Scientific African 12 (2021) e00778

Contents lists available at ScienceDirect

Scientific African

journal homepage: www.elsevier.com/locate/sciaf

A novel hybrid dimension reduction technique for efficient selection of bio-marker genes and prediction of heart failure status of patients

Kazeem Adesina Dauda*, Kabir Opeyemi Olorede, Samuel Adewale Aderoju

Department of Statistics and Mathematical Sciences, Kwara State University, Malate, Nigeria

ARTICLE INFO

Article history:
Received 15 December 2020
Revised 18 April 2021
Accepted 2 May 2021

Editor: B. Gyampoh

Keywords:
Deep neural network
Artificial neural networks
Back-propagation
Hidden layer
Genetic algorithm
Heart rhythm
Boruta algorithm

ABSTRACT

This study highlighted and provided a conceptual framework of a hybridized dimension reduction by combining Genetic Algorithms (GA) and Boruta Algorithm (BA) with Deep Neural Network (DNN). Among questions left unanswered sufficiently by both computational and biological scientists are: which genes among thousand of genes are statistically relevant to the prediction of patients' heart rhythm? and how they are associated with heart rhythm? A plethora of models has been proposed to reliably identify core informative genes. The premise of this present work is to address these limitations. Five distinct micro-array data on heart diseases have been taken into consideration to observe the prominent genes. We form three distinct set two-way hybrids between Boruta Algorithm and Neural Network (BANN); Genetic Algorithm and Deep Neural Network (GADNN) and Boruta Algorithm and Deep Neural Network (BADNN), respectively, to extract highly differentially expressed genes to achieve both better estimation and clearer interpretation of the parameters included in these models. The results of the filtering process were observed to be impressive since the technique removed noisy genes. The proposed BA algorithm was observed to select minimum genes in the wrapper process with about 80% of the five datasets than the proposed GA algorithm with 20%. Moreover, the empirical comparative results revealed that BADNN outperformed other proposed algorithms with prediction accuracy of 97%, 87%, and 100% respectively. Finally, this study has successfully demonstrated the utility, versatility, and applicability of hybrid dimension reduction algorithms (HDRA) in the realm of deep neural networks.

1. Introduction

Heart rhythm is a transmission of electrical signals that normally control heart rate and rhythm. The Abnormal rhythm of the heartbeat is due to irregularities in the transmission of the electrical signals known as arrhythmias. The irregularity of the heartbeat is divided into too fast or too low. A heartbeat is too fast if above 100 beats per minute (known as tachycardia)

* Corresponding author.
E-mail addresses: kazeem.dauda@kwasu.edu.ng (K.A. Dauda), kabir.olorede@kwasu.edu.ng (K.O. Olorede), samuel.aderoju@kwasu.edu.ng (S.A. Aderoju).

https://doi.org/10.1016/j.sciaf.2021.e00778

and too low if below 60 beats per minute (known as bradycardia)[1]. In this current century, some of the researchers such as Allison et al. [1], Beckmann et al. [2] have identified genetics (genes) as a primary cause of electrical cardiac disorders. Genetic testing in heart disease is the process of taking a sample of a person's DNA to look for changes (i.e. pathogenic mutations) that could cause inherited heart disease [1]. The major application of DNA microarrays can be classified into three distinct classes. Firstly, the researcher can find different expression levels between predefined groups of samples (e.g, identification of genes differentially expressed in the heart disease from normal and abnormal heart rhythm). A second application is a prediction of class, it requires identifying the class membership of the sample (e.g to predict whether or not the patient has normal or abnormal heart rhythm). Thirdly, the identification of subgroups that share common features [3]. Therefore, techniques for extracting the informative genes that are associated with heart disease are necessary, and most importantly the use of hybridized computing algorithms such as Genetic and Boruta Algorithms [4,5] (GA and BA) with deep neural (DNN) and artificial neural networks (ANN)[6] (BANN, GADBN, and BADBN) to undertake the complex task by reducing the dimensionality of the feature space. The most common architecture of artificial neural networks (ANN) considered in this study is feed-forward ANN. In order to select the most influential genes in the gene expression profile and to account for the gene's variability or consistent distribution pattern, the boruta [7] and genetic algorithms were applied [8]. These features selection techniques were different from other dimension reduction techniques, such as principal component analysis (PCA)[9] in that they identify genes that are highly associated with high (normal) and low (abnormal) heart rhythm. Unlike the PCA and others, they reduce the number of features in the data set, without reducing the dimensionality [10].

In the recent era, deep neural network (DNN) has gained more contests in pattern recognition and machine learning [11]. The deep belief networks (DBN) are the most common DNN, which is also regarded as the state of the art Artificial Neural Network (ANN) [6,12] are now commonly used in microarray data analysis.

In machine and statistical learning techniques, data processing has become necessary since this identifies important features and removes the irrelevant, redundant, or noisy features to reduce the dimensionality of feature space [13].

The major goal of this study is to devise an effective method of extracting features with the most important information to cardiac disease, using the GADNN, BANN, and BADNN due to their ability to explain complex relationships in the microarray data and vast learning ability. The proposed hybrid techniques in this work can be formed by three major components: Filter, Wrapper, and Embedded (FWE) Algorithms. This is parallel to some criteria that produce improve methods in solving high dimensional problems [14]. Therefore, this study inherits the merits of FWE. A plethora of works has introduced the filter and wrapper (FW) hybrid while only scanty literature has proposed a three-way hybrid FWE. In 2019, Shukla et al. [15] devised a hybridized technique using Conditional Mutual Information Maximization (CMIM) as a filter and Binary Genetic Algorithm (BGA) as a wrapper (CMIMBGA), they further extended their study to the k-nearest neighbor (k-NN) and support vector machine (SVM) as embedded. The two classifiers (i.e k-NN and SVM) were compared based on prediction accuracy, but their study did not implement deep neural techniques that are known to be very successful in the realm of machine learning [6]. Tong and Mintram [16] explored the effectiveness of a genetic algorithm-neural network (GANN) hybrid in analyzing gene expression tumour data, the GA was used as wrapper and ANN as embedded. Recently, Ding et al. [17] proposed the use of two-way hybridized dimension reduction that adds genetic algorithm and competitive swarm optimization. Unfortunately, their algorithm still suffers from computational time. Consequently, Yara et al. [6] deeply explained the novel approach and architecture of a Deep Neural Network algorithm [18] and compared the performance of DNN with Cortical Algorithm (CA)[19]. Some other measures of dimension reduction have been implemented in the literature and all in the realm of machine learning procedure [20]. More precisely, the novelty of this study comes from two major perspectives. Firstly, the research emphasizes extracting informative features from a high-dimensional and highly complex data set by improving on the classification results. Secondly, the use of random forest (RF) [21] to compute the fitness of GA and BA, 10 fold cross-validation [22] and parallel programming which is not very common in the context of dimension reduction was employed. Lastly, the use of DNN and ANN as embedded after GA and BA wrapper steps. Five publicly available benchmark data from Gene Expression Omnibus(GEO) repository [23] have been taken into consideration to assess and compare the adequacies of the proposed hybridized methods on the other existing methods. In the subsequent section, we first describe and explain the full concept of BA and GA followed by ANN and DNN. Consequently, we implement the proposed algorithms GADNN, BANN, and BADNN and compare their performances based on the five real-life high-dimensional microarray data set.

