Breast Cancer Diagnostic Tool Using Deep Feedforward Neural Network and Mother Tree Optimization

Wael Korani[1] and Malek Mouhoub[2](✉)

[1] University of Saskatchewan, Saskatoon, SK, Canada
wak137@usask.ca

[2] University of Regina, Regina, SK, Canada
mouhoubm@uregina.ca

Abstract. Automatic diagnostic tools have been extensively implemented in medical diagnosis processes of different diseases. In this regard, breast cancer diagnosis is particularly important as it becomes one of the most dangerous diseases for women. Consequently, regular and preemptive screening for breast cancer could help initiate treatment earlier and more effectively. In this regard, hospitals and clinics are in need to a robust diagnostic tool that could provide reliable results. The accuracy of diagnostic tools is an important factor that should be taken into consideration when designing a new system. This has motivated us to develop an automatic diagnostic system combining two methodologies: Deep Feedforward Neural Networks (DFNNs) and swarm intelligence algorithms. Swarm intelligence techniques are based on Particle Swarm Optimization (PSO) as well as the Mother Tree Optimization (MTO) algorithm we proposed in the past. In order to asses the performance, in terms of accuracy, of the proposed system, we have conducted several experiments using the Wisconsin Breast Cancer Dataset (WBCD). The results show that the DFNN combined with a variant of our MTO attains a high classification performance, reaching 100% precision.

Keywords: Neural network · Nature-inspired techniques · Classification · Breast cancer diagnosis

1 Introduction

Cancer represents the second highest cause of death in the world [1]. This disease affects several vital human organs: pancreas, liver, testis, prostate, lung, cervix uteri, melanoma of skin, breast, and so forth. Breast cancer is the second most common diagnosed cancer [2]. Early detection of breast cancer gives doctors and decision makers the opportunity to initiate an effective treatment method. There are several kinds of cancer classifications, but the most important is the binary one: benign or malignant. The benign stage of breast cancer is less invasive and

© Springer Nature Switzerland AG 2020

does not have high risk treatment, while the malignant stage may cause severe health complications [3].

Automatic diagnostic classifiers have been used to diagnose breast cancer at an early stage. These classifiers provide radiologists with more confidence to support their breast cancer diagnosis. These tools should be carefully designed with high accuracy so that they can provide reliable results. In this regard, the WBCD has been used for classification models evaluation purposes [4], and the models that we present below, achieved varying results with relatively good accuracy. Quinlan et al. introduced an improved variant of the C4.5 algorithm, and the authors evaluated the classifier on different datasets. The results showed that the accuracy of the proposed algorithm is 94.74% with tree size 25 ± 0.5 [5]. Hamilton et al. introduced a classifier called Rule Induction through Approximate Classification (RIAC), and the algorithm was evaluated on different datasets. RIAC achieved 94.99% accuracy compared to 96.0% accuracy when using C4.5 [6]. Salama conducted extensive experiments to compare different classifiers. The results showed that the best classifier is Sequential Minimal Optimization (SMO) with accuracy 97.72% [7]. Polat et al. conducted an analysis of the Least Square Support Vector Machines (LS-SVM) classifier using the WBCD. The results showed that the accuracy of the classifier is 94.44% using 80% of data as training and 20% as test data [8]. Nauck et al. introduced a neuro-fuzzy classifier, combining neural networks and fuzzy logic and called NEFCLASS, that achieved 95.04% accuracy [9]. Pena-Reyes et al. introduced a fuzzy-genetic classifier that combines genetic algorithms along with a fuzzy system. The authors evaluated the fuzzy-genetic system and achieved 97.8% accuracy [10]. Abonyi et al. introduced a fuzzy system that is an extension of the quadratic Bayes classifier. The proposed classifier allows each rule to represent more than one class and achieve 95.57% accuracy [11]. Paulin et al. conducted an extensive comparison between several back propagation algorithms to tune the DFNN. The results show that Levenberg Marquardt (LM) is the best algorithm with 99.28% accuracy [12]. Nahato et al. introduced a classifier using a Relation Method With A Back Propagation Neural Network (RS-BPNN). RS-BPNN has been compared with several published models, and obtained an accuracy of 98.6%, which outperforms all the other classifiers [13]. Abdel-Zahe et al. introduced the back-propagation neural network with Liebenberg Marquardt learning function. Here, the weights are initialized from the Deep Belief Network path (DBN-NN). The authors evaluated the accuracy of the classifier and obtained a score of 99.68%, which outperforms the other classifiers from the literature [14].

