Soft Computing (2019) 23:13139–13159
https://doi.org/10.1007/s00500-019-03856-0

METHODOLOGIES AND APPLICATION

An evolutionary deep belief network extreme learning-based for breast cancer diagnosis

Somayeh Ronoud[1] · Shahrokh Asadi[1]

Published online: 22 February 2019

Abstract

Cancer is one of the leading causes of morbidity and mortality worldwide with increasing prevalence. Breast cancer is the most common type among women, and its early diagnosis is crucially important. Cancer diagnosis is a classification problem, where its nature requires very high classification accuracy. As artificial neural networks (ANNs) have a high capability in modeling nonlinear relationships in data, they are frequently used as good global approximators in prediction and classification problems. However, in complex problems such as diagnosing breast cancer, shallow ANNs may cause certain problems due to their limited capacity of modeling and representation. Therefore, deep architectures are essential for extracting the complicated structure of cancer data. Under such circumstances, deep belief networks (DBNs) are appropriate choice whose application involves two major challenges: (1) the method of fine-tuning the network weights and biases and (2) the number of hidden layers and neurons. The present study suggests two novel evolutionary methods, namely $E_{(T)}$-DBN-BP-ELM and $E_{(T)}$-DBN-ELM-BP, that address the first challenge via combining DBN with extreme learning machine (ELM) classifier. In the proposed methods, because of the very large solution space of DBN topologies, the genetic algorithm (GA), which is able to search globally in the solutions space wondrously, has been applied for architecture optimization to tackle the second challenge. The third proposed method in this paper, $E_{(TW)}$-DBN, uses GA to solve both challenges, in which DBN topology and weights evolve simultaneously. The proposed models are tested using two breast cancer datasets and compared with the state-of-the-art methods in the literature in terms of classification performance metrics and area under ROC (AUC) curves. According to the results, the proposed methods exhibit very high diagnostic performance in classification of breast cancer.

Keywords Medical decision support system · Deep belief network · Extreme learning machine · Breast cancer diagnosis

1 Introduction and literature review

Cancer is the second leading cause of death globally and is responsible for an estimated 9.6 million deaths in 2018, according to World Health Organization[1] (WHO) report. Based on this report, men are mostly confronted with lung, prostate, colon, stomach and liver cancer, while in women, breast, colon, lung, cervical, and stomach cancer are the most prevalent. Among all different types of cancers, breast cancer is the second cause of cancer death after lung cancer which is the most prevalent among women.

One of the most basic obstacles in treating breast cancer is lack of appropriate method for early diagnosis. Today, due to advancements in medical field and the complexity of decisions related to diagnosis and treatment, the attention of experts is drawn to the use of smart equipment and Medical Decision Support Systems (MDSS), especially in the critical field of breast cancer diagnosis. The use of various types of smart systems in medicine is increasing. Also, the use of these equipment and systems can reduce potential mistakes caused by the fatigue or lack of experience of clinical professionals in detecting this type of cancer and deciding whether to be benign or malignant

Communicated by V. Loia.

✉ Shahrokh Asadi
shahrokh.asadi@ut.ac.ir

Somayeh Ronoud
s_ronoud@ut.ac.ir

[1] Data Mining Laboratory, Department of Engineering, College of Farabi, University of Tehran, Tehran, Iran

[1] http://www.who.int/mediacentre/factsheets/fs297/en/.

(Milovic 2012). Cancer diagnosis is a classification problem including complex and nonlinear relationships among its data sets. A wide range of data mining methods have been proposed so far in the literature to predict the diagnosis of breast cancer. Quinlan (1996) used C4.5 decision tree method to classify this cancer and achieved a classification accuracy of 94.74%. Nauck and Kruse (1999) used neuro-fuzzy method and obtained an accuracy of 95.06%. Pena-Reyes and Sipper (1999) applied fuzzy genetic method and achieved 97.36% classification accuracy. Albrecht et al. (2002) used a combination of perceptron with simulated annealing (SA) method and reported an accuracy of 98.8%. Abonyi and Szeifert (2003) implemented supervised fuzzy clustering technique to achieve an accuracy of 95.57%. Polat and Günes (2007) used Artificial Immune Recognition System (AIRS) and fuzzy resource allocation mechanism to achieve 98.51% accuracy and they also applied least square-support vector machine (LS-SVM) to obtain 98.53% accuracy.

Übeyli (2007) applied five different classifiers, i.e., SVM, probabilistic neural network, recurrent neural network, combined neural network and multilayer perceptron neural network, which respectively, achieved accuracies of 99.54%, 98.61%, 98.15%, 97.4%, and 91.92%. Örkcü and Bal (2011) compared performance of back-propagation neural network (BPNN), binary-coded genetic algorithm, and also real-coded genetic algorithm on breast cancer dataset and acquired accuracies of 93.1%, 94%, and 96.5%, respectively. Marcano-Cedeño et al. (2011) proposed a new artificial metaplasticity multilayer perceptron algorithm that outperformed BPNN on similar breast cancer dataset with 99.26% accuracy compared to BPNN with 94.51% accuracy rate for 60–40 training–testing samples. Similarly, Lavanya and Rani (2011) used decision tree algorithms and achieved 94.84% accuracy on Breast Cancer Wisconsin—Original (WBCO) dataset and 92.97% on Breast Cancer Wisconsin—Diagnostic (WDBC) dataset. Malmir et al. (2013) reported 97.75% and 97.63% accuracies by training a multilayer perceptron network, through applying imperialist competitive algorithm (ICA) and particle swarm optimization (PSO), respectively. Koyuncu and Ceylan (2013) achieved a higher classification accuracy of 98.05% by using nine classifiers in a Rotation Forest-Artificial Neural Network (RF-ANN). Xue et al. (2014) applied PSO algorithm for feature selection using novel initialization and updating mechanisms, which resulted in 94.74% accuracy rate. Sumbaly et al. (2014) also used decision tree to achieve 94.56% accuracy. Zheng et al. (2014) obtained 97.38% accuracy through applying a combination of K-means and SVM algorithms on WDBC dataset. Bhardwaj and Tiwari (2015) obtained 99.26% accuracy on WBCO dataset using tenfold cross-validation method in genetically optimized neural network (GONN) model.

In Karabatak's research (2015), a new classifier, namely weighted naïve Bayesian, was proposed which achieved a classification accuracy of 98.54% on WBCO dataset. Abdel-Zaher and Eldeib (2016) also achieved an accuracy of 99.68% through using of DBN composition along with Levenberg–Marquardt (LM) algorithm on the WBCO dataset considering 54.9–45.1% of data for training–testing partitions. Kong et al. (2015) also presented the jointly sparse discriminant analysis (JSDA) for feature selection and reached an accuracy of 93.85% for the WDBC dataset. Table 1 summarizes these methods.

Among the various approaches in the field of medical diagnosis, ANN is one of the data mining techniques, which has attracted the attention of many researchers for medical decisions support. The ANNs are flexible mathematical structures that can identify complex relationships between input and output data. The learning ability of ANNs has made them a powerful tool for various applications, such as classification, clustering, control systems, prediction, and many other applications (Ahmadizar et al. 2015). The feed-forward back propagation (BP) is the most commonly used architecture among various types of neural networks. In the literature, the BP's three-layer networks (including one hidden layer) are used as a global approximator for cancer diagnosis (Çinar et al. 2009; Saritas et al. 2010).

The most of machine learning techniques applied are shallow architectures, so far. Çınar et al. (2009) presented a system for early detection of prostate cancer using ANN and SVM with a structure containing one hidden layer. Saritas et al. (2010) also applied single-hidden-layer ANN for the detection of prostate cancer. Many other researchers also used architectures for neural networks, which typically had at most one or two layers of nonlinear feature transformations (Bhardwaj and Tiwari 2015; Chen et al. 2011; Flores-Fernández et al. 2012; Park et al. 2013; Wu et al. 2011; Zheng et al. 2014). It has been observed that shallow architectures have been effective in solving many simple or finite problems, but their limited capacity in modeling and representation can trigger problems in dealing with more complex real-world applications (Deng and Yu 2014). In fact, cancer diagnosis problems with complex, high-dimensions and noisy data cannot be solved simply by using conventional shallow methods that contain only a few nonlinear operations, and these methods do not have the capacity to accurately model such data (Längkvist et al. 2014). The complicated information processing mechanisms require deep architectures to extract complex structures and make input representations. Therefore, researchers became interested more in using deeper

Table 1 A list of data mining methods in breast cancer diagnosis literature

References	Data mining methods	Accuracy (%)
Quinlan (1996)	C4.5 decision tree	94.74[a]
Nauck and Kruse (1999)	Neuro-fuzzy	95.06[a]
Pena-Reyes and Sipper (1999)	Fuzzy-GA	97.36[a]
Albrecht et al. (2002)	Hybrid of perceptron and simulated annealing	98.8[a]
Abonyi and Szeifert (2003)	Supervised fuzzy clustering	95.57[a]
Polat and Günes (2007)	Artificial immune recognition system and fuzzy resource allocation mechanism	98.51[a]
	Least square-support vector machine	98.53[a]
Übeyli (2007)	Support vector machine	99.54[a]
	Probabilistic neural network	98.61[a]
	Recurrent neural network	98.15[a]
	Combined neural network	97.4[a]
	Multilayer perceptron neural network	91.92[a]
Örkcü and Bal (2011)	Back-propagation neural network	93.1[a]
	Binary-coded genetic algorithm	94[a]
	Real-coded genetic algorithm	96.5[a]
Marcano-Cedeño et al. (2011)	Artificial metaplasticity multilayer perceptron	99.26[a]
Lavanya and Rani (2011)	Decision tree	94.84[a]
		92.97[b]
Malmir et al. (2013)	MLP training with imperialist competitive algorithm	97.75[a]
	MLP training with particle swarm optimization	97.63[a]
Koyuncu and Ceylan (2013)	Rotation forest-artificial neural network	98.05[a]
Xue et al. (2014)	Feature selection with PSO	94.74[b]
Sumbaly et al. (2014)	Decision tree	94.56[a]
Zheng et al. (2014)	Hybrid of K-means and SVM	97.38[b]
Bhardwaj and Tiwari (2015)	Genetically optimized neural network	99.26[a]
Karabatak (2015)	Weighted naïve Bayesian	98.54[a]
Abdel-Zaher and Eldeib (2016)	Hybrid of DBN with Levenberg–Marquardt	99.68[a]
Kong et al. (2015)	Jointly sparse discriminant analysis	93.85[b]

[a]Accuracy on WBCO, [b]accuracy on WDBC

structures for the network (Asadi et al. 2012, 2013; Kazemi et al. 2016; Razavi et al. 2015; Shahrabi et al. 2013).

Learning techniques in deep neural networks refer to a class of machine learning techniques that model high-level abstractions in input data with hierarchical architectures and multiple layers. In these structures, hierarchical extraction of the features is possible, so that high-level features are formed in a combination of low-level features at several levels (Bengio 2009). Different types of deep neural networks are used for classification in the literature (Abdel-Zaher and Eldeib 2016; Cao et al. 2016; Ciompi et al. 2015; Wang et al. 2016; Yu et al. 2015). Deep belief network (DBN) (Hinton et al. 2006) is one of the deep networks that have achieved remarkable performance in prediction and classification problems using restricted Boltzmann machines (RBMs) for layer-wise unsupervised learning (network's weights pre-training), and supervised back-propagation algorithm for fine-tuning.