2. Settings and methodologies

In this section, we present the ideas and algorithm behind the proposed Hybrid Dimension Reduction Algorithms (HDRA) techniques.

2.0.1. Hybrid Dimension Reduction Algorithms (HDRA)

Two new methods of hybrid feature selection called BADNN and GADNN were developed in this study. The process consists of a filter, wrapper, and embedded methods, the simple algorithm of these processes was given in the flowchart in Fig. 1. The filter methods are the initial selection approach that is classifier independent and thus easily adopted by any classifiers. Moreover, it is easily scaled to a very high dimensional microarray data set and computationally easy and fast. This study adopted the use of T-test statistics as a filter by measuring how the genes were associated with the outcome variable (High and Low heart rhythm). Subsequently, the wrapper selection examines genes dependencies, which are the

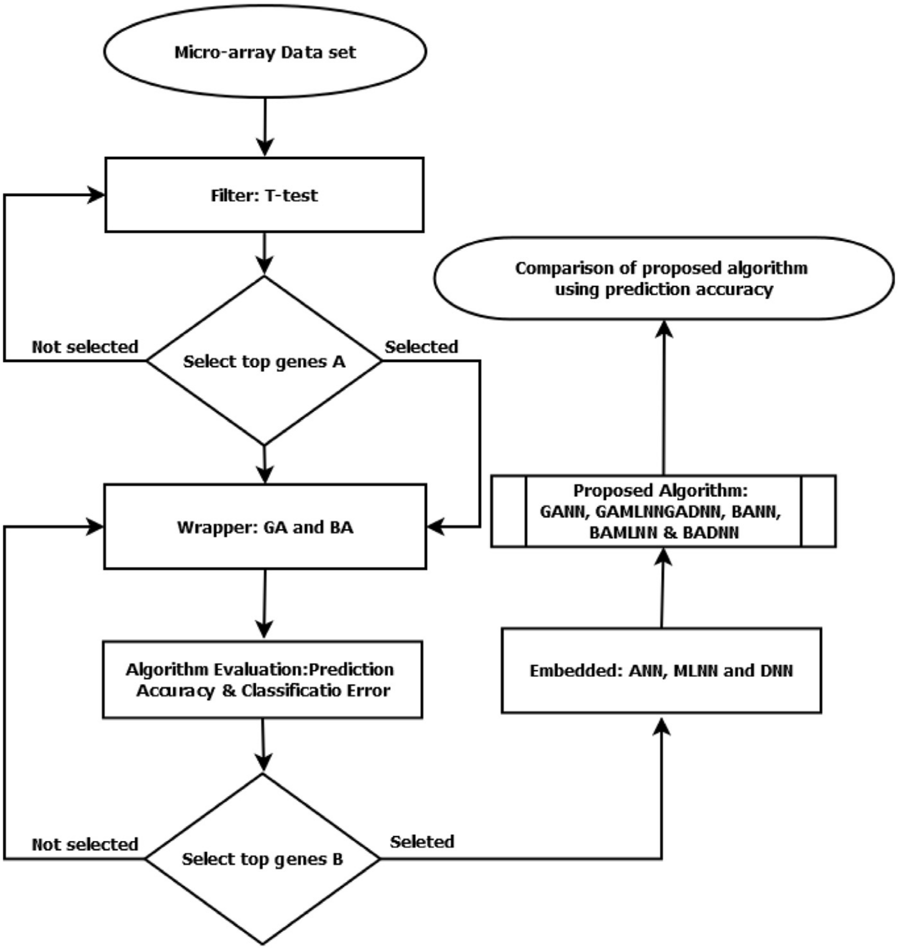
Micro-array Data set
Filter: T-test
Not selected
Select top genes A
Selected
Comparison of proposed algorithm using prediction accuracy
Wrapper: GA and BA
Proposed Algorithm: GANN, GAMLNNGADNN, BANN, BAMLNN & BADNN
Algorithm Evaluation:Prediction Accuracy & Classificatio Error
Embedded: ANN, MLNN and DNN
Not selected
Select top genes B
Seleted

Fig. 1. Proposed flowchart.

genes with low expression level but strong interaction can be easily detectable via the wrapper. The GA and BA were adopted as wrapper using the random forest as fitnesses and the proposed hybrid method was highly generic in nature. The detailed steps of this flowchart (Fig. 1) are described below:

1. Read in the high dimensional data set
2. Filter part for the first (A) dimension reduction using T-test method in Eq. 1, 2
3. Select the most influential genes based on step 2
4. Wrapper part for second (B) dimension reduction using GA and BA
5. Output and save the optimal subset genes from step 4
6. Evaluate the two Wrappers using prediction accuracy and misclassification error rate
7. Embedded part: we implement Artificial Neural Network (ANN) and Deep Neural Network (DNN)
8. Identify the best algorithm among the proposed GADNN, BANN, and BADNN using model prediction accuracy.

2.1. Filter: T-statistics (T-Filter)

The *t*-test is the classical method of selecting the most influential genes in the microarray data set and it has proven to be effective in the literature [13,24]. The major objective of using the *t*-test is to evaluate whether the means of the two classes are statistically different from each other. Each sample is labeled with $[1 and -1]$ respectively. For each gene F_j, the mean μ_j^1 (or μ_j^{-1}) and the standard deviation σ_j^1 (or σ_j^{-1}) are computed via the sample labelled 1 (or −1). Them, the score T^* value of the *t*-test can be obtained through Eq. 1

$$T^* = \frac{\mid \mu_j^1 - \mu_j^{-1} \mid}{\sqrt{\frac{(\sigma_j^1)^2}{n_1} + \frac{(\sigma_j^{-1})^2}{n_{-1}}}}, \tag{1}$$

where n_1 (or n_{-1}) is the number of samples labelled as 1 or -1.

Thus, the maximum probability of rejecting the null hypothesis (P-value) from Eq. 1 can be calculated using Eq. 2

$$P - value = 2 * \Phi(| T^* |, df), \tag{2}$$

where Φ is the cumulative probability of a normal distribution and df is the degree of freedom. In selecting the most important genes, we compare the $p - value$ of each pair with a level of significance α. Those pairs of the gene with $p - value \leq \alpha = 0.01$ are considered important genes.

2.2. Wrapper: Genetic Algorithm (GA)

The Genetic Algorithm (GA) was initially developed by John Holland in 1975 [25], which was motivated by the common selection process and heuristically works on parallel search. The GA is mainly used to solve the problem of optimization base on the process related to the generic scheme. There are two major basics of GA operations, these include the probability of crossover (pc) and mutation probability (pm), and the two control factors were Pc crossover and Pm mutation probabilities. The mechanism of the likelihood that enhances the finding of a globally optimal solution makes the GA more flexible and robust. The process of the GA is given in Algorithm 1 and a summary of the parameters used in this research is depicted in Table 1.