Despite the success results we listed above, training a neural network using back propagation [15] has some limitations. The back propagation algorithm requires some learning parameters such as, learning rate, momentum, gradient information, and predetermined structure. In addition, this algorithm assumes that the DFNN has a fixed structure; therefore, designing a near optimal DFNNs structure is still unsolved problem [16]. To overcome these limitations, several studies have used nature-inspired techniques for training, instead of back propagation. These techniques achieved better performance as they have a better ability to reach the global optimum, in practice (by preventing the search algorithm from being trapped in a local optimum). In this regard, the Artificial Bee Colony (ABC) and PSO algorithms have been used to adjust weights of DFNNs and achieved promising results [17,18].

Following the same idea, we propose a new system using a DFNN together with a swarm intelligent technique. In this regard, we consider PSO as well as new nature-inspired technique that we recently proposed and called MTO [19]. In order to asses the performance, in terms of accuracy, of the proposed system, we have conducted several experiments using the Wisconsin Breast Cancer Dataset (WBCD). The results show that the DFNN combined with a variant of our MTO, called MTO with Climate Change (MTOCL), attains a high classification performance, reaching 100% accuracy.

2 Dataset Description

The WBCD is produced by the University of Wisconsin hospital for diagnosing breast cancer based on the Fine Needle Aspiration (FNA) test [4]. This dataset has been used to evaluate the effectiveness of classifier systems. It is used to distinguish between benign and malignant cancers based on nine attributes from the FNA: Clump thickness, Uniformity of cell size, Uniformity of cell shape, Marginal Adhesion, Single Epithelial cell size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses as shown in Table 1. Attributes have integer values in range [1, 10]. These attributes play a significant role in determining whether a cell is cancerous or not. For example, thickness does not get grouped cancerous cells that are grouped in multilayers while benign cells are grouped in monolayers affecting clump thickness. The uniformity of size and shape play an important role as well to distinguish between cancerous cells and normal cells. In addition, normal cells have the ability to stick together while cancerous cells lose this feature. Epithelial cell size is one of the indicators of malignancy as well. The nuclei that are not surrounded by cytoplasm are called bare nuclei, which occurs in benign tumors. The bland chromatin is uniform in benign cells while it is coarser in malignant cells. Finally, pathologists can determine the grade of cancer through the number of mitoses [7].

The data set contains 699 case studies that are divided into: benign 458 (65.5%) and malignant 241 (34.5%). In addition, we removed all missing values (16 cases) during the experiments, given that their count is low. Removing data with missing values results in robust and highly accurate model. Each case study has 11 attributes including class label as shown in Table 1 and sample id number. We removed the sample id attribute during the experiment, the class attribute represents the output class, and the rest of attributes are the inputs. Figure 1 shows the distribution and frequency of all nine features that have values in range [1: 10] as shown in Table 1, and the attribute class has either 2 or 4. The distribution demonstrates that the values of each feature are well scattered in the 2D map.

