Transactions on Machine Intelligence 5(4) (2022) 256-264

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Detection of Benign and Malignant Breast Cancer Using Data Shuffling Ensemble Method

M.R. Rahmani Dahaghi[1], S. M. Hashemi[2], Z. Assarzadeh[3,*]

[1] M.Sc. Student, Department of Electrical Engineering, Payam Noor Higher Education Institute, Golpayegan, Iran
[2] Faculty Member, Department of Electrical Engineering, Payam Noor Higher Education Institute, Golpayegan, Iran
[3] Faculty Member, Payam Noor Higher Education Institute, Golpayegan, Iran

ARTICLE INFO

Article History:
Received 5 May 2022
Received in revised form 29 August 2022
Accepted 20 December 2022
Available online 21 December 2022

Keywords:
Artificial Neural Network,
Classification, Backpropagation,
Data-Shuffling Ensemble Method.

ABSTRACT

Breast cancer remains one of the most serious health concerns affecting women worldwide. Early detection of malignant tumors significantly improves survival rates and enhances patients' quality of life. As a result, the development of accurate and efficient diagnostic systems for breast cancer is of critical importance. Artificial neural networks (ANNs) have been widely applied in medical diagnosis, particularly in data classification tasks. The backpropagation algorithm is a commonly used technique for training neural networks; however, it suffers from certain limitations, such as slow convergence, susceptibility to local minima, and sensitivity to initial weight selection. To address these challenges, this study employs a data-shuffling ensemble method to improve classification accuracy. The effectiveness of both approaches is assessed in distinguishing between benign and malignant breast tumors using a well-established dataset. Experimental results indicate that the neural network utilizing the data-shuffling ensemble method achieves an accuracy of 99.3%, outperforming the backpropagation algorithm. These findings highlight the potential of ensemble learning techniques in enhancing diagnostic accuracy and reliability in medical applications. The study contributes to the ongoing advancements in artificial intelligence-based diagnostic tools, emphasizing the importance of robust classification techniques in improving breast cancer detection and patient outcomes.

1. INTRODUCTION

In 2013, the American Cancer Society predicted approximately 40,030 deaths due to cancer in the United States (39,620 women and 410 men). Breast cancer refers to the abnormal proliferation of cells in breast tissue and is the second leading cause of death in women after lung cancer. Since 1989, the mortality rate from breast cancer in women under the age of fifty has significantly decreased. This reduction in mortality rates can be attributed to early detection and effective treatment [1].

* Corresponding Author: assar3336@gmail.com
Faculty Member, Payam Golpayegan Higher Education Institute, Golpayegan, Iran

http://dx.doi.org/10.47176/TMI.2022.256

Breast cancer is one of the most common types of cancer and is treatable in its early stages. The treatment and causes of breast cancer are still under investigation, and since no preventive methods are currently available, early identification and diagnosis of this disease are crucial for a full recovery. In fact, the first step toward treatment is diagnosis, which is why research aimed at finding methods for the early detection of this disease has been conducted using data from patients at the University of Wisconsin School of Medicine [2].

In the field of machine learning, many methods have been proposed for breast cancer classification. This paper discusses two methods: the backpropagation algorithm and the data-shuffling ensemble method in detail. The structure of this paper is as follows: Section 2 reviews previous works on the use of machine learning algorithms for breast cancer detection. Section 3 provides an overview of the backpropagation algorithm and the data-shuffling ensemble method. Then, Section 4 compares the efficiency of the aforementioned methods, and Section 5 concludes the paper.

2. REVIEW OF RELATED WORKS

The use of machine learning techniques in medical diagnosis is gradually increasing. Data mining and soft computing techniques have been employed to extract rules and patterns from various datasets [3, 4].

Several of these techniques have demonstrated excellent results in classification problems, aiding medical professionals in disease identification [5–7]. Hamilton et al. achieved 96% accuracy using the RIAC algorithm [8]. In this method, inference rules are derived from sample-based theory using approximate classification.

Kouin Lan achieved 94.74% accuracy using the C4.5 decision tree algorithm with 10-fold cross-validation [9]. C4.5 is a system that learns decision tree classification, using a divide-and-conquer approach to develop decision trees, pioneered by Han, Marin, and Weston in 1996.

Pena-Ries achieved 36.97% accuracy using the fuzzy-GA method [10]. This method combines fuzzy logic and evolutionary algorithms to provide automated diagnostic systems. It is based on two features: high classification performance and results that include simple and interpretable rules.

John Aboni achieved 95.57% accuracy using a supervised fuzzy clustering technique [11]. This classifier can be considered an extension of the second-degree Bayesian classifier. The input variables for the fuzzy clustering technique are selected based on Fisher's discriminant analysis within the chosen class.

Pauline used a feedforward neural network and backpropagation algorithm for neural network training [12]. The network's performance was evaluated on the Wisconsin breast cancer dataset, and the highest accuracy of 99.28% was achieved using the Levenberg–Marquardt algorithm. Lavania achieved 92.97% classification accuracy using a decision tree algorithm [13]. This paper analyzes the performance of the decision tree classifier (CART) with and without feature selection.

Arpit Bihardwaj proposed the GONN algorithm, achieving accuracy rates of 98.24% (training data) and 97.73% (test data) with a 50/50 split [14]. This paper introduces a novel method known as the genetic-optimized neural network to solve classification problems, incorporating new cut and mutation operators that differ from standard ones. GhayoumiZadeh et al. (2016) introduced a method for segmenting cancerous regions in thermal images using fuzzy active contours, which enhanced the accuracy of tumor localization [15].