Algorithm 1: GA Algorithm

```
Result: Initialize the parameters $nPop = m, t_{max}, t = 0$
initialization;
while $t \leq t_{max}$ do
    Create pop $m$, $t_{max}$ ;
    for $k = 1$ to $m$ do
        Parents $[m_1, m_2]$= selection scheme $(m, nPop)$;
        Child=$X_{OR}[m_1, m_2]$;
        $Mu = mutation[Child]$;
    end
    Replace $m$ with $Child_1, Child_2, \cdots, Child_m$;
    $t = t + 1$;
end
Save the highest fitness value
```

2.2.1. GA Algorithm structure

Performing GA involves moving from the initial population, fitness evaluation, selection of best individual, crossover, mutation, and checking for the optimal individual after satisfying the pre-specify termination condition (see Algorithm 1 and Fig. 2). During this process, the most influential genes are identified and the chromosome which has N number of genes are represented with values 1 (selected) and 0 (unselected), respectively. The relevancy of features (genes) is addressed using the wrapper method based on classification performance. For the full details and the overall structure of GA see Algorithm 1 and Fig. 2

From the Algorithm 1, m is the number of population, r is a random number ranging from 0 to 1, the chrome mapped the selected and unselected gene with the help of cut-point σ value set to be 0.5. In the chromosome encoding, we used a binary bit string to denote an individual. Each individual denotes a candidate gene subset, i.e. each binary digit represents a gene. Therefore, the bit with value 1 means the corresponding gene is being selected and the bit with 0 means the gene is not selected, and the length of each is determined by the number of features N.

The underlying idea behind the flowchart in Fig. 1 is that GA is used to generate some random possible solutions called population that represent various variables (genes) and they combine the best solutions iteratively. The basic GA operations

Table 1
GA parameter initialization.

Parameters	Value
Population size	20
Mutation probability	0.1
Number of generation	100
Crossover probability	0.8
Unselected feature	0
Selected feature	1
Chromosome length	70

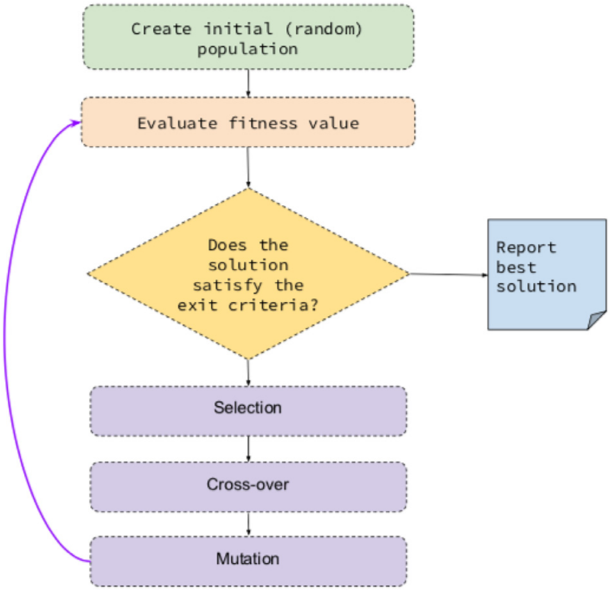
Create initial (random) population
Evaluate fitness value
Does the solution satisfy the exit criteria?
Report best solution
Selection
Cross-over
Mutation

Fig. 2. GA procedure flowchart.

are in line with this combination and these operations include selection, mutation, and cross-over as stated earlier. The process of picking the best-fitted individual in the generation is by the selection, creating two new individuals based on the two gene solutions is by cross-over and finally, mutation changes a gene randomly in the individual.

2.2.2. Proposed GA working structure (T-Filter+GA)

In the development of our hybridize method, we combined filter and wrapper techniques as a single algorithm in this very first part, while the second part will be given in the next section. The full details of the proposed methods are given in Algorithm 2 and the GA parameters used were described in Table 1 all as a proposed framework.

Algorithm 2: Proposed GA Working Structure.

```
Collect the most influential genes through the filter methods as expressed in equation 1, 2 in section~2.1;
Recall the selected genes in 1. ;
Implement wrapper using GA structure in algorithm~1 ;
while stopping condition is not met do
    Select parent from the population;
    Generate offspring by cross-over between parents;
    Generate offspring by mutation between parents;
    Evaluate the fitness of the updated solutions;
    Replace the worst individual of the population with the new population;
end
Return the best and the optimal genes subset;
Implement embedded method
```

2.3. Wrapper: Boruta Algorithm (BA)

Boruta Algorithm (BA) is an extension of knowledge introduced by Stoppiglia et al. [26] to determine important genes by comparing the importance of all real genes to the random probes. Miron and Witold [27] implemented the algorithm in the information system in an R package called Boruta. The BA was purposely developed to identify all relevant genes in the rem of classification learning algorithms. The main concept behind the BA is to compare the important genes with those of randomly called shadow genes using statistical tests and Random Forests. For further investigation and better clarification on this algorithm see Miron and Witold [27].

2.3.1. Proposed BA working structure (T-Filter+BA)

This section presents the implementation of the Hybrid Dimension Reduction Algorithms (HDRA) in the realm of BA. In the same vein, we combine filter, wrapper, and embedded techniques as a single algorithm in this very second part of our

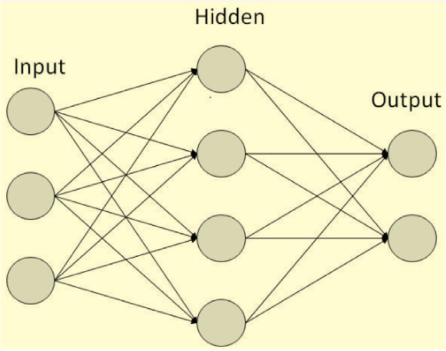

Fig. 3. Example of 3-layered ANN.

HDRA. Unlike the first part, where the GA was implemented in the wrapper, here BA is deemed fit to be used as wrapper and the prospective process is given in Algorithm 3

```
Algorithm 3: Proposed BA Working Structure.
1 Collect the most influential genes through the filter methods as expressed in equation 1, 2 in section~2.1;
2 Recall the selected genes in 1. ;
3 Implement wrapper using BA structure ;
4 Return the best and the optimal genes subset;
5 Implement embedded method
```

2.4. Embedded: Artificial neural network (ANN)

A particular machine learning method that is designed to mimic the neuron structure of the human brain by inherent data structures through an adaptive algorithm is called Neural Network [6]. The ANN mirrors the human brain through many simple processing elements. The elements are nodes that are acting like neurons and work cooperatively. Moreover, the nodes are interconnected using connection *links*, which are related with different connection *weights* ω and arranged in layers. Every connection has a weight attached which may have either a positive or negative value associated with it. Positive weights activate the neuron while negative weights inhibit it (see the flowchart of the ANN training process in Fig. 4). The ANN structure also consists of input *layers*, one or more hidden layers, and output layers. A simple model of an ANN is shown in Fig. 3. The output value $\hat{Y}$ (High and Low hearts rhythm) of input (x) signals is generated with some weight parameters ω and sum linearly with activation function f_e. The initial weight parameter ω_0 is known as offset or bias and connects to a unit x_0 that is permanently set to 1. The mathematical model of a neuron is given in Eq. 3

$$\hat{Y} = f_e(x, w) = \sum_{i=0}^{I} \omega_{ij} x_{ij} + b_j, \tag{3}$$

where i is an input variable with I dimension and j hidden nodes. The ANN training process in Fig. 4 parameters was optimized for constructing an ANN model, including hidden layer neurons, optimization algorithms, learning rate, and the number of iterations. After the optimization, the ANN model was trained by the training set to carry out effective prediction. Given the input and output data, the connection weights and thresholds between neurons were adjusted as variables to lessen prediction errors. The predicted output of the ANN model was linked with the trial data, and the biases were modified by calculating the error. When the error was less than the threshold $(E(n) < \epsilon)$ or the number of iterations reached the upper limit $(iterations > 100)$, the training process automatically stopped.