Table 1. List of attributes including the binary classification

No.	Attribute	Domain
1	Sample number	
2	Clump thickness	1–10
3	Uniformity of cell size	1–10
4	Uniformity of cell shape	1–10
5	Marginal adhesion	1–10
6	Single epithelial cell size	1–10
7	Bare nuclei	1–10
8	Bland chromatin	1–10
9	Normal nucleoli	1–10
10	Mitoses	1–10
11	Class	2 for benign and 4 for malignant

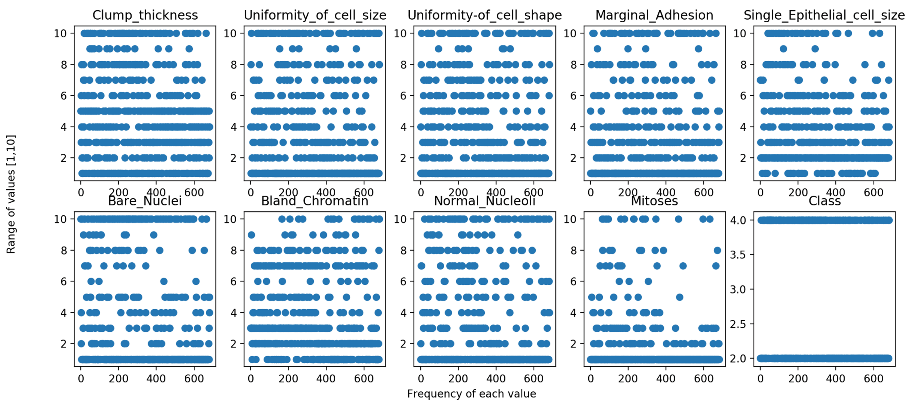

Fig. 1. Wisconsin breast cancer data: nine attributes (input) and class attribute (output)

2.1 Data Processing

Data normalization is one of the approaches that is used to obtain better results and minimize bias. In addition, data normalization can speed up the training process by starting the training process for any given feature within the same scale. The normalization process produces the same range of values for each input feature for the neural network. In our experiment, all input features are normalized in the range between 0 and 1.

Discretizing data is another significant process that makes the prediction process more effective. Desensitization is used to convert numerical values into categorical values. One way to discretize is dividing the entire range of values into a number of groups. In our experiment, the output class has values 2 or 4, which can be represented as 00100 or 00001. In addition, all missing data are removed from the dataset. The dataset is then divided into training dataset (80%) (559 samples) (60% training and 20% validating) and testing dataset (20%) (140 samples). The training dataset is applied to create the model, and testing dataset is used to evaluate the accuracy of the model.

3 Proposed Breast Cancer Diagnosis System

The architecture of our proposed system is summarized in Fig. 2. Here, the chosen technique (MTO, MTOCL or PSO) initially generates a random population (20 agents) using the parameters in Table 2, and each agent of the population represents all necessary weights for all layers: the weights between the input layer and the first hidden layer is $(9 * 30)$ weights, between the first hidden layer and the second hidden layer is $(30 * 15)$ weights, and between the second hidden layer and the output layer is $(15 * 5)$ weights, which totals to 795 weight. These weights, in range $[-4, 4]$, feed our DFNN to create our model. Thus, the initial generated agents produce 20 different DFNN models, and the mean square error (MSE) of each model is calculated as an indicator of its performance. The MSE is then returned back to the selected swarm intelligence algorithm, which represents the fitness value of each agent in the population. The swarm algorithm then updates the position of each agent (795 weights of DFNN) using different rules to improve its MSE (minimize its value). This process is repeated until the system achieves the minimum MSE values [17,18]. The number of iterations is set to 100 for PSO or MTO, and 20 for MTOCL, for 5 climate change (100 iterations in total).

Table 2. Parameter settings

Algorithm	Parameters settings
MTO or MTO-CL	Root signal $\delta = 2.0$
	MFN signal $\Delta = 0.03$
	Small deviation $\phi = 1.0$
	Population size = 20
PSO	Constriction factor $\chi = 0.72984$
	Acceleration coefficient $c_1 = 2.02$ and $c_1 = 2.02$
	Population size = 20

Our DFNN consists of one input layer (9 inputs), one output layer (5 digits), and two hidden layers respectively containing 30 and 15 neurons. We determine the number of neurons in each hidden layer after conducing several preliminary experiments. The input layer has a number of neurons that is equal to the number of input attributes (9) and the output layer has a number of neurons that is equal to the number of output classes (5) (after categorizing the output class to 00100 and 00001). In our network, the sigmoid function has been used as an activation function and mean square error as a fitness value. The network is connected and tuned separately with each of the three swarm intelligence algorithms through minimizing the mean square error until the input-output mapping is complete. We use the accuracy (number of predictions over by the number of samples) as an indicator to measure the performance of each proposed model.

