predictions to the total number of input samples. Mathematically, it is given by:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{18}$$

Specificity measures the model's ability to make true negative predictions for each available category. It is the fraction of the correctly predicted negative samples among the true negative and false positive samples within the class. The specificity determines how many true negative labeled beats were found by the model. Mathematically, it is given by:

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{19}$$

Sensitivity measures the fraction of correctly predicted positive samples among the true positive and false negative samples within the class. The sensitivity determines how many true positive labeled beats were found by the model. Mathematically, it is given by:

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{20}$$

Positive predictivity is defined as the quotient of true positive samples and the total predicted positive samples. The high precision percentage corresponds to a low false positive rate. Mathematically, it is given by:

$$Positive\ predictive = \frac{TP}{TP + FP} \times 100\% \tag{21}$$

The F1-score is a key metric to estimate the balance between sensitivity and positive predictivity metrics. In the case of unbalanced classes, the F1 score is the most suitable metric to evaluate the performance of the classification model. Mathematically, it is given by:

$$F1 - score = \frac{2 * TP}{2 * TP + FP + FN} \times 100\% \tag{22}$$

where TP: true positive, FN: false negative, TN: true negative, and FP: false positive.

5.3 Experiments

In our experiments, we have considered only three classes (N, S and V). Indeed, the classes F and Q are not represented by a large number of beats and they have been ignored in several studies (Garcia et al. 2017).

First, we performed several tests on SSAEs in order to define the best structure. We used a validation set taken from the training data. The DS1 group is then divided into two new subgroups, DS11= 101, 106, 108, 109, 114, 115, 116, 119, 122, 209, 223 and DS12= 112, 118, 124, 201, 203, 205, 207, 208, 215, 220, 230. DS11 and DS12 are used for

Table 3 The mean square error between the original and reconstructed ECG heartbeats of some SSAEs structures over MIT-BIH arrhythmia dataset

#Features	Structures of SSAEs	Validation set (MSE)	Test set (MSE)
10	1 AE; 260-10	0.0048	0.0066
	2 SSAEs; 260-100-10	0.0076	0.0091
	3 SSAEs; 260-160-60-10	0.00812	0.0099
30	**1 AE; 260- 30**	**0.0013**	**0.0015**
	2 SSAEs; 260-100-30	0.0044	0.006
	3 SSAEs; 260-160-60-30	0.0056	0.0078
50	1 AE; 260- 50	0.0014	0.0016
	2 SSAEs; 260-130-50	0.0039	0.0053
	3 SSAEs; 260-160-80-50	0.0054	0.0073

Bold values indicate the best results

training and evaluation respectively. This division is adopted from the work of Garcia et al. (2017).

Table 3 presents the mean square error between the original and reconstructed ECG heartbeats corresponding to different structures. We notice that one layer of deep SAE and thirty features provide the best results.

Figure 5 illustrates an example of ECG beat (from class N) and its corresponding coded features. In this example, the original signal is encoded using 30 features. These features reconstruct the original beat efficiently.

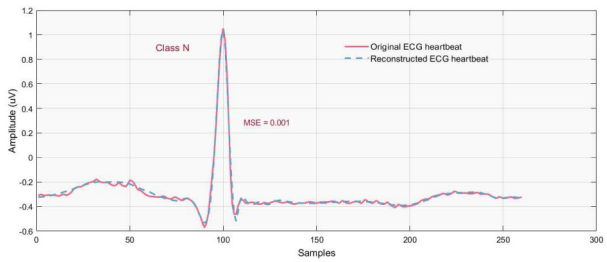
Table 4 Performance of the proposed system on MIT-BIH arrhythmia dataset

Overall accuracy (%)		93.11	
Classes	Sensitivity(%)	+predictive(%)	F1-score(%)
N	99.14	93.78	96.39
S	13.27	69.41	22.28
V	56.13	82.57	66.82
Average	56.18	81.92	61.83

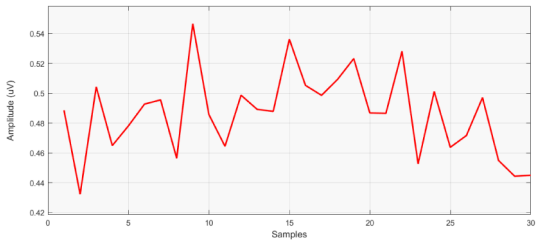
Consequently, the chosen architecture of the SSAE is: one hidden layer and thirty features. The proposed system is then trained using all the training data.