2.4.1. Training neural networks

The process of training the neural network required prediction error which is measured using a cost function, such as least square in Eq. 4.

$$\mathcal{L}_{net} = \sum_{i=0}^{I} (y_k - \hat{y}_k)^2 \tag{4}$$

To generalize better and speed up learning process in ANN, the cross-entropy cost function in Eq. 5 is used together with the sigmoid activation function by initialize the weight in Eq. 3 to 0. Consequently, the set of training data $(\mathcal{D}_{train})$ of input-

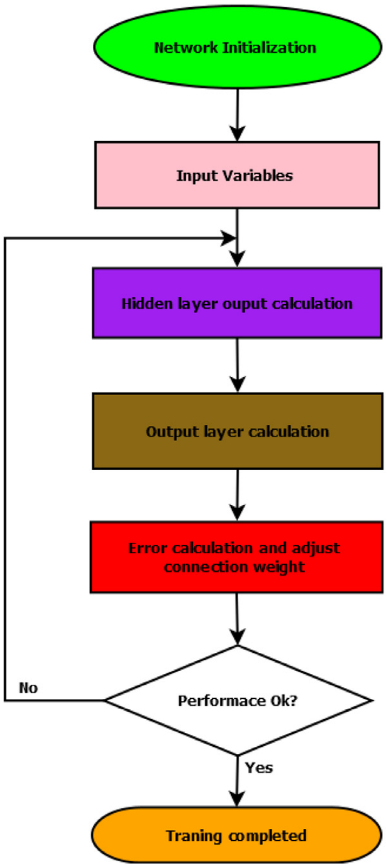

Fig. 4. ANN structure and a flow chart of the training process.

output pairings $\{(x_k, y_k)\}$ weight can be updated for each individual k using Eq. 6.

$$\mathcal{L}_{net} = -\sum_{k=1}^{(|\mathcal{D}_{train}|)} \log P(\frac{y_k}{\hat{y}_k}),$$

$$\mathcal{L}_{net} = -\sum_{k=1}^{(|\mathcal{D}_{train}|,df)} y_k \log \hat{y}_k + (1-\hat{y}_k)\log(1-\hat{y}_k). \tag{5}$$

$$\omega_i = \omega_i + \mathcal{E}(y_k - \hat{y}_k), \tag{6}$$

where $\mathcal{E}$ is the learning rate. Since this study ought to use sigmoid function, then the gradient descent update in Eq. 6 is overseen by cross-entropy. The literature has established that the cost functions of neural networks are highly non-convex [28]. Therefore, finding the global minimum of the cost function is difficult and optimization often converges in a local minimum. In minimizing the error function, the partial derivative of the network parameters is computed and the simple way to approximate the gradient of the cost function is to utilize the finite differences. Then, the partial derivatives used in an optimization algorithm to minimize the cost function in this study is stochastic gradient descent defined in Algorithm 4 .

Algorithm 4: Stochastic gradient descent: training size m, learning rate $\mathcal{E}$.

```
Randomly shuffle data set;
repeat
    for $i = 1$ to $m$ do
        $\omega_j \leftarrow \omega_j - \mathcal{E}\frac{\partial\mathcal{L}_{net}(\omega,x_i,y_i)}{\partial\omega_j}$(simultaneously for all $j$)
    end
until convergence;
```

Table 2
ANN parameter initialization .

Parameters	Setting
Learning algorithm	Feed forward (ANN)
Activation function	Sigmoid
Initial random weights	0.7
Architecture	Connection of pattern between nodes
Learning rate	0.01
Loss function	Cross-entropy
Layers	16

2.4.2. Proposed BANN Working Algorithm

The proposed techniques are typical combination of filter (Section 2.1), wrapper BA (Section 2.2.2) and thereafter the ANN (Section 2.4). The most important features must first be selected through FW before passing it on to the final algorithm that performs classification and then computes the prediction accuracy. The ANN network parameters used in the study were presented in Table 2.

In this section, we design an embedded model and selection technique called *Genetic Algorithm-Neural Network* (GANN) and *Borutal Algorithm-Neural Network (BANN)*. Algorithm 5 reveals the architecture design of GANN and BANN models. These two models consist of three main modules as presented in Algorithm 5, these are filter, wrapper, and embedded models and selections techniques.

```
Algorithm 5: Proposed BANN Working Algorithm.
1 Collect the most influential genes through the filter methods as expressed in equation 1, 2 in section~2.1;
2 Implement wrapper using GA and BA structures in Algorithm~2 and 3 ;
3 Implement embedded models: GANN and BANN via section~2.4
```

2.5. Embedded: Deep Neural Networks (DNN)

Deep learning (DL) is a technique that is currently trending and receiving much attention in various areas of study. It is a set of learning algorithms that are purposely developed to learn complex representations through a multilayer neural network with many hidden layers. DL is now gaining more attention in computational biology and has been used for analyzing DNA data [29,30]. There is four commonly available learning algorithm that can be used to build DL model for supervised learning problems, these include: convolutional neural networks (CNNs), DNNs, multilayer perceptrons (MLPs) and recurrent neural networks (RNNs)[31,32]. Since any of these algorithms can be used to build the DL model, then, we proposed the use of DBN for building the model. This study focuses on the deep neural network (DNN), which is the particular form of DL methodology, and two phases of the artificial neural network (ANN) model. Here, we present how the two phases of the artificial neural network (ANN) model were trained. The first phase focuses on the pre-training via restricted Boltzmann machine (RBM) and it is majorly used to initialize the network model. The second phase is the fine-turning phase that processes in a supervised manner. In the next section, we explain and discuss the selected DL algorithm model DNN in detail.