There are two major challenges in applying DBN: (1) Taking which method for training the network; (2) How many hidden layers and neurons should be included in the network? Since in usual DBN, after network unsupervised pre-training stage, the weights between the last hidden layer and output layer is randomly assigned, it sounds that selecting a method which opts these weights in a more intelligent way improves the classification's performance. ELM (Huang et al. 2006) is a competitive learning method having a considerable performance in terms of accuracy and computational time. One of the main objectives of this paper is to employ ELM in DBN in order to select the weights between the last hidden layer and output layer intelligently than randomly. Moreover, it has already been shown in the literature that combination of ELM with SVM yields desirable consequences. Liu et al. (2008) and Frénay and Verleysen (2011) provided a considerable contribution through presenting a method in which ELM was employed within SVM which resulted in better generalization performance. Huang et al. (2010) showed that optimization constrains of SVM can be lowered, if the core of ELM is used and the optimal solution could be consequently more effectively found. Therefore, one can expect that combination of ELM with DBN yield acceptable outcomes. In the present study, two new structures are presented by applying ELM classifier to improve DBN training.

On the other hand, due to very large solution space of deep network weights, it seems necessary to apply a method with global search characteristic. Genetic algorithm (GA) is incredibly capable of global search in solution space. Therefore, in the third proposed method in this research, a combination of GA and DBN has been used for fine-tuning and acquiring suitable DBN weights.

Architecture of a neural network is crucially important, as it affects learning capacity and generalization performance of network (Ahmadizar et al. 2015). Although deep learning methods has acquired acceptable results in different applications, it is difficult to determine which structure with how many layers or how many neurons in each layer is suitable for specific task, and also, a special knowledge is required for choosing reasonable values for necessary parameters (Guo et al. 2015). Therefore, a method for finding optimized architecture of DBN is required that be suitable for cancer diagnosis.

To the best of the authors' knowledge, there is no method in the literature to determine the number of hidden layers and neurons in DBN. Researchers also used predetermined structures for applying DBN in their applications (Abdel-Zaher and Eldeib 2016; Hrasko et al. 2015). Manual search is another strategy which has been widely used to do so (Hinton 2010; Larochelle et al. 2007; Shen et al. 2015). In this method, the DBN is frequently tested by different structures, and finally, a structure with the best performance is selected.

Discovering optimal architecture for a deep network is a search problem in which determining the optimized topology for neural networks is the goal. Evolutionary algorithms are appropriate choices for solving the neural networks architecture problem. GA is an evolutionary search method that is capable of finding optimal or near optimal solutions (Asadi 2019; Mansourypoor and Asadi 2017; Mehmanpazir and Asadi 2017; Tahan and Asadi 2018a, b). The most attractive GA characteristic is its flexibility in handling various types of objective functions. The main reasons for this success are as follows. GAs are capable of solving difficult problems quickly and confidently. They are also quite easy to interface with existing simulations and models. Moreover, they are extensible and easy to hybridize. All of these reasons can be summarized into one reason: GAs are robust. In spite of this fact that they do not guarantee to find the global optimum solution of a problem, they are generally suitable for finding acceptably good solutions to problems in a reasonable amount of time (Asadi and Shahrabi 2017). So, in this article's proposed models, i.e., $E_{(T)}$-DBN-BP-ELM, $E_{(T)}$-DBN-ELM-BP, and $E_{(TW)}$-DBN, GA has been used for the first time in the DBN architecture optimization.

In summary, the present research suggests three new evolutionary methods called $E_{(T)}$-DBN-BP-ELM, $E_{(T)}$-DBN-ELM-BP, and $E_{(TW)}$-DBN, to find the optimal or near optimal network architecture using GA. Additionally, to improve the DBN training, the new and different learning algorithms in each proposed method are presented.

The remainder of this paper is organized as follows. The used materials, proposed models, and its novelties are detailed in Sect. 2. In Sect. 3, evaluation of experimental results is presented. A discussion is provided in Sect. 4. Finally, the conclusion and future works are given in Sect. 5.

2 Methodology

In this paper, the DBN classic model is improved by applying the efficient classifier ELM, and two new combinations, $E_{(T)}$-DBN-ELM-BP and $E_{(T)}$-DBN-BP-ELM, are then presented. The third proposed model, $E_{(TW)}$-DBN, effectively utilizes the advantages of GA for DBN fine-tuning. In all three methods, the appropriate network architecture is evolved by employing GA. In order to understand more about the proposed models, first, Sect. 2.1 introduces DBN and ELM, and then, the proposed models are presented in Sect. 2.2, comprehensively.

2.1 Material

The infrastructure of DBN includes several layers of the RBMs. The layers of RBMs are placed on top of each other to build a DBN, forming a network that can extract high-level abstractions from the raw data. The DBN is detailed in Sect. 2.1.1. ELM is another method that has been used in the two proposed models of this paper and described in Sect. 2.1.2. ELM is an efficient learning algorithm for SLFN, which has a higher scalability and less computational complexity than the error back-propagation (BP) algorithm.

2.1.1 Deep belief network

A DBN is created with several layers of the RBM. RBM is an artificial neural network that has a single visible layer and a single hidden layer, in which unsupervised learning is performed. The visible layer represents the data, while another layer of hidden units represents features that capture higher-order correlations in the data. In a DBN, the hidden layer of each RBM is considered as the visible layer of the next RBM, with the last RBM hidden layer, as an exception. RBMs use the hidden layer for the probability distribution of visible variables (Hinton et al. 2006). A DBN for a problem with m input, C output and N RBM is shown in Fig. 1. Also, b and c in the first RBM are the biases of visible and hidden layer, respectively, that are not shown in other RBMs for the sake of simplicity.

The RBM (Smolensky 1986) is a generative stochastic neural network that can learn the probability distribution of input data sets. RBM is a type of Boltzmann machine with this limitation that visible and hidden units form a bipartite graph (there is no connection between nodes in a layer).

A RBM consists of a set of visible units, $v \in \{0,1\}^n$, and a set of hidden units, $h \in \{0,1\}^m$, where n and m represent the number of visible and hidden units, respectively. In a RBM, the joint configuration $(\mathbf{v}, \mathbf{h})$ considering bias has the following energy, as Eq. (1):

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i\in visible} b_i v_i - \sum_{j\in hidden} c_j h_j - \sum_i \sum_j W_{ij} v_i h_j, \tag{1}$$

where v_i and h_j are the binary state of the visible unit i and the hidden unit j. Also, b_i and c_j are, respectively, biases of visible and hidden layers, and W_{ij} is the connection weight between them. Lower energy indicates that the network is in a more desirable state. This network, for each possible state of visible and hidden vectors pairs, assigns a probability value using energy function as Eq. (2):

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})), \tag{2}$$

where Z is a partition function that can be obtained from total of $\exp(-E(\mathbf{v},\mathbf{h}))$ on all possible configurations, and it is used for normalization:

$$Z = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})). \tag{3}$$

The probability that the network assigns to the visible vector $\mathbf{v}$ is given by Eq. (4):

$$P(\mathbf{v}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h})) \tag{4}$$

The derivative of the log probability of a training vector with respect to a weight can be computed as Eq. (5):

$$\frac{\partial}{\partial W_{ij}} \log P(v) = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}, \tag{5}$$

where $\langle \cdots \rangle_p$ indicates the expected value according to the distribution P. This means that the training rule for updating the weights in the log probability of training data is as Eq. (6):

$$\Delta W_{ij} = \varepsilon\left(\langle \mathbf{v_i h_j} \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}\right) \tag{6}$$

where ε is the learning rate. Similarly, the weight-updating rule in the bias parameters is given through Eqs. (7) and (8):

$$\Delta b_i = \varepsilon\left(\langle \mathbf{v_i} \rangle_{\text{data}} - \langle v_i \rangle_{\text{model}}\right) \tag{7}$$

$$\Delta c_j = \varepsilon\left(\langle \mathbf{h_j} \rangle_{\text{data}} - \langle h_j \rangle_{\text{model}}\right) \tag{8}$$

Since the RBM is a bipartite graph, it is easy to calculate $\langle \mathbf{v_i h_j} \rangle_{\text{data}}$, called "positive phase." The hidden unit activations are mutually independent with respect to the visible unit activations (and vice versa):

$$P(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{m} P(v_i|\mathbf{h}) \tag{9}$$

The individual activation probabilities, i.e., the state of a visible node with respect to the hidden vector is shown by Eq. (10):

$$P(v_i = 1|\mathbf{h}) = \text{sigm}\left(b_i + \sum_j W_{ij} h_j\right) \tag{10}$$

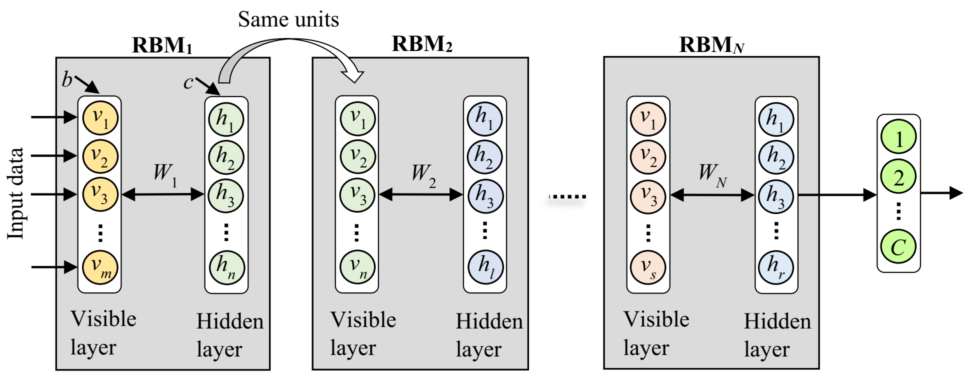


Fig. 1 The network structure of deep belief network

where "sigm" is the logistic sigmoid function defined as $(\mathrm{sigm}(x) = 1/(1+\exp(-x)))$. Similarly, for the randomly selected training input $\mathbf{v}$, the binary state h_j of each hidden unit j is set to 1 with the probability according to Eq. (11):

$$\boldsymbol{P}(\boldsymbol{h_j} = 1|\mathbf{v}) = \mathrm{sigm}\left(\boldsymbol{c_j} + \sum_i \boldsymbol{W_{ij}v_i}\right). \tag{11}$$

By calculating this probability, this hidden unit is turned on (value is changed to 1), if a random generated number from uniform distribution over range (0, 1) be less than the probability value.

The accurate calculation of Eq. (6), called "negative phase," is difficult because it involves all possible states of the model (Palm 2012). Researchers have presented various algorithms for calculating negative phase (Hinton 2010; Keyvanrad and Homayounpour 2015; Le Roux and Bengio 2008; Tieleman 2008; Tieleman and Hinton 2009). These algorithms differ in the choice of approximation for the gradient of the objective function. Currently, one of the most popular methods is Contrastive Divergence (CD-k),[2] especially CD-1 (Hinton 2010). The CD-1 method uses one step of Gibbs sampling. The advantages of this method include: (1) it is fast; (2) it has low variance; and (3) it is an acceptable approximation for likelihood gradient (Keyvanrad and Homayounpour 2015).