Figure 6 presents the confusion matrix obtained for the test data. This matrix represents the correct and incorrect predictions of all heartbeats. Each row represents the actual class, and each column represents the predicted class.

Table 4 details the obtained results: sensitivity (Se), positive prediction (+ P) and F1 score (%) of each class, as well as the average value of each metric. According to Table 4, the lowest performance corresponds to class S. This can be explained by the fact that this class is represented by a very small number of training samples. The overall accuracy of the proposed classifier is 93.11%, and the average sensitivity,



(a) Original and reconstructed ECG beats



(b) The coded features

Fig. 5 An example of ECG signal coding using SAE (class N)

Fig. 6 The confusion matrix obtained using the proposed system over MIT-BIH arrhythmia dataset

Predicted class

Actual class	Classes	N	S	V	Total
	N	43817	60	318	44195
	S	1536	245	65	1846
	V	1370	48	1814	3232
	Total	46723	353	2197	49273

positive prediction and F1-score are 56.18%, 81.92% and 61.83%, respectively.

Table 5 compares the obtained results with other works applied to the same dataset, i.e., MIT-BIH arrhythmia. These works include a large variety of classification methods and feature extracting techniques. From Table 5, we notice that the performance of our classifier outperformed all these works, except the two works of Shi et al. (2020) and Wang et al. (2020b). Although these two works provided better results, our system is much simpler as it is based only on one layer of deep SAE and it considers a smaller number of features. Indeed, the method in the first work (Shi et al. 2020) is based on a deep network, which consists of multiple layers, including a convolutional layer, a pooling layer, an LSTM layer, a concatenated layer and fully connected layers. This method is based on automatic features, obtained from the first convolution layer, and seven RR interval features added to the fully connected layer. In the second work (Wang et al. 2020b), the classification system is based on a two fully connected neural network classifiers. This system is based on 105 features, including RR interval, morphological features, statistical features and wavelet packet entropy. It should also be noted that the RR interval features are used in several compared methods (Zhang and Luo 2014; Shi et al. 2020; Wang et al. 2020a, b; Chen et al. 2017). The used RR interval features include previous, posterior, local average RR interval; mean of RR interval... etc. In Zhang and Luo (2014), the authors used 143 features. Among these features, 5 are based on RR-intervals. The remaining features comprise intra-beat features, wavelet coefficients, and morphological features. They used combined support vector machines as classifier and they have achieved an accuracy of 87.88%. Chen et al. (2017) have used 30 projection coefficients and 3 weighted RR intervals. They also used support vector machines as classifier and their model reached an overall accuracy of 93.1%. Wang et al. (2020a) used 320 morphological features extracted by one-dimensional (1D) CNN and 4 RR intervals features. These features are concatenated and used as the input of MLP. The model has obtained an overall accuracy of 92.53%. In this work, we used 30 coded features and only 2 RR interval features: previous RR and posterior RR.

To further assess the proposed model, we analyze its performance on classes V and S separately. This is in accordance with the AAMI recommendations, which suggest that the detection of V and S should be considered individually. Indeed, due to the imbalanced number of beats in each class, the overall accuracy is insufficient for evaluating the quality of the classification. The number of true positive N beats dominates the overall accuracy and the trained classifier may provide good overall accuracy without considering the detection of the most important arrhythmias (V and S beats).

From this Table 6, we note that our model outperforms these works in terms of accuracy, specificity, and positive prediction corresponding to class S. Concerning class V, the best results in terms of these tree metrics (i.e., accuracy, specificity, and positive prediction) are given by Zhang et al. (2014). The model proposed in Zhang et al. (2014) is based on the selection of effective feature subsets for distinguishing a class from others by making one-versus-one