2.5.1. Proposed GADNN and BADNN working Algorithm

The proposed techniques are typical combination of filter (Section 2.1), wrapper BA/GA (Section 2.3.1; 2.2.2) and thereafter the DNN (Section 2.5). The best features must first be selected through FW before passing it on to the final algorithm that performs classification and then computes the prediction accuracy. The embedded model and selection techniques called Genetic Algorithm-Deep Neural Network (GADNN) and Borutal Algorithm-Deep Neural Network(BADNN) were designed to optimize and select the best features as presented in Algorithm 6 . The Algorithm 6 consists of three main modules, which are filter, wrapper, and embedded models. The first module performs the preprocessing step and the selection of features based on the algorithm in Section 2.1. The selected features in module 1 will then pass on to the next module called wrapper where further dimension reduction will take place. The final module utilizes the selected feature in module 2 to build the final model through the ANN and DNN.

```
Algorithm 6: Proposed GADNN and BADNN Working Algorithm.
1 Collect the most influential genes through the filter methods as expressed in equation~1, 2 in section~2.1 ;
2 Implement wrapper using GA and BA structures in Algorithm~2 and 3 ;
3 Implement embedded models: GADNN and BADNN via section~2.5
```

Table 3
Number of genes selected by the filtering process .

Dataset	Sample Size	Number of Genes	Filtering method	
			Adopted Filter2.1	CMIM[15]
GSE34788	120	32,960	800	8114
GSE86569	22	16,383	275	4438
GSE115574	59	16,383	481	8192
GSE112266	53	108	62	53
GSE68475	21	16,383	227	2751

2.6. Performance measures

This section presents the various used performance indices based on the classification and variable important measure (Relative Importance).

2.6.1. Prediction accuracy and classification error

We use the prediction accuracy and classification error to evaluate the proposed algorithms in our study. Prediction accuracy was calculated for each GADNN, BANN, and BADNN respectively. The performance measures are defined as:

Let TP = True positive, FP = False Positive, TN = True Negative and FN = False Negative, then

$$PredictionAccuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$ClassificationError = \frac{FP + FN}{TP + TN + FP + FN}$$

.

2.6.2. Relative importance by Garson Algorithm

The implementation and experimentation of the proposed algorithms further extended to observe variables (genes) that are highly important to the heart rhythm. This is achieved by using the techniques called Relative Importance developed by Garson Algorithm [33]. The purpose of this method as established in literature [34,35] is to observe whether there is positive, negative, and or no association between the gene expression and the response variable such as heart rhythm classes. The Garson technique partitions the hidden layer weights into components associated with each input node. Thereafter, the percentage of all hidden node weight associated with the input node was used to measure the relative importance of that attribute.

3. Numerical results

This section presents the explanatory examples that highlights the similarities and differences in the proposed and adopted Algorithm (2, 3,5 and 6) by strictly working with flowchart in Fig. 1 and by following the description in Section 2.0.1. Five different datasets on heart rhythm were downloaded from NCBI Gene Expression Omnibus(GEO) repository [23]. The descriptions of these datasets were fully highlighted in the literatures [36–39]. Subsequently, we present the various results generated from our proposed hybridize techniques graphically and numerically.

3.1. Filter: Dimension reduction results

The first part of the proposed Hybrid Dimension Reduction Algorithms (HDRA) (Section 2.0.1) is to implement the filtering process discussed in Section 2.1. The various results of the first implementation were presented in the subsequent table and the major objective of this section is to perform dimension reduction by showcasing the number and the important genes selected. The results of the five microarray gene expression data were presented in Table 3 below.

The adopted filter selected various genes from the five datasets as presented in Table 3. Of the 32,960 original genes in the GSE34788 data, only 800 were selected as core relevant biomarker genes by the filter method. From the 16,383 genes in the GSE86569 data, only 275 were selected as important genes for predicting heart rhythm. Approximately three percent (481) of the 16,383 genes in the GSE115574 data were selected as relevant genes by the filter method. From the GSE112266 data, only 53 (49%) of the 108 original genes were selected by the CMIM filter method. Finally, only 227 (1%) of the original 16,383 genes in the GSE6847 data were selected as important genes by the adopted filter method. In all, the number of genes selected by the adopted filter is far lower than the existing method, and thus better in feature selection and dimension technique. The most used method of visualizing gene expression data is to display its heatmap, which can also be combined with the cluster dendrograms. The purpose of the heatmap is to identify genes that are highly or lowly expressed and biological signatures that are associated with a particular condition (i.e disease condition). The heatmap of the GSE34788 dataset is presented in Fig. 5. In the heatmap (Fig. 5) of the 30 genes, the data is displayed in a grid

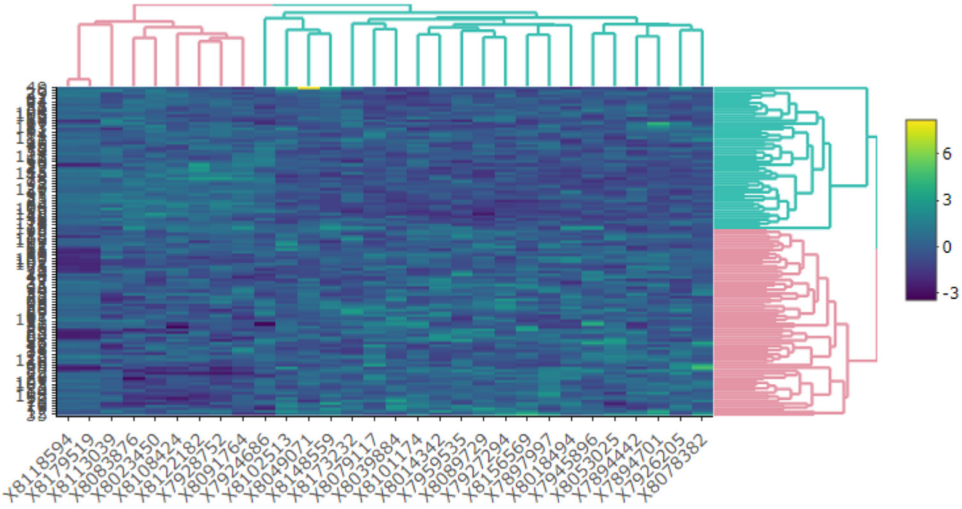

Fig. 5. Heatmap of the GSE34788 micro-array data.

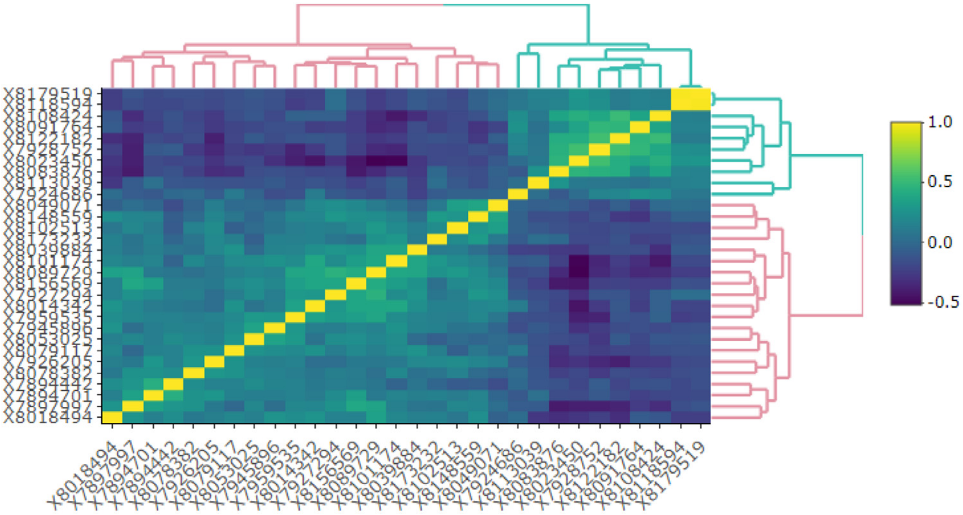

Fig. 6. Correlation matrix heatmap for GSE34788 micro-array data.