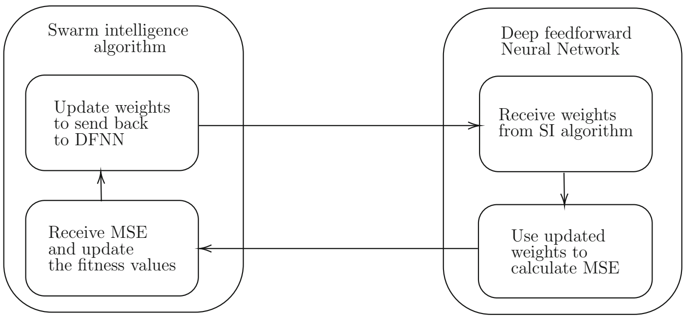
Swarm intelligence algorithm
Update weights to send back to DFNN
Receive MSE and update the fitness values
Deep feedforward Neural Network
Receive weights from SI algorithm
Use updated weights to calculate MSE

Fig. 2. The proposed system of training the feedforward neural network using SI algorithm

3.1 Mother Tree Optimization (MTO)

MTO [19] pseudocode is shown in Algorithm 1. MTO uses a set of cooperating agents that evolve in a fashion inspired by communication between Douglas fir trees, and mediated by the mycorrhizal fungi network that transfers nutrients between plants of the same or different species. The population is a group of Active Food Sources (AFSs) whose size is denoted by $\mathrm{N_T}$. Agents in the population are arranged in descending order based on their fitness values. Agents performs feeding and receiving operations in each iteration. During feeding operation, the population is partitioned into feeders and non-feeders. Each member in the feeder group can feed an offspring N_{os}. The number of agents in feeder N_{Frs} and non-feeder N_{NFrs} groups are given by

$$\begin{aligned} N_{os} &= \frac{N_T}{2} - 1, \\ N_T &= N_{Frs} + N_{NFrs} \\ N_{Frs} &= \frac{N_T}{2} + 1 \end{aligned} \tag{1}$$

During the receiving operations, the population is divided into four different groups. Agents update their positions according to the group to which they belong. The population is partitioned into a single Top Mother Tree (TMT) (an agent receiving nutrients from a random source), a set of Partially Connected Trees (PCTs) that has $\mathrm{N_{PCTs}}$ agents, and Fully Connected Trees (FCTs) group that has $\mathrm{N_{FCTs}}$ agents. The numbers of agents in the PCTs and FCTs groups are given by

$$\begin{gathered} N_{PCTs} = N_T - 4, \\ N_{FCTs} = 3, \\ N_T = N_{FCTs} + N_{PCTs} + 1. \end{gathered} \tag{2}$$

The TMT performs an exploitation process at each iteration (using two levels of exploitation) by searching for better solution around its position. The updated position of the first level of exploitation of the TMT is given by:

$$P_1(x_{k+1}) = P_1(x_k) + \delta R(d) \tag{3}$$

$$R(d) = 2\ (round\ (rand(d,1)) - 1)\ rand(d,1), \tag{4}$$

where root signal step size δ is equal to 2.0 that has been adopted based on preliminary experiments, $R(d)$ is a random fixed vector that depends on the seed number, $P(x_k)$ is the position of an agent at iteration k. After each iteration of the first exploitation level, it compares the fitness value of the new position with the current one. If the new position has a higher fitness value than the current one, then it will move to the next exploitation level; otherwise it does not. In each iteration of the second level, the TMT evaluates a new position in a random direction with smaller step size $\varDelta$. The value of $\varDelta$ is set to 0.03 after preliminary experiments.