Table 5 Comparison of the classification results on MIT-BIH arrhythmia dataset

Authors, year	Number of classes	Extraction features method	Classifier	Performance (%)
Zhang and Luo (2014)	4	F1	Combined SVM	Acc: 87.88
Garcia et al. (2017)	3	TVCG + complex network + PSO	SVM	Acc: 92.4
Luo et al. (2017)	4	Three layers SDA + multi-layer DNN	DNN	Acc: 89.3
Chen et al. (2017)	3	Projections + wRR	SVM	Acc: 93.1
Raj and Ray (2018)	5	Sparse Decomposition	PSO optimized LSTwin SVM	Acc: 89.93; Sen: 72.35
Li et al. (2019)	3	F2	CNN	Acc: 91.44
Shi et al. (2019)	4	F3	XGBoost	Acc: 92.1
Shi et al. (2020)	3	Raw data + RR	MIDNN	Acc: 94.2
Wang et al. (2020b)	3	CNN	CNN	Acc: 93.4
Song et al. (2020)	5	RC Network + ITML	Logistic Regression Classifier	Acc: 89.11; Sen: 74.13; Spe: 95.95
Wang et al. (2020a)	4	CNN + RR-intervals	MLP	Acc: 92.53
Yan et al. (2021)	3	Raw heartbeat data	CNN + SNN	Acc: 90
This study	3	SAE + RR-intervals	RBFNN + RVFLN	Acc: 93.11; Sen: 55.66; Spe: 80.55

F1: RR-intervals, wavelet coeff.,morphological features; RR: the time between the R peaks of two heartbeats; TVCG: temporal vectorcardiogram; MIDNN: multiple input layers deep neural network; SDA: stacked denoising autoencoder; F2: morphological, RR-intervals,beat-to-beat correlation feature; F3: RR-intervals, morphological, statistical, higher order statistic, wavelet transform, wavelet packet entropy; wRR: weighted RR; PSO: particle swarm optimization; LST: least square Twin; SVM: support vector machine; RC Network: residual-concatenate network; ITML: information-theoretic metric learning; SNN; spiking neural network; XGBoost: extreme gradient boosting

Table 6 Classification performance compared to the state-of-the-art methods on classes S and V

Authors, year	Overall accuracy (%)	Class S				Class V			
		Acc(%)	Spe(%)	Sen(%)	+P(%)	Acc(%)	Spe(%)	Sen(%)	+P(%)
Mar et al. (2011)	88.99	93.28	93.67	83.22	33.53	97.35	98.09	86.75	75.89
De Lannoy et al. (2011)	85	92.59	–	–	–	85.12	–	–	-–
Zhang et al. (2014)	88.33	93.33	93.95	79.06	35.98	**98.63**	**99.54**	85.48	**92.75**
Luo et al. (2017)	89.3	96.2	99.3	15.4	47.3	95.5	97.9	60.4	66.8
Sellami and Hwang (2019)	88.34	92.41	92.80	82.04	30.44	97.18	97.54	92.05	72.13
Guo et al. (2019)	–	93.61	96.40	62.70	61.21	93.71	94.77	91.25	88.30
Shi et al. (2020)	**94.2**	–	–	**90.74**	47.85	–	–	**92.92**	84.49
Wang et al. (2020b)	93.4	–	–	90.3	43.5	–	–	84.1	89.5
This study	93.11	**96.58**	**99.82**	13.27	**69.41**	96.38	99.20	56.13	82.57

Bold values indicate the best results

comparison. It should be noted that these authors mentioned that the RR interval features are important features which are often selected. In terms of sensitivity, the best results for both classes S and V are given by Shi et al. (2020). However, as it was mentioned in discussing Table 5, the model presented in this work has a large architecture which consists of multiple layers, i.e., convolutional, pooling, LSTM, concatenated, and fully connected layers.

Considering the above comparisons, we can conclude that the proposed model has given promising results with a simple structure, as it consists of a few neural layers and it considers only two RR interval features.

6 Conclusion

This paper introduced a neural system for heartbeat classification based on three types of neural networks and two types of features. The aim is to make use of the advantages and complementary properties of the combined neural networks. In order to solve the problem of imbalanced data, we used an oversampling method, i.e., SMOTE.

To evaluate the proposed system, we used the well-known MIT-BIH arrhythmia dataset and we performed an inter-patient test. Given the lack of methods for automatic design of autoencoders, we performed tests with several structures. We used a validation set, taken from the training data, to evaluate each structure. We found that only one layer and thirty features provided the best results. Subsequently, the proposed system was based on this structure. The proposed system was compared with some state-of-the-art methods and it outperformed most of them and provided promising performance. This confirms the efficiency of the proposed approach, which consists of extracting coded features using deep SAE, and then using these features besides two RR interval features as inputs to a hybrid RBFNN-RVLFN classifier.