where each row represents a sample and each column represents a gene. The color and intensity of the boxes are used to represent changes in gene expression, the shades of yellow and green indicate elevated expression (i.e. highly significant to the sample) while shades of purple indicate decreased expression (i.e. zero significance to the sample), while blue indicates unchanged expression (i.e. genes with scale measures zero). Similarly, the heatmap of GSE86569, GSE115574, GSE112266, and GSE68475 micro-array dataset after the filtering were presented in Figure SM1. Additionally, identifying how genes interact in the micro-array dataset is very important when deciding which gene to be included in the algorithm. One of the fastest ways to strengthen a model is to identify and reduce the genes in the dataset that are highly correlated. Correlated genes add much noise and inaccuracy to the model and thus, make the model difficult to achieve the desired outcome. Hence, this study presents the correlation matrix of the 30 genes from the GSE34788 dataset shown in Fig. 6. If two independent genes have a strong relationship, then they are considered either positively or negatively correlated, however, correlation ranges from -1 to $+1$. Values closer to zero mean there is no linear trend between the two genes. Values closer to pm1 imply strong positive and negative associations. The diagonals are all $1's$ (yellow) because those squares are correlating each variable to itself (i.e. perfect correlation). The correlation matrix heatmap of other datasets was presented in Figure SM2.

3.2. Wrapper: Genetic Algorithm (GA) and Boruta Algorithm (BA) results

The second part of this study is to implement the algorithms in sections 2.2.2 and 2.3.1. For the five micro-array datasets, further dimension reduction was performed on the genes selected by the filtering process on each dataset. This

Table 4
Comparative performance of the proposed T-Filter+B and T-Filter+GA with CMIM+BGA based on the number of the genes selected and executed time .

Dataset	T-Filter+BA2.3.1		T-Filter+GA2.2.2		CMIM+BGA[15]	
	# of Genes	Time (sec.)	# of Genes	Time (sec.)	# of Genes	Time (sec.)
GSE34788	**16**	**68.52**	314	7729.53	2902	117766.38
GSE86569	**12**	**7.59**	26	3900.64	1824	43739.84
GSE115574	**19**	**21.60**	373	7090.17	2695	62501.96
GSE112265	17	**6.53**	19	4798.96	3	**3**876.91
GSE68475	23	**7.87**	**17**	32910.46	1183	8011.11

The **bold values** represent good results (performances).

Table 5
Comparative performance (Proportion of accuracy and error rate) of T-Filter+B, T-Filter+GA and CMIM+BGA Algorithms .

Dataset	T-Filter+BA2.3.1		T-Filter+GA2.2.2		CMIM+BGA[15]	
	Accuracy	Error rate	Accuracy	Error rate	Accuracy	Error rate
GSE34788	**0.9000**	**0.1000**	0.8670	0.1330	0.8087	0.1913
GSE86569	0.9091	0.0909	**0.9500**	**0.0500**	0.7833	0.2167
GSE115574	**0.8983**	**0.1017**	0.8833	0.1167	0.8167	0.1833
GSE112266	**0.7736**	**0.2264**	0.7600	0.2400	0.8166	0.1834
GSE68475	**0.9524**	**0.0476**	0.9167	0.0833	0.9500	0.05

The **bold values** represent good results (performances)

second process is termed *Wrapper* as detailed in the flowchart in Fig. 1. The GA and BA algorithms were trained with only previously selected genes in Table 3 through Filter-selected genes as predictor variables (T-Filter+BA and T-Filter+GA). The details of the best-selected subset among the genes using the two algorithms and the execution time (in seconds) for the five micro-array datasets were presented in Table 4. Moreover, the algorithms were further evaluated by the means of classification error rate and prediction accuracy as shown in Table 5.

Table 4 revealed the best genes selected by the algorithms. The T-Filter+BA algorithm selects a minimum number of genes with minimum execution time across three datasets: With GSE34788, 16 out of 800 were selected with an execution time of 68.52 secs., with GSE86569, 12 out of 275 genes were selected within just 759 secs. and with GSE115571, only 19 out of 481 genes were selected within 21.60 secs. respectively. The CMIM+BGA algorithm performs better by selecting only 3 of the 53 genes in the GSE112265 data within 3876.91 seconds. Obviously, the T-Filter+BA algorithm consistently outperformed other algorithms in terms of model selections and computational complexity for all the data sets except with the GSE112265 and GSE68475 where the T-Filter+GA selected fewer genes than T-Filter+BA. However, The BA cannot be said to have performed poorly on the GSE112265 and GSE68475 data since with additional genes it selected, its classification error is much smaller, even at the lower computational time than the T-Filter+GA and CMIM+BGA algorithms (see Table 4).

Results in Table 5 showcase the classification error rate and prediction accuracy of the proposed (T-Filter+BA and T-Filter+GA) and the existing algorithms through the five datasets. The results of the prediction accuracy and classification error rate revealed that the T-Filter+BA algorithm gives the highest prediction accuracy and minimum error rate on four datasets: GSE34788, GSE115574, GSE112266, and GSE68475 (90.0%, 89.8%, 77.4% and 95.2%), respectively. Both the T-Filter+BA and the T-Filter+GA algorithms compete favorably on the GSE86569 data to 1 place of decimal, while low prediction accuracy was observed with CMIM+BGA algorithm. The results of the T-Filter+BA algorithm were further complemented with the Area Under Curve (AUC) plot on the GSE34788 dataset, the AUC is 0.9833 as shown in Figure SM3. Other AUC plots were presented in Figure SM4. The most highly influential genes selected by T-Filter+BA algorithms across the five datasets were visualized using box plots. Figure SM5 presents the importance of the selected genes in dataset GSE34788 hierarchically and gives a crystal clear call on the significance of variables in the dataset. In general, it can be concluded that based on the median normalization permutation importance (tick middle lines in boxplots), gene X7917148 is the most relevant. This is followed by gene X8016628 and the least important gene is X8027381. The boxplot of the other four datasets was presented in Figure SM6.

3.3. *Embedded process: Artificial Neural Network (ANN) and Deep Neural Networks (DNN) results*

The last part of this study is to implement the Artificial Neural Network (ANN) and Deep Neural Networks (DNN) methods discussed in sections 2.4 and 2.5. After the feature reduction method filter and the redundant features selection method wrapper (FW), then the next step is to implement the ANN and DNN. The various comparative results are presented in Table 6.

The three proposed combinatorial algorithms and the two existing methods (GANN and CMIMBGA) algorithms were evaluated on five gene expression datasets. The empirical results in Table 6 revealed that BADNN outperformed other proposed

Table 6
Proportion of prediction accuracies of the proposed T-Filter+BA+ANN (BANN), T-Filter+BA+DNN (BADNN) and T-Filter+GA+DNN (GADNN) and Algorithms and two existing GA+ANN (GANN), CMIM+BGA+SVM (CMIMBGA-SVM) and CMIM+BGA+KNN (CMIMBGA-KNN) .