In the CD-1, for computation $\langle \boldsymbol{v_i h_j} \rangle_{\mathrm{model}}$, firstly, the visible units v_i are set to the input sample. Then the hidden states of h_j are calculated according to Eq. (11). By one-step reconstruction of visible and hidden units, v'_i and h'_j, are produced using Eqs. (10) and (11). Therefore, the weights can be updated according to Eq. (12):

$$\Delta \boldsymbol{W_{ij}} = \varepsilon\left(\langle \mathbf{v_i h_j} \rangle_{\mathrm{data}} - \langle \boldsymbol{v'_i h'_j} \rangle_{\mathrm{reconstruction}}\right) \tag{12}$$

Also, the updating rules for the biases of visible and hidden layers are as Eqs. (13) and (14):

$$\Delta \boldsymbol{b_i} = \varepsilon\left(\langle \mathbf{v_i} \rangle_{\mathrm{data}} - \langle \boldsymbol{v'_i} \rangle_{\mathrm{reconstruction}}\right) \tag{13}$$

$$\Delta \boldsymbol{c_j} = \varepsilon\left(\langle \mathbf{h_j} \rangle_{\mathrm{data}} - \langle \boldsymbol{h'_j} \rangle_{\mathrm{reconstruction}}\right). \tag{14}$$

Algorithm 1 indicates the CD-1 pseudo-code to train a RBM, which includes one step of Gibbs sampling (Palm 2012). The *rand()* function produces random uniform numbers over range (0, 1). This procedure is repeatedly called with *$v_0 = t$* sampled from the training distribution for RBM. In this algorithm, ε is a learning rate for the stochastic gradient descent in contrastive divergence.

```
Algorithm 1. CD-1 ($v_0$, $\varepsilon$, $W$, $c$, $b$)
$v_0$ is a sample from the training distribution for the RBM
$\varepsilon$ is a learning rate for the stochastic gradient descent in contrastive divergence
$W$ is the RBM weight matrix, of dimension 'number of hidden units' × 'number of inputs'
$c$ is the RBM biases vector for hidden units
$b$ is the RBM biases vector for input units

for all training sample as $t$ do
    $v_0 \leftarrow t$
    $h_0 \leftarrow sigm(v_0 W + c) > rand(\ )$
    $v_1 \leftarrow sigm(h_0 W^T + b) > rand(\ )$
    $h_1 \leftarrow sigm(v_1 W + c) > rand(\ )$
    $W \leftarrow W + \varepsilon(v_0 h_0 - v_1 h_1)$
    $b \leftarrow b + \varepsilon(v_0 - v_1)$
    $c \leftarrow c + \varepsilon(h_0 - h_1)$
end for
```

2.1.2 Extreme learning machines

The ELM is a simple and efficient learning algorithm of the single-hidden-layer feed-forward neural networks (SLFN) family, which aims at avoiding duplicate and costly training process as well as improving the generalization

[2] Contrastive Divergence with k step of Gibbs Sampling.

performance (Qu et al. 2016). In the ELM, the hidden layer does not need to be tuned (Huang et al. 2012). That is, the connection weights between the input layer and the hidden layer of the SLFN, as well as the hidden biases and neurons, are generated randomly and without additional tuning. Also, the connection weights between the hidden layer and the output layer are calculated using the efficient least squares method (Qu et al. 2016). Figure 2 shows the structure of the ELM.

The output function of ELM is as Eq. (15).

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(x) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{15}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^T$ is the vector of output weights between the hidden layer with L nodes and the output nodes. Also, $\mathbf{h}(\mathbf{x}) = [h_1(x), \ldots, h_L(x)]$ is the output vector of the hidden layer with respect to the input $\mathbf{x}$. Actually, $\mathbf{h}(\mathbf{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden layer feature space $\mathbf{H}$. For binary classification applications, the ELM decision function is as Eq. (16).

$$f_L(\mathbf{x}) = \mathrm{sigm}(\mathbf{h}(\mathbf{x})\boldsymbol{\beta}). \tag{16}$$

Unlike conventional learning algorithms (e.g., the BP algorithm), ELM tends to reach not only the smallest training error but also to the smallest norm of output weight, which leads to better network performance (Huang et al. 2012):

$$\text{Minimize: } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2, \|\boldsymbol{\beta}\| \tag{17}$$

where $\mathbf{H}$ is the hidden layer's output matrix:

$$\mathbf{H} = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \ddots & \vdots \\ h_1(x_N) & \cdots & h_L(x_N) \end{bmatrix} \tag{18}$$

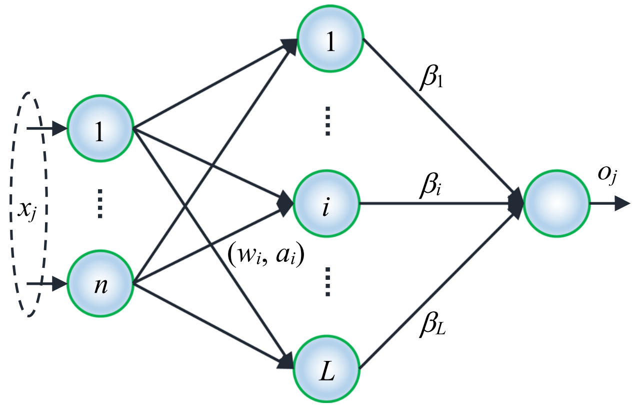


Fig. 2 Extreme learning machine

If desired matrix $\mathbf{T} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N]^T$ (Huang et al. 2012) is composed of labeled samples, the output weight $\boldsymbol{\beta}$ can be defined as Eq. (19):

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} = \mathbf{H}^T\left(\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{19}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$ (Huang et al. 2006). The ELM output layer behaves like a linear solver in the new feature space $\mathbf{H}$, and the output weights are only the system parameters that need to be tuned and can be mathematically calculated using Eq. (19).

2.2 Proposed models

BP algorithm has yielded acceptable and desirable results in shallow network. Random assignment of network initial weights in BP learning algorithm resulted in occurrence of several problems in training of deeper networks, where training of networks with more than one or two hidden layers using BP was indeed impossible. The main idea of unsupervised pre-training of DBN using RBMs was rectifying this problem and assignment of appropriate initial weights to network (to provide a good start in fine-tuning stage using BP). The process of DBN training involves three stages: (1) pre-training: training a sequence of learning modules, in a greedy layer-wise way, using unsupervised data; (2) the first fine-tuning: using random weights for the last layer; and (3) the second fine-tuning: applying back propagation for fine-tuning of the network, using supervised data. The unsolved problem here is random selecting the weights between the last hidden layer (in the last RBM) and output layer. One of the goals of $E_{(T)}$-DBN-ELM-BP model is to use ELM in the first fine-tuning stage for intelligent selecting of these weights which provide a more appropriate initial point for BP algorithm. Actually, ELM similar to RBMs provides an initial weight to network and the network main training in the second fine-tuning stage is happened using BP algorithm.

The second proposed method to improve the fine-tuning steps is $E_{(T)}$-DBN-BP-ELM model, which uses the BP and ELM algorithms in the first and second fine-tuning steps, respectively. Finally, the third proposed method, $E_{(TW)}$-DBN, uses the advantages of GA in DBN training. The all three proposed models, described in following subsections, use GA to optimize the network architecture, which is a challenging problem in using deep neural networks.

2.2.1 Topology evolving of DBN-ELM-BP algorithm: $E_{(T)}$-DBN-ELM-BP

To have a better understanding about this model, first, the new training method, DBN-ELM-BP, is described. Given the local search property of the error back-propagation algorithm, if we can use non-random and more appropriate network weights instead of using random weights at the beginning of this algorithm, the algorithm converges earlier and leads to better classification performance. This is also the philosophy of DBN pre-training; however, the problem is that the network can be pre-trained only up to the last hidden layer and the weights between the last hidden layer and the output layer are randomly selected.

DBN-ELM-BP solves this problem using the ELM classifier. In this model, after an unsupervised pre-training of network and a supervised ELM classification, a supervised BP stage is added to the DBN-ELM network. Figure 3 illustrates the process of training this model graphically. In the methodology of this model a DBN unsupervised pre-training stage is first performed. Then, the ELM classifier is used to calculate the weights between the last hidden layer and the output layer (the first fine-

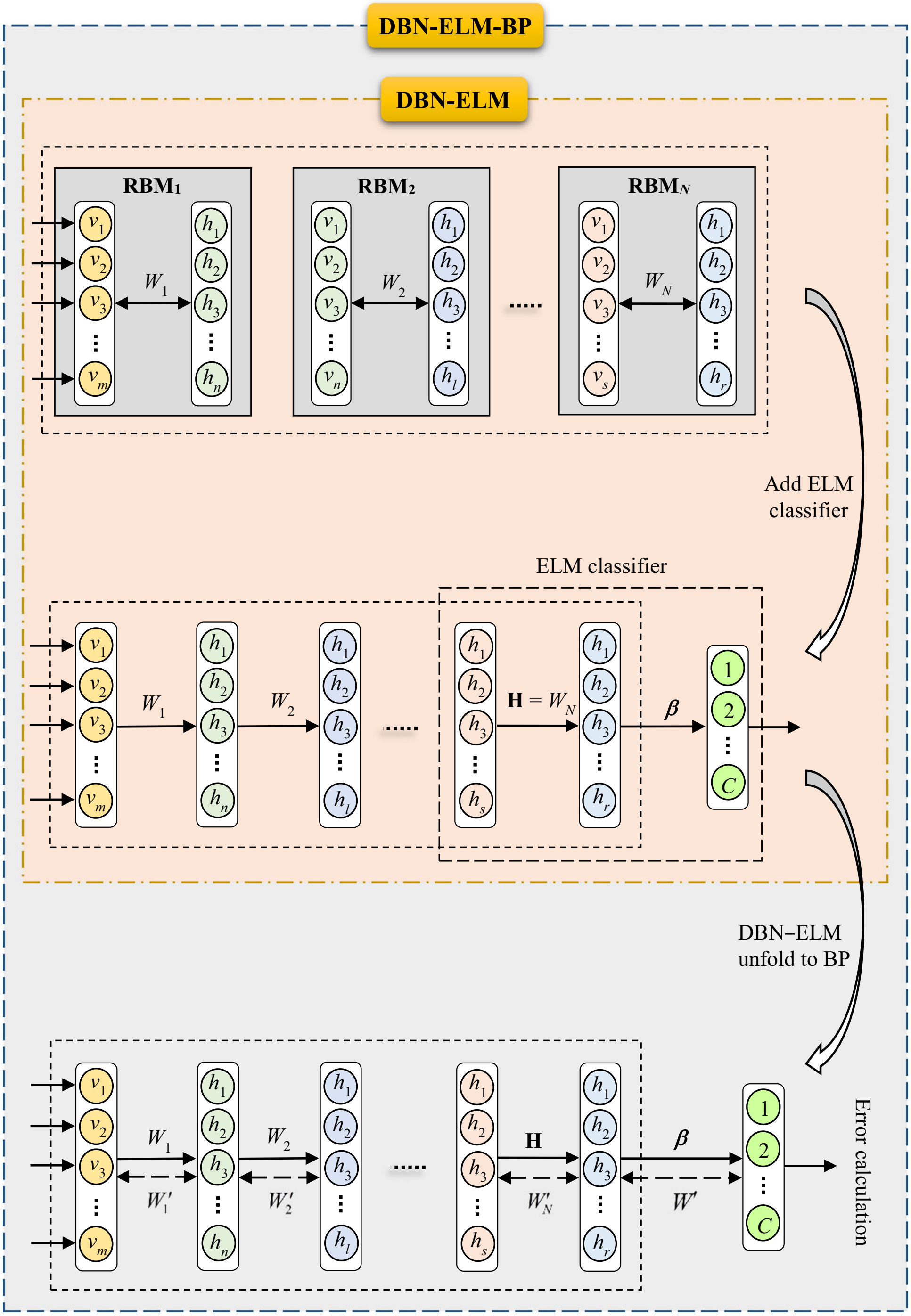


Fig. 3 The network structures and the training steps of the DBN-ELM-BP

tuning stage), so that the matrix **H** is considered equal to the weights matrix obtained from the last RBM of the DBN (i.e., W_N) and the matrix $\boldsymbol{\beta}$ is calculated. Finally, the error is computed, and BP updates the weights matrix (shown below the dash arrows) and trains the entire network (second fine-tuning stage). The dash arrows in Fig. 3 represent the update of the weights matrix by the BP algorithm.