$$P_1(x_{k+1}) = P_1(x_k) + \varDelta R(d) \tag{5}$$

where $\varDelta$ is the MFN signal step size. The user may tune the values of δ and $\varDelta$ depending on the optimization problem. The FPCTs group has $(\frac{\mathrm{N_T}}{2} - 2)$ agents and starts from the agent ranked 2 and ends at the agent ranked $(\frac{\mathrm{N_T}}{2} - 1)$. **The members of FPCTs** group update the position as follows:

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=1}^{n-1} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)), \tag{6}$$

where $P_n(x_k)$ represents the current position of any member in range $[2, \frac{\mathrm{N_T}}{2} - 1)]$, $P_i(x_k)$ represents the current position of a candidate solution that has better number of nutrients, and $P_n(x_{k+1})$ represents the updated position of this member. **The members of the FCTs** group start at candidate solution ranked $\frac{\mathrm{N_T}}{2}$ to candidate solution ranked $\frac{\mathrm{N_T}}{2} + 2$. The updated position is given by

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=n-N_{os}}^{n-1} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)). \tag{7}$$

The LPCTs group members start at candidate solution ranked $\frac{N_T}{2} + 3$ to the end of the population. The updated position is given by

$$P_n(x_{k+1}) = P_n(x_k) + \sum_{i=n-N_{os}}^{N_T-N_{os}} \frac{1}{n-i+1}(P_i(x_k) - P_n(x_k)). \tag{8}$$

3.2 MTO Algorithm with Climate Change (MTOCL)

The MTOCL extends MTO with two phases: elimination and distortion as shown in Algorithm 1. In the elimination phase, the candidate solutions that have the lowest fitness are removed and are replaced by new random candidate solutions in the search space. In our experiments, the elimination percentage has been adopted based on preliminary experiments to be 20% of the total population. In the distortion phase, the rest of the population (80%) is distorted by slightly deviating their candidate solutions positions.

The MTO algorithm and its variant MTOCL have been tested on several recommended optimization benchmark functions. The results showed that MTO algorithm achieves better performance in terms of solution quality and number of function evaluations compared to several PSO variants. In addition, the results showed that MTOCL has the capability to solve more complex problems that MTO could not solve [19].

3.3 Particle Swarm Optimization (PSO)

In 1995, Eberhart and Kennedy introduced the first idea of particle swarm optimization as shown in Algorithm 2 [20]. The PSO algorithm mimics the movement of a flock of birds. Each bird in the flock is associated to a particle (candidate solution). The position of each particle in the search space is updated based on the previous best position of the particle itself (local position) and the best position of the entire flock (global position). The PSO algorithm updates the position of each particle using the following equation [20]:

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}, \tag{9}$$

where x_{id} is the position of a particle i, the superscript k denotes the iteration rank, and v_{id} is the velocity of the particle i. The velocity of the particle i is updated using the following equation:

$$v_{id}^{k+1} = \chi(v_{id}^{k} + c_1 \times r_1[P_{id}^{k} - x_{id}^{k}] + c_2 \times r_2[P_{gd}^{k} - x_{id}^{k}]), \tag{10}$$

where χ is the constriction factor, the v_{id}^{k} is the previous velocity of the particle i that provides the necessary momentum for moving around the search space. The constants c_1 and c_2 are also known as the acceleration coefficients, and r_1 and r_2 are uniform distribution random numbers in range [0, 1]. P_{id}^{k} is the local best position for the particle i at iteration k, and P_{gd}^{k} is the global best