As future work, we plan to further explore the idea of combining automatic features with handcrafted features. We want to develop classification models that are essentially based on deep learning architectures, but that permit the integration of some key handcrafted features.To tackle the problem of unbalanced data, we have used the SMOTE method but the sensitivity of the proposed method has been low, especially for class S. Therefore, we plan to use other sophisticated methods of unbalanced data and to explore more key features to identify class S. Therefore, we can have high performance with compact structures. We also plan to extend this approach to specific-patient applications and potential use with long-time monitoring of wireless body area networks. Indeed, this type of system requires simple, yet effective data-analysis models.

References

AAMI (1998) Testing and reporting performance results of cardiac rhythm and st segment measurement algorithms. ANSI/AAMI EC 38:1998

Angelov P, Gu X, Kangin D (2017) Empirical data analytics. Int J Intell Syst 32(12):1261–1284

Benoudjit N, Archambeau C, Lendasse A, Lee JA, Verleysen M et al (2002) Width optimization of the gaussian kernels in radial basis function networks. ESANN Esann 2:425–432

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Chen S, Hua W, Li Z, Li J, Gao X (2017) Heartbeat classification using projected and dynamic features of ecg signal. Biomed Sig Process Control 31:165–173

De Chazal P, O'Dwyer M, Reilly RB (2004) Automatic classification of heartbeats using ecg morphology and heartbeat interval features. IEEE Trans Biomed Eng 51(7):1196–1206

De Lannoy G, François D, Delbeke J, Verleysen M (2011) Weighted conditional random fields for supervised interpatient heartbeat classification. IEEE Trans Biomed Eng 59(1):241–247

Garcia G, Moreira G, Menotti D, Luz E (2017) Inter-patient ecg heartbeat classification with temporal vcg optimized by pso. Sci Rep 7(1):1–11

Guo L, Sim G, Matuszewski B (2019) Inter-patient ecg classification with convolutional and recurrent neural networks. Biocyber Biomed Eng 39(3):868–879

Ji Y, Zhang S, Xiao W (2019) Electrocardiogram classification based on faster regions with convolutional neural network. Sensors 19(11):2558

Katuwal R, Suganthan PN (2019) Stacked autoencoder based deep random vector functional link neural network for classification. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2019.105854

Li F, Xu Y, Chen Z, Liu Z (2019) Automated heartbeat classification using 3-d inputs based on convolutional neural network with multi-fields of view. IEEE Access 7:76295–76304

Luo K, Li J, Wang Z, Cuschieri A (2017) Patient-specific deep architectural model for ecg classification. J Healthc Eng. https://doi.org/10.1155/2017/4108720

Mar T, Zaunseder S, Martínez JP, Llamedo M, Poll R (2011) Optimization of ecg classification by means of feature selection. IEEE Trans Biomed Eng 58(8):2168–2177

Martis RJ, Acharya UR, Min LC (2013) Ecg beat classification using pca, lda, ica and discrete wavelet transform. Biomedical Signal Processing and Control 8(5):437–448

Moody GB, Mark RG (2001) The impact of the mit-bih arrhythmia database. IEEE Engineering in Medicine and Biology Magazine 20(3):45–50

Mousavi, S, Afghah, F (2019) Inter-and intra-patient ecg heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach. In ICASSP 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp) (pp. 1308–1312)

Murat, F, Yildirim, O, Talo, M, Baloglu, UB, Demir, Y, Acharya, U. R (2020) Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. Computers in biology and medicine, 103726

Nayak DR, Dash R, Majhi B, Pachori RB, Zhang Y (2020) A deep stacked random vector functional link network autoencoder for diagnosis of brain abnormalities and breast cancer. Biomedical Signal Processing and Control 58:101860

Ng A (2011) Sparse autoencoder. CS294A Lecture notes, 72(2011), 1–19

Nurmaini S, Darmawahyuni A, Sakti Mukti AN, Rachmatullah MN, Firdaus F, Tutuko B (2020) Deep learning-based stacked denoising and autoencoder for ecg heartbeat classification. Electronics 9(1):135

Pao Y- H, Takefuji Y (1992) Functional-link net computing: theory, system architecture, and functionalities. Computer 25(5):76–79