In the evolutionary model $E_{(T)}$-DBN-ELM-BP, network topology is optimized by GA. The fitness function of this model for each chromosome is the classification accuracy obtained from the DBN-ELM-BP network.

2.2.2 Topology evolving of DBN-BP-ELM algorithm: $E_{(T)}$-DBN-BP-ELM

In the DBN-BP-ELM training method, after the pre-training stage of the network, the first fine-tuning stage is performed by the BP and then the ELM classifier is used for the second fine-tuning stage. The network structure and

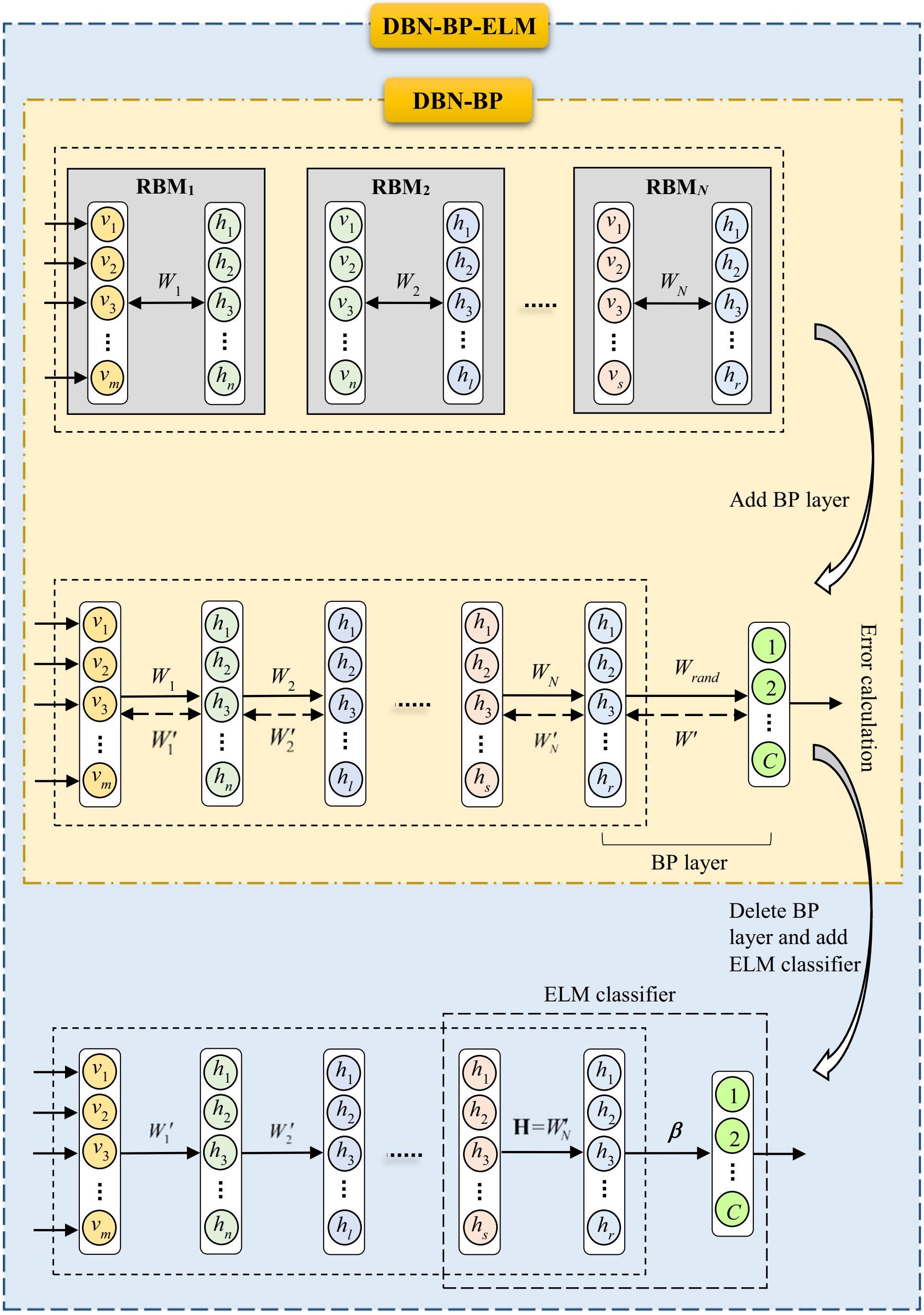


Fig. 4 The network structures and the training steps of the DBN-BP-ELM

training process of this model are graphically illustrated in Fig. 4. The evolutionary model E$_{(T)}$-DBN-BP-ELM uses GA to find the optimal or near optimal architecture for the network. The fitness function for evaluating the chromosomes of this model is the classification accuracy obtained from the network trained by the DBN-BP-ELM method.

In the following, the steps and operators of these two evolutionary models to find the optimal structure of the DBN are described. The steps needed for evolving network topology are described below:

Step 1. Chromosomes encoding

In the two described models, each chromosome is a deep neural network with direct coding, i.e., the network topology is directly represented by the vector of positive integers. The number of genes in each chromosome indicates the number of hidden layers of the deep network and the value of each gene represents the number of neurons in the corresponding hidden layer. Figure 5 illustrates an example of the chromosome representation and its corresponding DBN for a problem with nine input features x_1, x_2, …, and x_9, as well as an output C specifying the class label.

Step 2. Population initialization

The population size (number of chromosomes) is M which are randomly generated. To reduce the very large search space of the problem, the maximum number of genes of the initial population (chromosomes) is assigned to five genes. The value of each gene is also initialized randomly in a given interval considering the number of input features of each data set. It is possible to increase or decrease the number of genes and the value of each gene by genetic operators, during the running the algorithm.

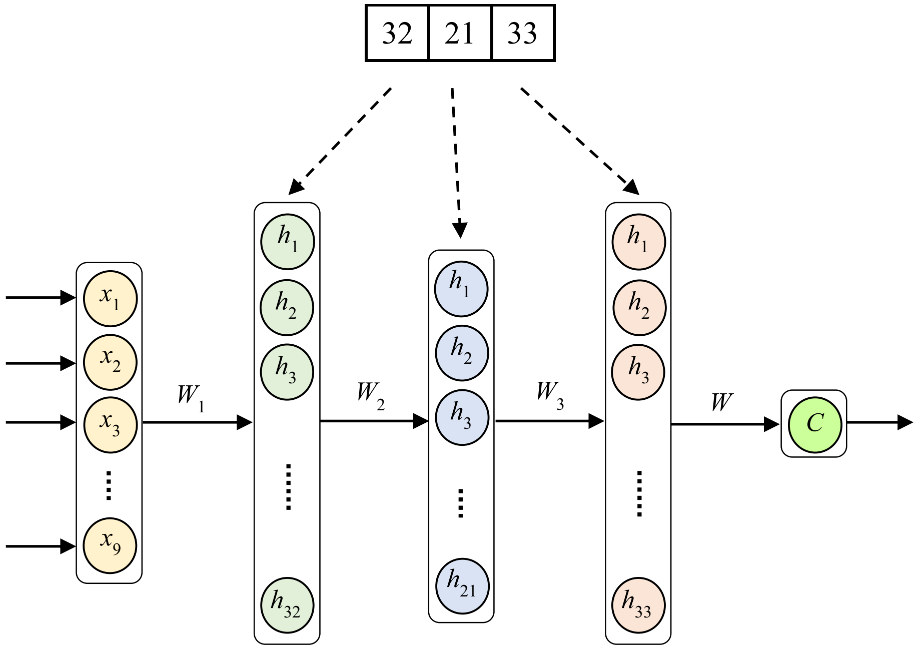


Fig. 5 An example of chromosome encoding in E(T)-DBN-ELM-BP and E(T)-DBN-BP-ELM

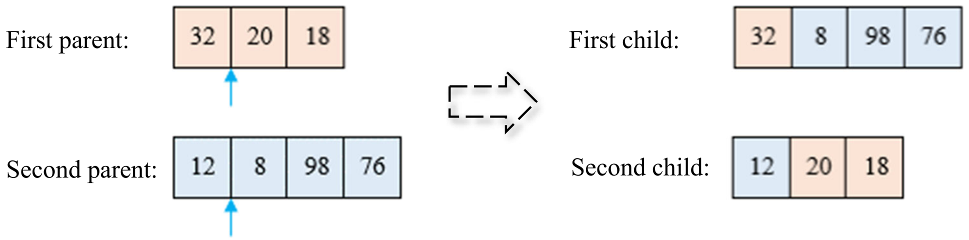


Fig. 6 An example of one-point crossover in E(T)-DBN-ELM-BP and E(T)-DBN-BP-ELM

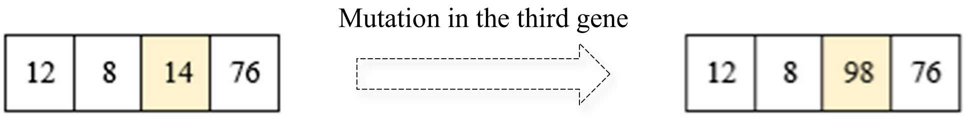


Fig. 7 An example of mutation in E(T)-DBN-ELM-BP and E(T)-DBN-BP-ELM

Step 3. Evaluation

In order to evaluate and obtain the fitness value for each chromosome in each model, the accuracy percentage of the training data with respect to the relevant model is calculated. The classification accuracy percentage is calculated using confusion matrix elements as Eq. (20):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100 \tag{20}$$