Algorithm 1. The MTO algorithm and MTOCL variant

```
Require: : $N_T, P_T, d, K_{rs}, Cl,$ and $El$
    $N_T$: The population size (AFSs)
    $P_T$: The position of the active food sources
    $d$: The dimension of the problem
    $K_{rs}$: The number of kin recognition signals
    $Cl$: The number of climate change events (0 for MTO)
    $El$: The elimination percentage
    Distribute $T$ agents uniformly over the search space $(P_1, \dots, P_T)$
    Evaluate the fitness value of $T$ agents $(S_1, \dots, S_T)$
    Sort solutions in descending order based on the fitness value and store them in $S$
    $S = Sort(S_1, \dots, S_T)$
    The sorted positions with the same rank of $S$ stored in array $A$
    $A = (P_1, \dots, P_T)$
    loop
        for $k_{rs} = 1$ to $K_{rs}$ do
            Use equations (3)–(8) to update the position of each agent in $A$
            Evaluate the fitness of the updated positions
            Sort solutions in descending order and store them in $S$
            Update $A$
        end for
        if $Cl = 0$ then
            $BREAK$;
        else
            Select the best agents in $S$ ((1 - El) S)
            Store the best selected position in $Abest$
            Distort $Abest$ (mulitply by random vector)
            $Distort(Abest) = Abest \ * R(d)$
            Remove the rest of the population $(El)S$
            Generate random agents equal to the the number of removed agents
            $Cl = Cl - 1$
        end if
    end loop $(Cl > 0)$
    $S = Sort(S_1 \dots S_T)$
    Global Solution = Min(S)
    return Global Solution
```

position at iteration k. The vector toward the local best position for each particle is calculated by $[P_{id}^k - x_{id}^k]$, and it is known as the "cognitive" component. The vector toward the global best position for each particle is calculated by $[P_{gd}^k - x_{id}^k]$, and it is known as the "social" component. The social component represents the collaborative effect of the particles to find the global solution, and it helps other particles toward the global best particle found so far.

Algorithm 2. Particle Swarm Optimization

```
Generate random particles and random velocities.
while Stopping condition is not satisfied do
    for particle $i$= 1 to population size ($n$) do
        Update the velocity using Eq. 10
        Update the position using Eq. 9
        Evaluate the fitness value $f$
        if current fitness value ($f_i$) < local best fitness ($f_{lbest}$) then
            $f_{lbest} = f_i$
        end if
        if current fitness value ($f_i$) < global best fitness ($f_{gbest}$) then
            $f_{gbest} = f_i$
        end if
    end for
end while
```

4 Experimentation

The following steps have been carried out to implement and evaluate the system with each of the following swarm intelligence techniques: MTO and MTOCL [19] in addition to PSO [20]. Firstly, data is cleaned (missing data is removed), and input data is normalized and output data is discretized. In our experiment, data is divided into 80% training (60% training and 20% validating) and 20% testing. The DFNN then is created and connected with early selected swarm intelligence algorithm. In the first iteration, DFNN weights are initialized randomly, and DFNN use the weights to calculate the Root Mean Square RMS error as follows:

$$RMS = \sqrt{(Y_{actual} - Y_{predicted})^2}, \tag{11}$$

The selected swarm intelligence algorithm receives RMS error back from the DFNN and updates the position based on different rules. The network is trained using each of the three algorithms. The network is tested using testing data. The accuracy of the network has been calculated for each network.

4.1 Performance Evaluation

The confusion matrix is a tool that is used to calculate the precision, recall, and F1 score of a classifier. The precision is used to evaluate the relationships between true positive and total predicted positive values, which is usually used when the costs of false positive is high (in our case malignant). The recall test measures the relationship between true positive and total actual positive values. Finally, the F1 Score measures the balance between precision and recall. It is important to highlight that accuracy can be largely contributed by a large number number of true negative, which does not have much weight, but false positive has much business cost.

Table 3 shows the results of few tests for each of the three swarm intelligent algorithms. The results demonstrate the effect of each of the algorithms for training the DFNN on the precision, recall, and F1 score tests of the model. Here, MTO achieves better results in training the DFNN than MTOCL and PSO. Indeed, the calculated precision, recall, and F1 score are 100% for MTO and 97.9% for MTOCL. However, PSO achieves the worst results.