	Proposed algorithms			Existing algorithms		
Dataset	BANN	BADNN	GADNN	GANN[16]	CMIMBGA-SVM[15]	CMIMBGA-KNN[15]
GSE34788	0.6417	0.9083	**0.9853**	0.8294	0.8000	0.8167
GSE86569	0.7272	0.9545	**1.0000**	0.9500	0.5455	0.7273
GSE115574	0.7966	**0.9661**	0.9492	0.9200	0.5254	0.7627
GSE112266	0.7358	**0.8679**	0.8302	0.7333	0.5094	0.9623
GSE68475	0.9048	**1.0000**	0.9524	0.9000	0.5238	0.8095

The **bold values** represent good results (performances)

algorithms base on the best maximum accuracies from three datasets: GSE115574 0.9661, GSE112266 0.8679 and GSE68475 1.0000 respectively. However, the proposed GADNN algorithm performs better in two datasets with the classification accuracy of 0.9833 in the GSE34788 dataset and 1.0000 in the GSE86569 dataset. In general, as depicted in Table 6 the two proposed algorithms achieve better performance over other methods previously studied in the literature for the microarray dataset. These results reflect better utility, versatility, and efficacy of the BADNN and GADNN (i.e Hybrid Dimension Reduction Algorithms (HDRA)) in a high-dimensional micro-array dataset in term of feature selection, identification of gene signature, and prediction accuracy. To identify the best and highly influential genes in the five datasets, further analysis was done and presented.

3.4. Further analysis

Further analysis was explored on the proposed HDRA to identify genes the better influence the heart rhythm and to identify patients that are at risk when experiencing normal and abnormal heart rhythm. The Deep Learning architecture of the GSE34788 dataset is present in Figure SM7 and the other four datasets were present in Figure SM8 and SM9. The results of the most prominent genes among thousands of genes in the dataset GSE34788 were presented in Figure SM10 and Table SM1. The results of other datasets were presented in figures and tables (Figures SM11 and SM12, Tables SM1 to SM5). The results of the training deep Learning architecture presented in Figure SM7 revealed the training process of the proposed algorithms and the plot consists of trained synaptic weight, the hidden layers, and the output. The connection of the input genes and hidden layers was achieved after about 1000 iteration as preliminary declared during the training process. The variable important results were achieved using the Garson algorithm techniques highlighted in the previous section and the results of the important genes in the GSE34788 dataset were tabulated in Table SM1 and visualized in Figure SM10. The results revealed that the X8016628 gene has a perfect negative relationship with the heart rate (High and Low responders). Moreover, we also observed from this figure and table that genes X7952739, X8091764, and X8140762 have an intermediate negative association with heart rate, while genes X9078382, X7917148, X8027391, and X8159243 have an intermediate positive association with heart rate. Meanwhile, X8108424, X7956152, X7969023, X8023195, X7901982, X8031646, and X7897172 genes are weakly associated (i.e. negatively and positively) with the heart rate, since they have a relatively important value close to zero and most likely they have some marginal effect on the heart rate. Specifically, the only gene that is not associated with the heart rate is X7969264, this gene has been observed not to influence High and Low responders in the data GSE34788, although this gene has initially been captured by filter and wrapper, our embedded techniques identify this gene as none influential one in the face of variable importance. Similarly, other dataset results (Figures SM11 and SM12, Tables SM1 to SM5) can also be interpreted in this direction and all influential genes were also presented graphically.

4. Conclusion

This work has presented an intelligent dimension reduction approach through T-Filter+BA+ANN (BANN), T-Filter+GA+DNN(GADNN), and T-Filter+BA+DNN$BADNN) hybridizations to extract and identify the highly differentially expressed genes in heart rhythm diagnosis and mammography. The study has also demonstrated the mechanism for learning using a combination of the Filter, Wrapper, and Embedded techniques. We compared the hybrid dimension reduction algorithms (HDRA) that were proposed to identify all relevant genes and optimal prediction performance in the realm of deep learning and artificial neural network. Three of the five data used identified BADNN as the most powerful algorithm, followed by GADBN based on prediction accuracy. This study further observed that using GAs and BGA as a wrapper process seems to be very time-consuming and provides less prediction accuracy. Additionally, we also noticed that the GANN and CMIM+BGA select higher genes than the BAs and therefore failed to account for the principle of parsimony. The BADNN algorithm was further used to detail the effect of all the selected genes on the heart rhythm (heart rate). The results revealed that the X8016628 gene has a perfect negative relationship with the heart rate (High and Low responders). We also observed that genes X7952739, X8091764, and X8140762 have an intermediate negative association with heart rate, while genes X9078382, X7917148, X8027391, and X8159243 have an intermediate positive association with heart rate. Some of the genes that are identified to be weakly associated (i.e. negatively and positively) with heart rate are X8108424, X7956152, X7969023, X8023195, X7901982, X8031646, and X7897172, since they have a relatively important value close to zero and most likely they have some marginal effect on the heart rate. Specifically, the only gene that does not associate with the heart rate is X7969264. This gene has been observed not to influence High and Low responders in the data GSE34788.

Finally, we recommend BAs as wrappers in performing dimension reduction and BADNN as the most powerful algorithm in the identification of highly expressed genes in a three-way hybrid. We note that the proposed BADNN algorithm also has the capability of working perfectly with low dimensional data.

Declaration of Competing Interest

Authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

CRediT authorship contribution statement

Kazeem Adesina Dauda: Conceptualization, Investigation, Writing - original draft, Data curation, Methodology. **Kabir Opeyemi Olorede:** Methodology, Writing - review & editing. **Samuel Adewale Aderoju:** Resources, Writing - original draft.

Acknowledgment

This research study has been financially supported by TETFUND INSTITUTIONAL BASED RESEARCH (TETFund) (Reference No.:KWASUIBR/CSP/250918/VOL5/TETF/0058) through Kwara State University, Malete.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.sciaf.2021.e00778