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

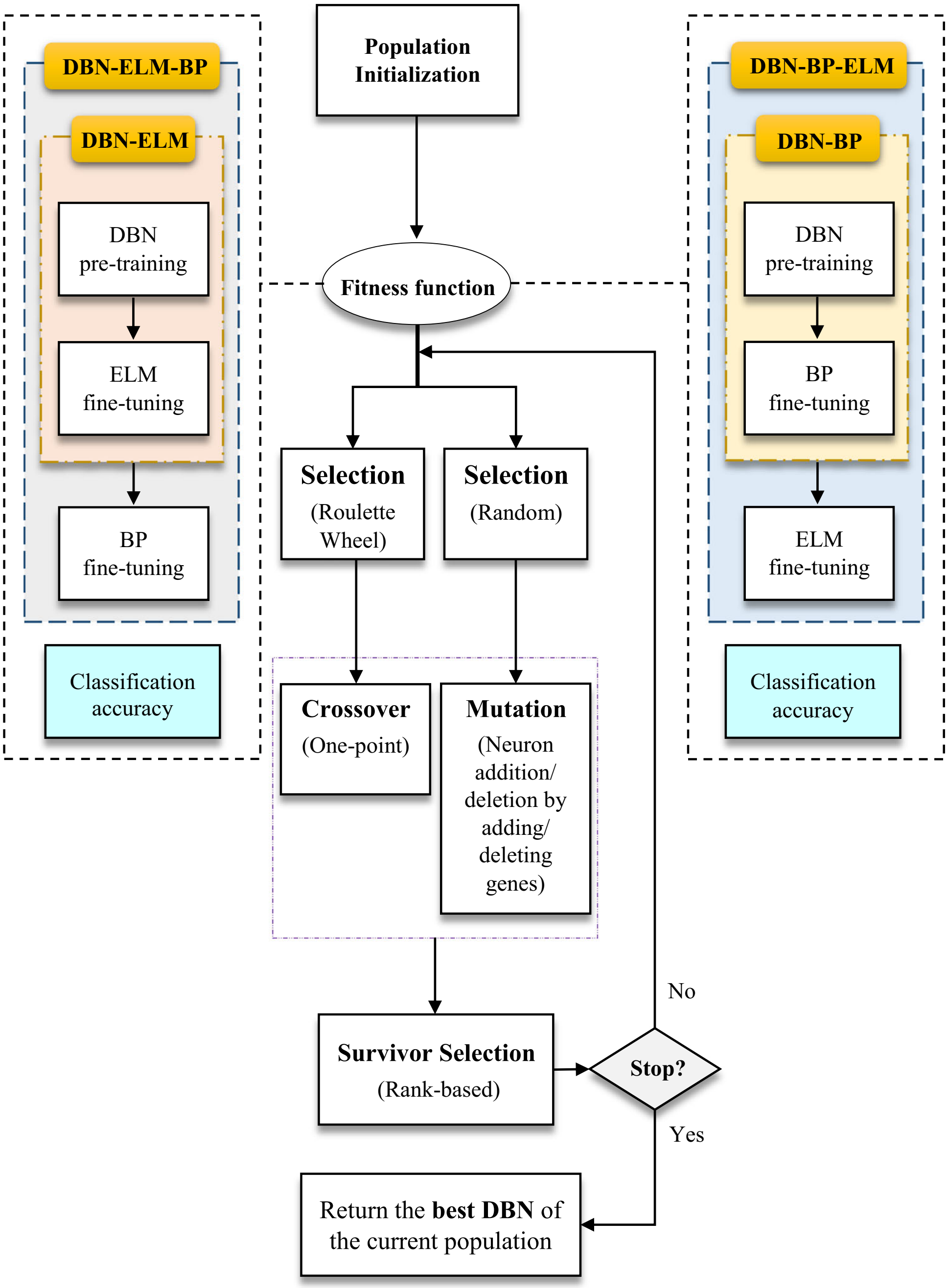
Step 4. Selection

In order to select the parents for crossover, the well-known roulette wheel selection mechanism is used. A random selection mechanism is also used so as to select a chromosome for mutation.

Step 5. Crossover

After the parent selection process, a single-point crossover, which is the most popular crossover operator in the literature, is used to generate the new offsprings and search in the solutions space. This operator, via random selection of the crossover point, generates chromosomes of varying

Fig. 8 The GA framework for E(T)-DBN-ELM-BP and E(T)-DBN-BP-ELM



lengths. Figure 6 shows an example of the single-point crossover in these two topology evolving models. The random point of the crossover is shown with a blue arrow.

Step 6. Mutation

First, a chromosome is selected randomly. Then, the mutation operator generates a new chromosome by randomly selecting a gene (a layer) and reducing or increasing its value (the number of neurons in that layer). The aim of this operator is to avoid being trapped in local optima, through exploration of new solutions space. Figure 7 shows an example of the mutation operator in the two topology evolving models. In this example, the mutation operator reduces the number of hidden neurons in the third hidden layer of the network from 14 to 98.

Step 7. Survivor selection

At this step, the current population's chromosomes and the chromosomes obtained from the crossover and the mutation are all sorted in descending order based on their fitness values. Then, M chromosomes with better fitness value/rank are used to form the new populations as survivors.

Step 8. Stopping criteria

When the number of generations (iterations) reaches to the predefined maximum number of generations, the algorithm stops; otherwise, it returns to Step 4 to generate a new generation. Figure 8 summarizes the GA framework for obtaining an optimal or near optimal topology from the two $E_{(T)}$-DBN-ELM-BP and $E_{(T)}$-DBN-BP-ELM models.

2.2.3 Topology and weights evolving of DBN: $E_{(TW)}$-DBN

The third proposed model of this paper is called $E_{(TW)}$-DBN, in which GA is used to optimize network topology and weights, simultaneously. In this model, a number of initial population chromosomes are pre-trained by DBNs. The GA steps for evolving of network topology and weights in $E_{(TW)}$-DBN model are described below:

Step 1. Chromosome encoding

The used chromosome in $E_{(TW)}$-DBN includes two parts: topology and weight. The number of genes in the topology part represents the number of hidden layers and the value of each gene signifies the number of neurons in that layer.

Step 2. Population initialization

In the initial population, the maximum number of genes in the topology part of a chromosome is limited to five genes, and the value of each gene is selected randomly in an interval according to the number of features of the input data set. To generate the initial population, the topology is first generated randomly. Then, based on the generated topologies, a vector of real numbers is generated randomly over range (0, 1), for the chromosome's weight part. The pre-trained DBN weights are used for weight part of $K = \frac{1}{5}M$ chromosome (M is the number of initial population chromosomes). Figure 9 illustrates a simple example of chromosome encoding in the $E_{(TW)}$-DBN model.

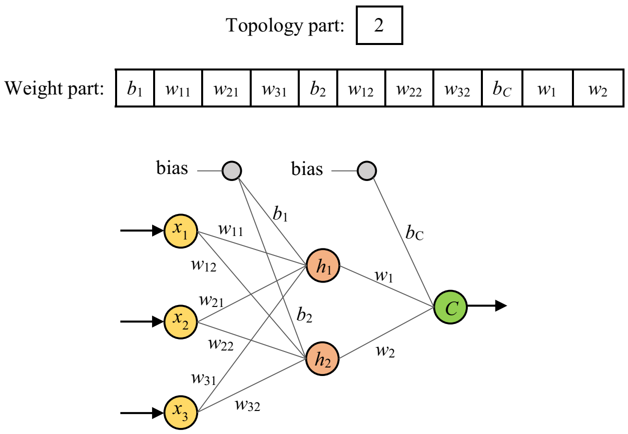


Fig. 9 A simple example of chromosome encoding in E(TW)-DBN

Step 3. Evaluation

The fitness of each chromosome is obtained through a feed-forward pass and calculating the classification accuracy percentage for training data.

Step 4. Selection

Parent selection for crossover and generating two new offsprings are carried out using the roulette wheel selection mechanism. A random selection method is also used to select the chromosomes for mutation.

Step 5. Crossover

Given that the chromosomes have two parts, the crossover operator with a predefined probability is applied to the weight part of two parents, while with another probability is applied to the topology part. This means that, a random number is firstly generated over (0, 1). If this value is less than or equal to 0.7, then crossover is applied to the weight part of the two parents, otherwise to the topology part.

Intermediate crossover is used for weight part crossover. In this operator, due to the unequal length of the weight parts and dependence of the length of this part to the topology part, a point in the parent with a shorter length is first randomly selected. Then, the mean is taken from the genes of the two parents up to the crossover point. The remaining genes of the two generated offsprings will be the same as their parents. In this way, the new weights are still proportional to the topology part (since the number of genes of each offspring is equal to its parent).

Figure 10 schematically shows the crossover operator for the weight part of the two parents, through a simple example. The randomly selected crossover point is shown

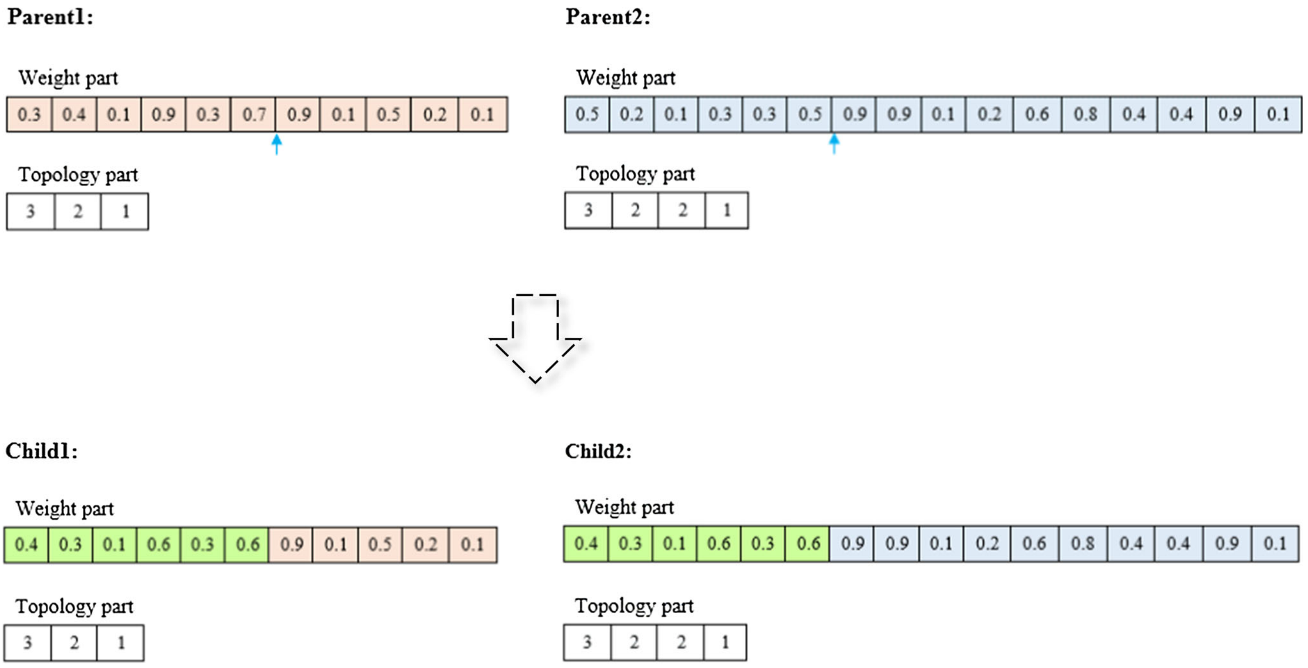


Fig. 10 An example of crossover in weight part of two chromosomes for E(TW)-DBN

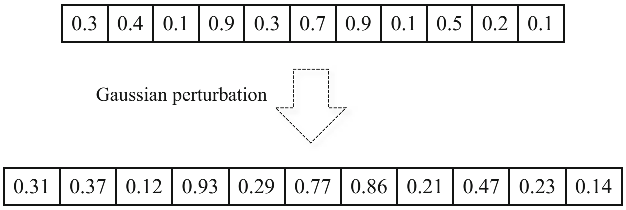


Fig. 11 An example of the mutation in weight part of E(TW)-DBN

with blue arrow. For the topology part of the chromosomes, the single-point crossover shown in Fig. 6 is used. By changing the child's chromosome length or the genes value, which occurs when the crossover is applied to the topology part, the weights part should also be regenerated and updated according to the new topology. Due to this reason, the crossover probability of the weight part is initially considered more.