Table 3. The results of precision, recall, and F1 score tests

Test	SI algorithm	Result %
Precision	PSO	95.9
	MTO	100
	MTOCL	97.9
Recall	PSO	97.9
	MTO	100
	MTOCL	97.9
F1 score	PSO	96.9
	MTO	100
	MTOCL	97.9

5 Conclusion

We propose a high accuracy automatic diagnostic system using DFNN and three swarm intelligence algorithms: PSO, MTO, and MTOCL. In order to assess the performance of our system, we conducted several experiments on the WBCD dataset. The results are very promising as our system is able to reach a 100% precision, when using the MTO technique. We anticipate that our system can produce reliable results for hospitals and clinics, allowing patients to receive an instant diagnosis for breast cancer after performing the FNA test.

References

1. Brenner, H., Rothenbacher, D., Arndt, V.: Epidemiology of stomach cancer. In: Verma, M. (ed.) Cancer Epidemiology, pp. 467–477. Springer, Heidelberg (2009). https://doi.org/10.1007/978-1-60327-492-0_23
2. Parkin, D.M., Bray, F., Ferlay, J., Pisani, P.: Estimating the world cancer burden: Globocan 2000. Int. J. Cancer **94**(2), 153–156 (2001)
3. Rangayyan, R.M., El-Faramawy, N.M., Desautels, J.L., Alim, O.A.: Measures of acutance and shape for classification of breast tumors. IEEE Trans. Med. Imaging **16**(6), 799–810 (1997)

4. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. Technical report, University of Wisconsin-Madison Department of Computer Sciences (1990)
5. Quinlan, J.R.: Improved use of continuous attributes in C4.5. J. Artif. Intell. Res. **4**, 77–90 (1996)
6. Hamilton, H.J., Cercone, N., Shan, N.: RIAC: A Rule Induction Algorithm Based on Approximate Classification. Princeton, Citeseer (1996)
7. Salama, G.I., Abdelhalim, M., Zeid, M.A.: Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC) **32**(569), 2 (2012)
8. Polat, K., Güneş, S.: Breast cancer diagnosis using least square support vector machine. Digit. Signal Proc. **17**(4), 694–701 (2007)
9. Nauck, D., Kruse, R.: Obtaining interpretable fuzzy classification rules from medical data. Artif. Intell. Med. **16**(2), 149–169 (1999)
10. Pena-Reyes, C.A., Sipper, M.: A fuzzy-genetic approach to breast cancer diagnosis. Artif. Intell. Med. **17**(2), 131–155 (1999)
11. Abonyi, J., Szeifert, F.: Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recogn. Lett. **24**(14), 2195–2207 (2003)
12. Paulin, F., Santhakumaran, A.: Classification of breast cancer by comparing back propagation training algorithms. Int. J. Comput. Sci. Eng. **3**(1), 327–332 (2011)
13. Nahato, K.B., Harichandran, K.N., Arputharaj, K.: Knowledge mining from clinical datasets using rough sets and backpropagation neural network. Comput. Math. Methods Med. **2015**, 13 (2015)
14. Abdel-Zaher, A.M., Eldeib, A.M.: Breast cancer classification using deep belief networks. Expert Syst. Appl. **46**, 139–144 (2016)
15. Werbos, P.J.: The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting, vol. 1. Wiley, Hoboken (1994)
16. Hush, D.R., Horne, B.G.: Progress in supervised neural networks. IEEE Signal Process. Mag. **10**(1), 8–39 (1993)
17. Ozturk, C., Karaboga, D.: Hybrid artificial bee colony algorithm for neural network training. In: 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 84–88. IEEE (2011)
18. Karaboga, D., Akay, B., Ozturk, C.: Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 318–329. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73729-2_30
19. Korani, W., Mouhoub, M., Spirty, R.: Mother tree optimization. In: Proceedings of the 2019 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2019), pp. 2206–2213. IEEE (2019)
20. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, MHS 1995, pp. 39–43. IEEE (1995)