References

[1] A.L. Cirino, C.Y. Ho, Genetic testing for inherited heart disease. circulation, NIH Public Accesss 128 (1) (2013) 4–8.
[2] B.M. Beckmann, A. Pfeufer, S. Kääb, Inherited cardiac arrhythmias, Deutsches Aerzteblatt Online (2011).
[3] A.L. Tarca, R. Romero, S. Draghici, Analysis of microarray experiments of gene expression profilings, Am. J. Obstet. Gynecol. 195 (2) (2007) 373–388.
[4] M. Amiri, H.R. Pourghasemi, G.A. Ghanbarian, S.F. Afzali, Assessment of the importance of gully erosion effective factors using boruta algorithm and its spatial modeling and mapping using three machine learning algorithms, Sustainability 340 (2019) 55–69, doi:10.1016/j.geoderma.2018.12.042.
[5] L. Scrucca, On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution, R J 9 (1) (2017) 187–206. https://journal.r-project.org/archive/2017/RJ-2017-008
[6] Y. Rizk, N. Hajj, N. Mitri, M. Awad, Deep belief networks and cortical algorithms: acomparative study for supervised classification, Applied Computing and Informatics (2018), doi:10.1016/j.aci.2018.01.004.
[7] M.B. Kursa, W.R. Rudnicki, Feature selection with the boruta package, J Stat Softw 36 (11) (2010) 1–13. http://www.jstatsoft.org/v36/i11/
[8] P. Zhang, B. Verma, K. Kumar, Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection, Pattern Recognit Lett 26 (7) (2005) 909–919.
[9] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (2014) 16–28.
[10] A. Shah, J. Iqbal, Spatial distribution and mobility assessment of carcinogenic heavy metals in soil profiles using geostatistics and random forest, boruta algorithm, Sustainability 10 (2018) 799, doi:10.3390/su10030799.
[11] D.C. Ciresan, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks 32 (2012) 333–338.
[12] S. Nersisyan, M. Shkurnikov, A. Poloznikov, A. Turchinovich, B. Burwinkel, N. Anisimov, A. Tonevitsky, A post-processing algorithm for miRNA microarray data, Int J Mol Sci 21 (4) (2020) 12–28.
[13] F. Tan, Improving Feature Selection Techniques for Machine Learning, Georgia State Universit, 2007 Ph.D. thesis.
[14] Iñaki, P. Larranaga, R. Blanco, A.J. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains, Artif Intell Med 31 (2) (2004) 91–103.
[15] A.K. Shukla, P. Singh, M. Vardhan, A new hybrid feature subset selection framework based on binary genetic algorithm and information theory, Int J Comput Intell Appl 18 (3) (2019) 22, doi:10.1142/s1469026819500202.
[16] D.L. Tong, R. Mintram, Genetic algorithm-neural network (GANN): a study of neural network activation functions and depth of genetic algorithm search applied to feature selection, Int. J. Mach. Learn. Cybern. 1 (1) (2010) 75–87, doi:10.1007/s13042-010-0004-x.
[17] Y. Ding, K. Zhou, W. Bi, Feature selection based on hybridization of genetic algorithm and competitive swarm optimizer, Soft Comput. 24 (15) (2020) 11663–11672, doi:10.1007/s00500-019-04628-6.
[18] G. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, Neural Comput 18 (7) (2006) 1527–1554.
[19] N. Hajj, Y. Rizk, M. Awad, A mapreduce cortical algorithms implementation for unsupervised learning of big data, Procedia Comput Sci 53 (2015) 327–334.
[20] R. Armananzas, M. Iglesias, D.A. Morales, L. Alonso-Nanclares, Voxel-based diagnosis of Alzheimer's disease using classifier ensembles, IEEE J. Biomed. Health Inf. 21 (3) (2017) 778–784.
[21] H. Ishwaran, U.B. Kogalur, Random forests for survival, regression, and classification (RF-SRC), R package version 2.5.1. (2017).
[22] H. Marter, M. Martern, Multivariate analysis of quality: an introduction, John Wiley & Sons Ltd., Chichester, 2001.
[23] T. Barrett, S.E. Wilhite, P. Ledoux, C. Evangelista, I.F. Kim, M. Tomashevsky, K.A. Marshall, K.H. Phillippy, P.M. Sherman, M. Holko, A. Yefanov, H. Lee, N. Zhang, C.L. Robertson, N. Serova, S. Davis, A. Soboleva, NCBI GEO: archive for functional genomics data sets–update, Nucleic acids researchl 41 (Database issue) (2013) D991–D995, doi:10.1093/nar/gks1193.

[24] H. Liu, J. Li, L. Wong, A comparative study on feature selection and classification methods using gene expression profiles and proteomic pattern, Genomic Informatics 13 (2002).

[25] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning,machine learning, Nature 8 (2) (1989) 95–99.

[26] H. Stoppiglia, G. Dreyfus, R. Dubois, Y. Oussar, Ranking a random feature for variable and feature selection, Journal of Machine Learning Research 3 (2003) 1399–1414.

[27] B.K. Miron, R.R. Witold, Feature selection with boruta package, J Stat Softw 36 (11) (2010).

[28] P.O. GLAUNER, Comparison of Training Methods for Deep Neural Networks, Imperial College London Department of Computing (Machine Learning), 2015 Master's thesis.

[29] Y. Bengio, Learning deep architectures for AI, Found Trends Mach Learn 2 (2009) 1–127.

[30] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.

[31] J. Smolander, M. Dehmer, F. Emmert-Streib, Comparing deep belief networks with support vector machines for classifying gene expression data from complex disorders, FEBS Open Bio 9 (2019), doi:10.1002/2211-5463.12652.

[32] J. Smolander, M. Dehmer, F. Emmert-Streib, Deep belief network based hybrid model for building energy consumption prediction, Energies 11 (1) (2018) 242, doi:10.3390/en11010242.

[33] G. Garson, Interpreting neural-network connection weights, AI Expert 6 (1991) 46–51.

[34] S.O. Odeyemi, M.A. Akinpelu, R. Abdulwahab, K.A. Dauda, S. Chris-Ukaegbu, Scour depth prediction for Asa Dam Bridge, Ilorin, using artificial neural network, Int. J. Eng. Res. Afr. 47 (2020) 46–51. https://www.scientific.net/jera.47.53

[35] A.D. Kazeem, N.B. Akinbowale, O.O. Kabir, O.A. Sulaiman, R.O. Oluwaseun, Effectiveness of contraceptive usage among reproductive ages in Nigeria using artificial neural network (ANN), Computing and Information Systems Journal 22 (1) (2018) 24–34.

[36] E. Rampersaud, L. Nathanson, J. Farmer, K. Meshbane, R.L. Belton, A. Dressen, M. Cuccaro, A. Musto, S. Daunert, S. Deo, N. Hudson, J.M. Vance, D. Seo, A. Mendez, D.M. Dykxhoorn, M.A. Pericak-Vance, P.J. Goldschmidt-Clermont, Genomic signatures of a global fitness index in a multi-ethnic cohort of women, Ann. Hum. Genet. 77 (2) (2013) 147–157. PMID: 23289938

[37] L. Yun-Shien, Different expression of inflammation-related proteins in human heart failure, GEO,VI (2017). https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE86569

[38] W. Li, L. Wang, Y. Wu, Z. Yuan, J. Zhou, Weighted gene co-expression network analysis to identify key modules and hub genes associated with atrial fibrillation, Int J Mol Med. 45 (2) (2020) 401–416, doi:10.3892/ijmm.2019.4416.

[39] R. Annunziata, A. Ritter, A.E. Fortunato, A. Manzotti, S. Cheminant-Navarro, N. Agier, M.J.J. Huysman, P. Winge, A.M. Bones, F. Bouget, M. Cosentino-Lagomarsino, J. Bouly, A. Falciatore, BHLH-PAS protein RITMO1 regulates diel biological rhythms in the marine diatom phaeodactylum tricornutum, Proceedings of the National Academy of Sciences 116 (26) (2019) 13137–13142, doi:10.1073/pnas.1819660116. https://www.pnas.org/content/116/26/13137