Step 6. Mutation

The mutation and crossover operators with a given probability are applied to the weight part, while with another probability are applied to the topology part. In the case of mutation in the weight part, for each gene a random amount drawn from a Gaussian distribution, with mean zero and standard deviation selected at random from {0.2, 0.5, 1, 2, 5}, is added to the current gene value (Ahmadizar et al. 2015). Figure 11 graphically illustrates an example of the mutation on the weight part. In this case, the topology part of the offspring will be the same as that of its parents. Mutation in the topology part of chromosome is applied by random selection of a gene and increasing or decreasing its value. This is done through increasing or decreasing a number in range of [1, the amount of that gene] (as in Fig. 7). In this case, the weigh part needs to be updated according to the topology part.

Step 7. Survivor Selection

To select the survivor, the ranked-based selection mechanism is used.

Step 8. Stopping criteria

The GA stopping criteria in the $E_{(TW)}$-DBN model is to reach to a certain number of generations. If the stopping criteria are not met, the algorithm returns to Step 4 so as to create the new population.

3 Experimental results

In this section, performance evaluation of the proposed models is carried out using classification accuracy, sensitivity, and specificity measures on breast cancer data sets. The calculating of classification accuracy is shown in Eq. (20), while the sensitivity and specificity are also calculated as Eqs. (21) and (22):

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \tag{21}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \times 100 \tag{22}$$

In addition to the above-mentioned measures, the receiver operating characteristic (ROC) curve (Bradley

Table 2 Properties of the datasets used in the experiments

Datasets	# of benign instances	# of malignant instances	# of total instances	# of attributes	# of classes
WBCO	444	239	683	9	2
WDBC	357	212	569	30	2

Table 3 Genetic algorithm parameters for evolutionary models

Parameters	Models	
	E(TW)-DBN	E(T)-DBN-ELM-BP, E(T)-DBN-BP-ELM, E(T)-DBN-BP & E(T)-DBN-ELM
# of generations	400	100
Population size	50[a]	40[a]
	60[b]	50[b]
Crossover rate	0.7	0.8
Mutation rate	0.2	0.15
# of genes	Topology part: random in [1, 5]	Random in [1, 5]
	Weight part: according to the topology	
Value of genes	Topology part: random in [0, 100][a]	Random in [0, 100][a]
	Random in [0, 200][b]	Random in [0, 200][b]
	Weight part: random in [0, 1]	

[a] For WBCO; [b] for WDBC

1997) and the area under the ROC curve (AUC) (Huang and Ling 2005) are also applied for models evaluation.

The performance of the proposed algorithms ($E_{(T)}$-DBN-ELM-BP, $E_{(T)}$-DBN-BP-ELM and $E_{(TW)}$-DBN) is investigated using two breast cancer data sets. For further evaluation, the results of the proposed algorithms are compared with the results of $E_{(T)}$-DBN-BP and $E_{(T)}$-DBN-ELM models (proposed and implemented by the authors). The $E_{(T)}$-DBN-BP model optimizes the architecture for a DBN which uses BP to fine-tuning the network (the inner part of Fig. 4 depicts the DBN-BP training process graphically). The network training process in $E_{(T)}$-DBN-ELM model is also shown in the inner part of Fig. 3.

3.1 Data sets

To evaluate the proposed algorithms' performance, two available and most popular breast cancer data sets, namely Breast Cancer Wisconsin—Original (WBCO) and Breast Cancer Wisconsin—Diagnostic (WDBC) on UCI machine learning repository are used in the experiments. The main features of these data sets are summarized in Table 2, including the number of benign instances, the number of malignant instances, the total number of instances, the number of attributes, as well as the number of classes for each data set. The total number of samples in WBCO data set is 699. Among them, 16 samples were rejected due to incomplete features. So, we reduced the used samples down to 683 entries to compare our results easier with the others.

3.2 Implementation

MATLAB R2013a software has been used to implement the proposed models. The partition of training–testing data in each data set is considered 80–20% for all models (Garro et al. 2016; Pham and Sagiroglu 2000). In each run, the training and testing data are selected among the main data sets randomly for each neural network.

3.3 Performance evaluation of the proposed models

This subsection evaluates the classification performance of the three proposed evolutionary algorithms: $E_{(T)}$-DBN-ELM-BP, $E_{(T)}$-DBN-BP-ELM, and $E_{(TW)}$-DBN. For further evaluation, the results of the $E_{(T)}$-DBN-BP and $E_{(T)}$-DBN-ELM models implemented by the authors are also compared with the proposed algorithms.

3.3.1 Parameters tuning

Many parameters may impact how algorithm yields appropriate solutions. Various combinations of parameters can lead to a variety of solutions. Therefore, in this subsection, the focus is on tuning the parameters of $E_{(T)}$-DBN-

ELM-BP, $E_{(T)}$-DBN-BP-ELM, $E_{(TW)}$-DBN, $E_{(T)}$-DBN-BP, and $E_{(T)}$-DBN-ELM models. Table 3 shows the GA parameters used for evolutionary models. These parameters are set through trial-and-error method.

3.3.2 Classification performance

Each proposed evolutionary model is run 20 times and the average results are then calculated and shown in Table 4, including the average classification accuracy, sensitivity and specificity of the models. In this table, the topology obtained from the best run of the algorithms is also shown for each model.

According to the results presented in Table 4, following observations and analysis can be extracted:

(a) The highest classification accuracy for the WBCO data set is 99.75%, which is derived from the proposed $E_{(T)}$-DBN-ELM-BP. In the WDBC data set, the highest accuracy is 99.12% obtained from the proposed $E_{(T)}$-DBN-BP-ELM. These good results are due to the two steps of the DBN fine-tuning and the use of an efficient ELM classifier.

(b) The $E_{(TW)}$-DBN model has the least accuracy (98.05%) compared to other models on the WBCO data set, but the accuracy of this model on the WDBC data set ranks the third (with an accuracy of 98.54%).

(c) The best possible sensitivity, 100%, has been derived for the WBCO data set by $E_{(T)}$-DBN-ELM-BP, $E_{(T)}$-DBN-BP-ELM and $E_{(T)}$-DBN-ELM models. This output is also yielded for the WDBC data set by $E_{(T)}$-DBN-ELM-BP model.

(d) In general, in both experimental data sets, classification performance of models that use only one fine-tuning step (i.e., $E_{(T)}$-DBN-BP and $E_{(T)}$-DBN-ELM) has been lower than that of the proposed models.

(e) The proposed model $E_{(T)}$-DBN-ELM-BP outperforms the other methods in "accuracy" and "sensitivity" measures on the WBCO data set and has yielded a deep topology of 9-48-19-1.

3.3.3 ROC curves and AUC values

The ROC curve is a graphical plot and a fundamental tool for evaluating a diagnostic classification system. This curve has two dimensions: true positive rate (*x*-axis) versus false positive rate (*y*-axis). Each point on the ROC curve shows a pair of sensitivity–specificity related to a specific threshold of a decision. The AUC is a suitable measure for evaluating the performance of a disease diagnostic system. The AUC value ranges over [0, 1]. The closer this value to 1 is, the more reliable the diagnostic system is Asadi and Shahrabi (2016, 2017), Asadi et al. (2013), and Fotouhi et al. (2019).

Figures 12 and 13 show ROC curves and their AUC values for WBCO and WDBC data sets, respectively. According to these figures and the AUC value shown in each ROC curve, one can conclude that the two proposed models $E_{(T)}$-DBN-BP-ELM and $E_{(T)}$-DBN-ELM-BP, with two steps of fine-tuning and using the ELM in one of these two steps, can obtain the highest AUC (i.e., AUC = 1) in both data sets.

4 Discussion

In this section, at first, the classification accuracy of the proposed methods is compared to some of the data mining techniques since 2011 introduced in Sect. 1. Then, performance analyses of the proposed models are presented.

Table 5 presents the abbreviation of the data mining techniques existing in the WBCO classification literature, along with its accuracy. Due to the use of decision tree

Table 4 Classification performance of topology evolving models and their optimized topology

Models	Datasets							
	WBCO				WDBC			
	Accuracy (%)	Sensitivity (%)	Specificity (%)	Topology	Accuracy (%)	Sensitivity (%)	Specificity (%)	Topology
E(T)-DBN-ELM-BP	**99.75**	**100**	98.80	9-48-19-1	98.83	**100**	98.27	30-37-113-1
E(T)-DBN-BP-ELM	99.45	**100**	**99.17**	9-27-42-1	**99.12**	98.81	**99.31**	30-19-274-1
E(TW)-DBN	98.05	99.46	97.29	9-18-1-1	98.54	98.55	98.61	30-12-1
E(T)-DBN-BP	98.29	99.26	97.86	9-32-21-33-1	97.95	97.36	98.19	30-68-1
E(T)-DBN-ELM	98.91	**100**	98.32	9-98-1	96.49	95.29	97.19	30-277-1