PhysioNet (2001) PhysioNet: MIT-BIH arrhythmia databasee. PhysioNet: MIT-BIH arrhythmia databasee. https://archive.physionet.org/cgi-bin/atm/ATM. ([Online; accessed 1-July-2020])

Raj S, Ray KC (2018) Sparse representation of ecg signals for automated recognition of cardiac arrhythmias. Expert systems with applications 105:49–64

Rao, CR, Mitra, SK, et al (1972) Generalized inverse of a matrix and its applications. In Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics

Roguia S, Mohamed N (2019) An optimized rbf-neural network for breast cancer classification. International Journal of Informatics and Applied Mathematics 1(1):24–34

Salah H, Nemissi M, Seridi H, Akdag H (2019) Subtractive clustering and particle swarm optimization based fuzzy classifier. International Journal of Fuzzy System Applications (IJFSA) 8(3):108–122

Sellami A, Hwang H (2019) A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. Expert Systems with Applications 122:75–84

Shi H, Wang H, Huang Y, Zhao L, Qin C, Liu C (2019) A hierarchical method based on weighted extreme gradient boosting in ecg heartbeat classification. Computer methods and programs in biomedicine 171:1–10

Shi H, Qin C, Xiao D, Zhao L, Liu C (2020) Automated heartbeat classification based on deep neural network with multiple input layers. Knowledge-Based Systems 188:105036

Siouda, R, Nemissi, M, Seridi, H (2020) A genetic algorithm-based deep rbf neural network for medical classification. In Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition Proceedings of the 1st international conference on intelligent systems and pattern recognition (pp. 27–32)

Siouda R, Nemissi M, Seridi H (2021) Ecg beat classification using neural classifier based on deep autoencoder and decomposition techniques. Progress Artif Intell 10:333

Soares E, Angelov P, Gu X (2020) Autonomous learning multiple-model zero-order classifier for heart sound classification. Appl Soft Comput 94

Song X, Yang G, Wang K, Huang Y, Yuan F, Yin Y (2020) Short term ecg classification with residual-concatenate network and metric learning. Multimed Tools Appl 79:22325–22336

Sun K, Zhang J, Zhang C, Hu J (2017) Generalized extreme learning machine autoencoder and a new deep neural network. Neurocomputing 230:374–381

Tang J, Deng C, Huang GB (2015) Extreme learning machine for multilayer perceptron. IEEE Trans Neural Netw Learn Syst 27(4):809–821

Wang T, Lu C, Yang M, Hong F, Liu C (2020a) A hybrid method for heartbeat classification via convolutional neural networks, multilayer perceptrons and focal loss. PeerJ Comput Sci 6:e324

Wang H, Shi H, Lin K, Qin C, Zhao L, Huang Y, Liu C (2020b) A high-precision arrhythmia classification method based on dual fully connected neural network. Biomed Sig Process Control 58:104

Yang J, Bai Y, Li G, Liu M, Liu X (2015) A novel method of diagnosing premature ventricular contraction based on sparse auto-encoder and softmax regression. Bio-med Mater Eng 26(s1):S1549–S1558

Yang J, Bai Y, Lin F, Liu M, Hou Z, Liu X (2018) A novel electrocardiogram arrhythmia classification method based on stacked sparse auto-encoders and softmax regression. Int J Mach Learn Cybernet 9(10):1733–1740

Yan Z, Zhou J, Wong, WF (2021) Energy efficient ecg classification with spiking neural network. Biomed Sig Process Control 63:102170

Yildirim O, Baloglu UB, Tan R-S, Ciaccio EJ, Acharya UR (2019) A new approach for arrhythmia classification using deep coded features and lstm networks. Comput Methods Prog Biomed 176:121–133

Zhang Z, Luo X (2014) Heartbeat classification using decision level fusion. Biomed Eng Lett 4(4):388–395

Zhang L, Suganthan PN (2016) A survey of randomized algorithms for training neural networks. Inform Sci 364:146–155

Zhang Z, Dong J, Luo X, Choi KS, Wu X (2014) Heartbeat classification using disease-specific feature selection. Comput Biol Med 46:79–89

Zhang Y, Wu J, Cai Z, Du B, Philip SY (2019) An unsupervised parameter learning model for rvfl neural network. Neural Netw 112:85–97

Zhou S, Tan B (2020) Electrocardiogram soft computing using hybrid deep learning cnn-elm. Appl Soft Comput 86:105778