Bold indicates the best result in each criterion

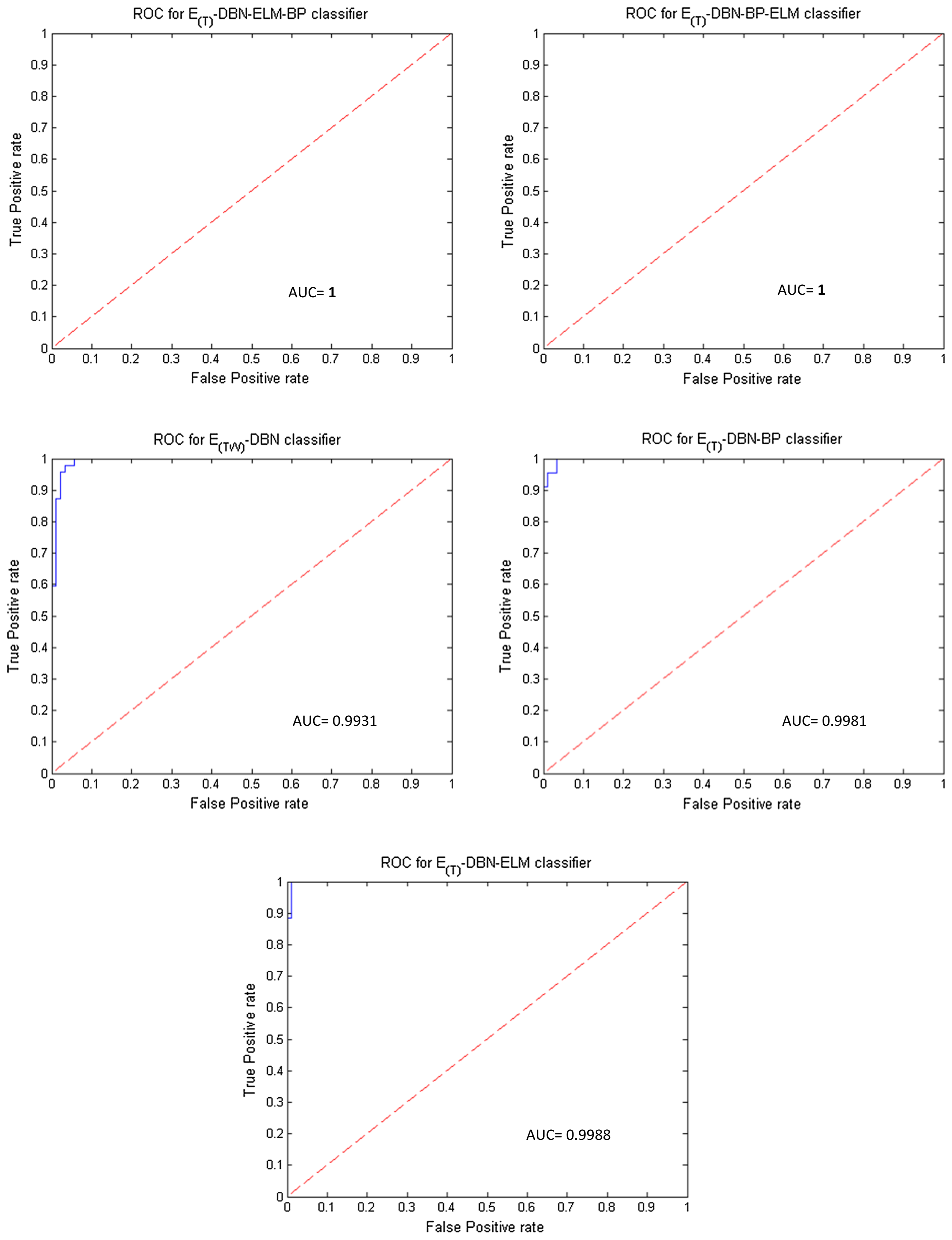


Fig. 12 ROC curves and AUC values of classifiers for WBCO

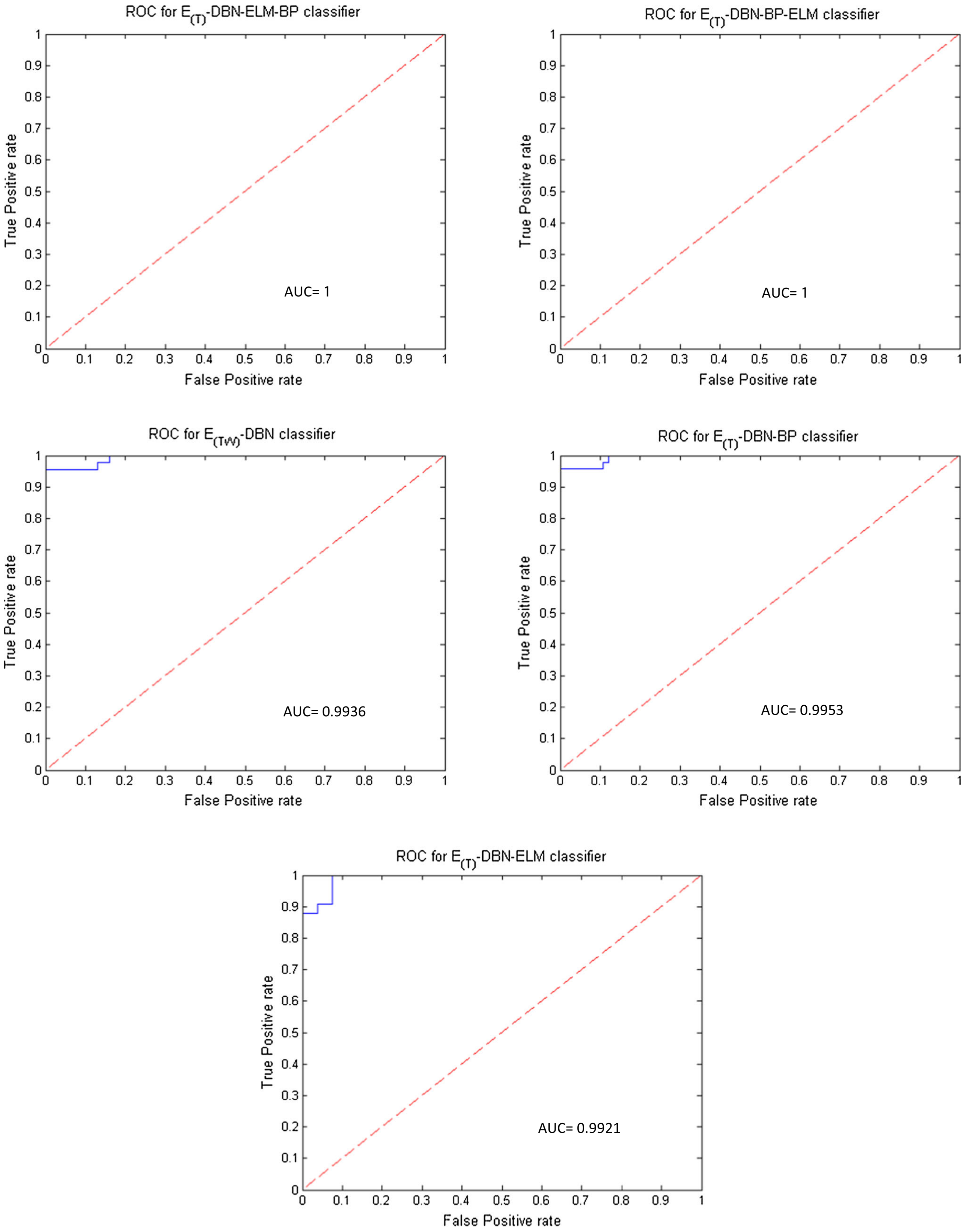


Fig. 13 ROC curves and AUC values of classifiers for WDBC

Table 5 Accuracy of data mining methods in WBCO classification since 2011 until now

Methods name	Year	References	Accuracy (%)
BP	2011	Örkcü and Bal (2011)	93.1
BC-GA			94
RC-GA			96.5
AMMLP	2011	Marcano-Cedeño et al. (2011)	99.26
DT-1	2011	Lavanya and Rani (2011)	94.84
ICA-MLP	2013	Malmir et al. (2013)	97.75
PSO-MLP			97.63
RF-ANN	2013	Koyuncu and Ceylan (2013)	98.05
DT-2	2014	Sumbaly et al. (2014)	94.56
GONN	2015	Bhardwaj and Tiwari (2015)	99.26
W-NB	2015	Karabatak (2015)	98.54
DBN-NN-LM	2016	Abdel-Zaher and Eldeib (2016)	99.68

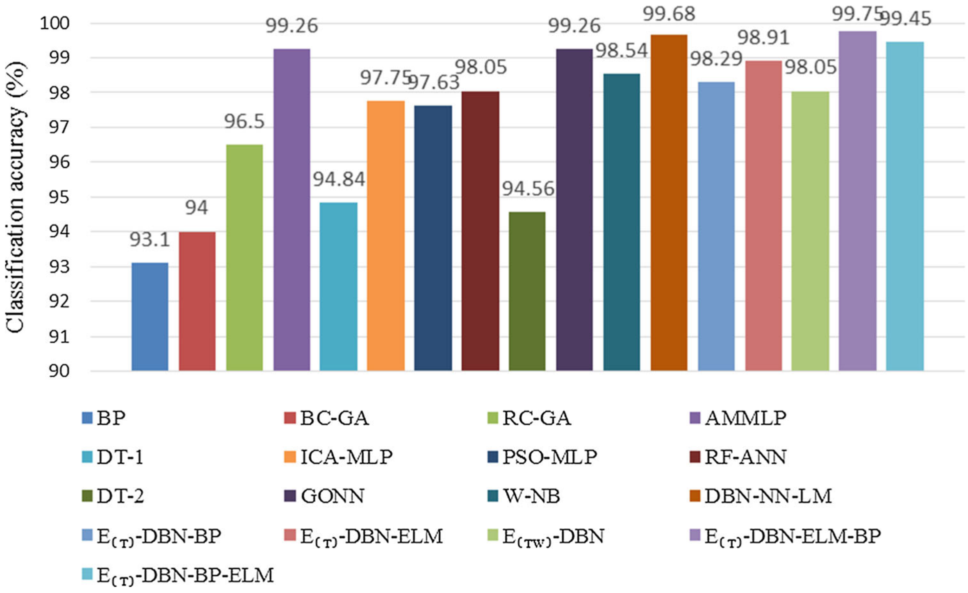


Fig. 14 The comparison of classification accuracy percentages of proposed models with other models in the literature for WBCO

method in two separate studies, the numbers 1 and 2 are located next to the abbreviated name.

Figure 14 graphically shows the classification performance results of the proposed methods and the techniques presented in Table 5 belonging to the WBCO.

According to Fig. 14, the superiority of $E_{(T)}$-DBN-ELM-BP method with a mean accuracy of 99.75% is evident compared to other methods in classification of WBCO. Subsequently, the DBN-NN-LM method with an accuracy of 99.68% and the proposed $E_{(T)}$-DBN-BP-ELM technique with an accuracy of 99.45%, had the best performances, respectively.

It should be noted that the providers of the DBN-NN-LM method have tested different partitions for training–testing data and achieved an accuracy of 99.68% at a specific partition 54.9–45.1%, while the proposed models in this article use a partition 80–20% for training–testing data sets, which randomly select among the main data set at each run.

In Table 6, the classification accuracy of different data mining methods is shown on the WDBC data set.

Figure 15 presents the proposed methods' performance in graphical form compared to other methods in the cancer detection literature that is presented in Table 6 on the WDBC data set.

According to Fig. 15, the proposed $E_{(T)}$-DBN-BP-ELM method with a mean accuracy of 99.12% ranks first in the classification performance, compared to other methods. Subsequently, two other proposed methods, $E_{(T)}$-DBN-ELM-BP and $E_{(TW)}$-DBN, stand on the second and third

Table 6 Accuracy of data mining methods in WDBC classification since 2011 until now

Methods name	Year	References	Accuracy (%)
DT-1	2011	Çınar et al. (2009)	92.97
Xue14	2014	Xue et al. (2014)	94.74
K-SVM	2014	Zheng et al. (2014)	97.38
JSDA	2015	Kong et al. (2015)	93.85

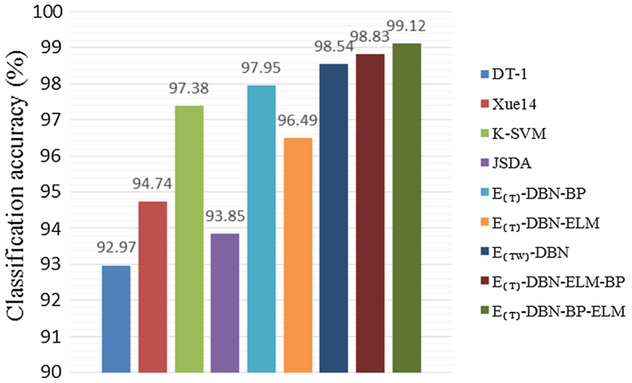


Fig. 15 The comparison of classification accuracy percentages of proposed models with other models in the literature for WDBC

positions, respectively. Figures 14 and 15 suggest that combining the DBN with ELM for the diagnosis of breast cancer is very attractive, having two steps of the DBN fine-tuning using an efficient ELM classifier.

In general, one can claim that the proposed methods in this paper, especially $E_{(T)}$-DBN-ELM-BP and $E_{(T)}$-DBN-BP-ELM, improve the classification performance in both WBCO and WDBC data sets.

5 Conclusion and future work

In this paper, three models for the diagnosis of breast cancer were developed based on DBN. A common method for DBNs training is using an unsupervised pre-training phase (restricted Boltzmann machine training with a layer-wise manner using CD-1 algorithm) and a supervised fine-tuning phase (applying back-propagation algorithm). In the two presented models in this paper called $E_{(T)}$-DBN-ELM-BP and $E_{(T)}$-DBN-BP-ELM, the extreme learning machine (ELM) and the back-propagation (BP) algorithm were applied in the DBN fine-tuning. In the third proposed model, $E_{(TW)}$-DBN, the genetic algorithm (GA) was applied to the DBN fine-tuning. In addition, these models use the GA to optimize the DBN structure in order to answer this question: how many hidden layers and how many neurons in each layer in DBNs should be used.

To evaluate the proposed models, extensive experiments were carried out and models were compared in different aspects. Classification accuracy, sensitivity, specificity, and AUC measurements for classifiers were used to evaluate the proposed models. In summary, the following results were obtained concerning classification performance of the proposed models:

(a) The first two proposed models in the breast cancer diagnosis have achieved remarkable performance over $E_{(TW)}$-DBNs and existing approaches in the literature, with an accuracy of 99.75% through $E_{(T)}$-DBN-ELM-BP model on WBCO data set and an accuracy of 99.12% through $E_{(T)}$-DBN-BP-ELM on the WDBC data set.

(b) The high AUC value of the $E_{(T)}$-DBN-ELM-BP and $E_{(T)}$-DBN-BP-ELM (i.e., AUC = 1), indicates a very good diagnostic performance of these two models in detecting breast cancer.

Considering the impressive performance of the first two proposed models for cancer diagnosis, it can be expected that these models also perform well in other cancer data sets. Consequently, it is suggested that the proposed models be used to diagnose other types of cancer data. Also, to improve individual performance of the classifiers, using these models in providing an ensemble learning approach seems appropriate.

Due to the success combination of the ELM with DBN, its combination with a different kind of deep network such as autoencoder is suggested for future researches.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest to this work and this study was not funded by any grant.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abdel-Zaher AM, Eldeib AM (2016) Breast cancer classification using deep belief networks. Expert Syst Appl 46:139–144

Abonyi J, Szeifert F (2003) Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recogn Lett 24:2195–2207

Ahmadizar F, Soltanian K, AkhlaghianTab F, Tsoulos I (2015) Artificial neural network development by means of a novel combination of grammatical evolution and genetic algorithm. Eng Appl Artif Intell 39:1–13

Albrecht AA, Lappas G, Vinterbo SA, Wong C, Ohno-Machado L (2002) Two applications of the LSA machine, neural information processing, 2002. In: Proceedings of the 9th international conference on ICONIP'02. Publishing, pp 184–189

Asadi S (2019) Evolutionary fuzzification of RIPPER for regression: case study of stock prediction. Neurocomputing 331:121–137

Asadi S, Shahrabi J (2016) ACORI: a novel ACO algorithm for Rule Induction. Knowl Based Syst 97:175–187

Asadi S, Shahrabi J (2017) Complexity-based parallel rule induction for multiclass classification. Inf Sci 380:53–73

Asadi S, Hadavandi E, Mehmanpazir F, Nakhostin MM (2012) Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. Knowl Based Syst 35:245–258

Asadi S, Shahrabi J, Abbaszadeh P, Tabanmehr S (2013) A new hybrid artificial neural networks for rainfall–runoff process modeling. Neurocomputing 121:470–480

Bengio Y (2009) Learning deep architectures for AI. Foundations and trends® Mach Learn 2:1–127

Bhardwaj A, Tiwari A (2015) Breast cancer diagnosis using genetically optimized neural network model. Expert Syst Appl 42:4611–4620

Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30:1145–1159

Cao L-L, Huang W-B, Sun F-C (2016) Building feature space of extreme learning machine with sparse denoising stacked-autoencoder. Neurocomputing 174:60–71

Chen H-L, Yang B, Liu J, Liu D-Y (2011) A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. Expert Syst Appl 38:9014–9022

Çınar M, Engin M, Engin EZ, Ateşçi YZ (2009) Early prostate cancer diagnosis by using artificial neural networks and support vector machines. Expert Syst Appl 36:6357–6361

Ciompi F, de Hoop B, van Riel SJ, Chung K, Scholten ET, Oudkerk M, de Jong PA, Prokop M, van Ginneken B (2015) Automatic classification of pulmonary peri-fissural nodules in computed tomography using an ensemble of 2D views and a convolutional neural network out-of-the-box. Med Image Anal 26:195–202

Deng L, Yu D (2014) Deep learning: methods and applications. Found Trends Signal Process 7:197–387

Flores-Fernández JM, Herrera-López EJ, Sánchez-Llamas F, Rojas-Calvillo A, Cabrera-Galeana PA, Leal-Pacheco G, González-Palomar MG, Femat R, Martínez-Velázquez M (2012) Development of an optimized multi-biomarker panel for the detection of lung cancer based on principal component analysis and artificial neural network modeling. Expert Syst Appl 39:10851–10856

Fotouhi S, Asadi S, Kattan MW (2019) A comprehensive data level analysis for cancer diagnosis on imbalanced data. J Biomed Informs 90:1–30

Frénay B, Verleysen M (2011) Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing 74:2526–2531

Garro BA, Rodríguez K, Vázquez RA (2016) Classification of DNA microarrays using artificial neural networks and ABC algorithm. Appl Soft Comput 38:548–560

Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2015) Deep learning for visual understanding: A review. Neurocomputing 187:27–48

Hinton G (2010) A practical guide to training restricted Boltzmann machines. Momentum 9:926

Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554

Hrasko R, Pacheco AG, Krohling RA (2015) Time series prediction using restricted Boltzmann machines and backpropagation. Proc Comput Sci 55:990–999

Huang J, Ling CX (2005) Using AUC and accuracy in evaluating learning algorithms. IEEE Trans Knowl Data Eng 17:299–310

Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501

Huang G-B, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74:155–163

Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cyber Part B Cybern 42:513–529

Karabatak M (2015) A new classifier for breast cancer detection based on Naïve Bayesian. Measurement 72:32–36

Kazemi S, Hadavandi E, Shamshirband S, Asadi S (2016) A novel evolutionary-negative correlated mixture of experts model in tourism demand estimation. Comput Hum Behav 64:641–655

Keyvanrad MA, Homayounpour MM (2015) Deep belief network training improvement using elite samples minimizing free energy. Int J Pattern Recognit Artif Intell 29:1551006

Kong H, Lai Z, Wang X, Liu F (2015) Breast cancer discriminant feature analysis for diagnosis via jointly sparse learning. Neurocomputing

Koyuncu H, Ceylan R (2013) Artificial neural network based on rotation forest for biomedical pattern classification. In: 2013 36th international conference on telecommunications and signal processing (TSP). Publishing, pp 581–585

Längkvist M, Karlsson L, Loutfi A (2014) A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recogn Lett 42:11–24

Larochelle H, Erhan D, Courville A, Bergstra J, Bengio Y (2007) An empirical evaluation of deep architectures on problems with many factors of variation. In: Proceedings of the 24th international conference on machine learning. Publishing, pp 473–480

Lavanya D, Rani DKU (2011) Analysis of feature selection with classification: breast cancer datasets. IJCSE 2:756–763

Le Roux N, Bengio Y (2008) Representational power of restricted Boltzmann machines and deep belief networks. Neural Comput 20:1631–1649

Liu Q, He Q, Shi Z (2008) Extreme support vector machine classifier, Pacific-Asia conference on knowledge discovery and data mining. Publishing, pp 222–233

Malmir H, Farokhi F, Sabbaghi-Nadooshan R (2013) Optimization of data mining with evolutionary algorithms for cloud computing application. In: 2013 3rd international econference on computer and knowledge engineering (ICCKE). Publishing, pp 343–347

Mansourypoor F, Asadi S (2017) Development of a reinforcement learning-based evolutionary fuzzy rule-based system for diabetes diagnosis. Comput Biol Med 91:337–352

Marcano-Cedeño A, Quintanilla-Domínguez J, Andina D (2011) WBCD breast cancer database classification applying artificial metaplasticity neural network. Expert Syst Appl 38:9573–9579

Mehmanpazir F, Asadi S (2017) Development of an evolutionary fuzzy expert system for estimating future behavior of stock price. J Ind Eng Int 13:29–46

Milovic B (2012) Prediction and decision making in health care using data mining. IJPHS 1:69–78

Nauck D, Kruse R (1999) Obtaining interpretable fuzzy classification rules from medical data. Artif Intell Med 16:149–169

Örkcü HH, Bal H (2011) Comparing performances of backpropagation and genetic algorithms in the data classification. Expert Syst Appl 38:3703–3709

Palm RB (2012) Prediction as a candidate for learning deep hierarchical models of data. Technical University of Denmark, Palm, p 25

Park K, Ali A, Kim D, An Y, Kim M, Shin H (2013) Robust predictive model for evaluating breast cancer survivability. Eng Appl Artif Intell 26:2194–2205

Pena-Reyes CA, Sipper M (1999) A fuzzy-genetic approach to breast cancer diagnosis. Artif Intell Med 17:131–155

Pham D, Sagiroglu S (2000) Neural network classification of defects in veneer boards. Proc Inst Mech Eng Part B J Eng Manuf 214:255–258

Polat K, Güneş S (2007) Breast cancer diagnosis using least square support vector machine. Digit Signal Proc 17:694–701

Qu B, Lang B, Liang J, Qin A, Crisalle O (2016) Two-hidden-layer extreme learning machine for regression and classification. Neurocomputing 175:826–834

Quinlan JR (1996) Improved use of continuous attributes in C4.5. J Artif Intell Res 4:77–90

Razavi SH, Ebadati EOM, Asadi S, Kaur H (2015) An efficient grouping genetic algorithm for data clustering and big data analysis. Computational intelligence for big data analysis. Publishing, pp 119–142

Saritas I, Ozkan IA, Sert IU (2010) Prognosis of prostate cancer by artificial neural networks. Expert Syst Appl 37:6646–6650

Shahrabi J, Hadavandi E, Asadi S (2013) Developing a hybrid intelligent model for forecasting problems: case study of tourism demand time series. Knowl Based Syst 43:112–122

Shen F, Chao J, Zhao J (2015) Forecasting exchange rate using deep belief networks and conjugate gradient method. Neurocomputing 167:243–253

Smolensky P (1986) Information processing in dynamical systems: Foundations of harmony theory. Parallel Distributed Processing: Volume 1: Foundations. MIT Press, Cambridge 1987:194–281

Sumbaly R, Vishnusri N, Jeyalatha S (2014) Diagnosis of breast cancer using decision tree data mining technique. Int J Comput Appl 98:16–24

Tahan MH, Asadi S (2018a) EMDID: evolutionary multi-objective discretization for imbalanced datasets. Inf Sci 432:442–461

Tahan MH, Asadi S (2018b) MEMOD: a novel multivariate evolutionary multi-objective discretization. Soft Comput 22:301–323

Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th international conference on machine learning. Publishing, pp 1064–1071

Tieleman T, Hinton G (2009) Using fast weights to improve persistent contrastive divergence. In: Proceedings of the 26th annual international conference on machine learning. Publishing, pp 1033–1040

Übeyli ED (2007) Implementing automated diagnostic systems for breast cancer detection. Expert Syst Appl 33:1054–1062

Wang Y, Xie Z, Xu K, Dou Y, Lei Y (2016) An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning. Neurocomputing 174:988–998

Wu Y, Wu Y, Wang J, Yan Z, Qu L, Xiang B, Zhang Y (2011) An optimal tumor marker group-coupled artificial neural network for diagnosis of lung cancer. Expert Syst Appl 38:11329–11334

Xue B, Zhang M, Browne WN (2014) Particle swarm optimisation for feature selection in classification: novel initialisation and updating mechanisms. Appl Soft Comput 18:261–276

Yu W, Zhuang F, He Q, Shi Z (2015) Learning deep representations via extreme learning machines. Neurocomputing 149:308–315

Zheng B, Yoon SW, Lam SS (2014) Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms. Expert Syst Appl 41:1476–1482

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.