Seryasat and Haddadnia (2018) further contributed to this field by evaluating an ensemble learning framework for mass classification in mammograms, significantly improving classification performance [16]. Additionally, Rahmani-Seryasat et al. (2015) proposed a novel approach for classifying breast cancer tumors and analyzing their fractionation, contributing to more precise diagnostic methods [17]. These works collectively highlight the impact of advanced image processing techniques in the early detection and accurate classification of breast cancer.

Rahmani Seryasat et al. also investigated ensemble regression techniques in social media contexts, successfully predicting Facebook post comment volumes with a stacked ensemble model [18].

3. MACHINE LEARNING ALGORITHMS

This section briefly describes the backpropagation algorithm and the data-shuffling ensemble method.

3.1. Backpropagation Algorithm

The feedforward neural network is inspired by biological neural networks and consists of a large number of units, called artificial neurons, which are connected by weighted links. These units are arranged in layers, including an input layer, one or more hidden layers, and an output layer. Let k represent the total number of layers in a network, and n_l represent the total number of units in the l-th layer, where $0 \leq l \leq k$. Layer 0 is the input layer, layer k is the output layer, and layers between 0 and k are hidden layers. The architecture of a feedforward neural network is shown in Figure 1.

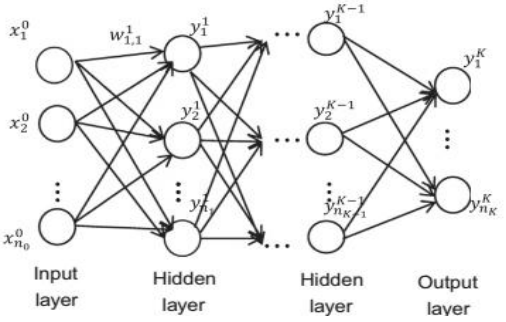


Fig. 1. Architecture of a Feedforward Neural Network

$$x_j^l = \sum_i w_{ji}^l y_j^{L-1}(K) \tag{1}$$

where w_{ji}^l is the weight connecting unit i in layer $l-1$ to unit j in layer l. The activation value of unit j in layer l is given by:

$$y_j^l = f(x_j^l - \theta_j^l) \tag{2}$$

where *f* is the activation function, which is generally continuous and differentiable. θ_j^l is the potential value of unit *j* in layer l, which is represented by the weights from the neurons with a fixed output value of 1. Figure 2 illustrates the relationship between the input and output of unit *j* in layer l.

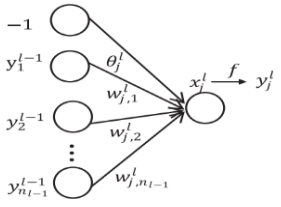


Fig. 2. The relationship between the input and output of unit j in layer l.

y_1^{l-1}، y_2^{l-2} ،.....، y_{nl-1}^{l-1} represent the outputs of units in layer $l-1$. The output of a bias neuron is 1. This diagram visually demonstrates how θ_j^l is treated as a weight that is adjusted during the training process. In other words, we update and refine the potential values of the units with the weight parameters. With this in mind, we can apply weight update rules to train the potential value of unit θ_j^l. If the output pattern differs from the target output, an error occurs, and then backpropagation or reverse propagation through the network is performed, from the output layer to the input layer. The weights are adjusted according to the propagated error, and the network's output is corrected to be closer to the desired output. Both forward and backward passes are performed sequentially, and the error between the output pattern and the target pattern is minimized. The error is typically represented as:

$$\boldsymbol{E_0 = \frac{1}{2} \sum_{d=1}^{D} \sum_{j=1}^{n_k} (y_{jd}^{k} - o_j^{d})^2} \tag{3}$$

where o_j is the target output of unit *j*, y_j^k is the actual output of unit j in the output layer, and *D* represents the number of input patterns. In the implementation of the backpropagation algorithm, we can apply an online learning strategy to update the network's weights.

Thus, we only need to take the derivative of the error for a single input pattern $\boldsymbol{E_{01} = \frac{1}{2} \sum_{j=1}^{n_k} (y_j^k - o_j)^2}$ with respect to the network parameters. For the output layer, we have:

$$w_{ji}^{k}(t+1) = w_{ji}^{k}(t) + \eta \left(-\nabla_{w_{ji}^{k}} E_{01}\right) = w_{ji}^{k}(t) + \eta \frac{dy_j^k}{dx_j^k} (y_j^k - o_j) y_i^{k-1} = w_{ji}^{k}(t) + \eta \delta_j^k y_j^{k-1} \tag{4}$$

where δ_j^k is defined as:

$$\boldsymbol{\frac{dy_j^k}{dx_j^k} (y_j^k - o_j)} \tag{5}$$

For the *l* hidden layers, where $0 < l < k$, we have:

$$w_{ji}^{l}(t+1) = w_{ji}^{l}(t) + \eta \delta_j^l y_i^{l-1} \tag{6}$$

With

$$\delta_j^l = \frac{dy_j^l}{dx_j^l} \sum_s \delta_s^{l+1} w_{sj}^{l+1} \tag{7}$$

Based on the chain rule of derivatives, calculating the derivative of E_{01} with respect to the weight w_{ji}^l includes the derivative of y_j^l with respect to x_j^l. Note that the derivative of y_j^l with respect to x_j^l depends on the activation function used in the neural network. Therefore, as general formulas for any activation functions, equations (4)–(7) are applied. The term $\frac{dy_j^l}{dx_j^l}$refers to the derivative of y_j^l with respect to x_j^l, and specific formulas are not provided. In practical applications, we will compute $\frac{dy_j^l}{dx_j^l}$based on the specific class of the activation function and then derive E_{01} with respect to the network's weights.

In the standard backpropagation learning algorithm, training examples are fed into the network in sequence, and the weights are updated for each training example using equations (4) and (6).

3.2. GMDH Neural Network

GMDH-type neural networks [19,20] are automatically formed using the self-organizing exploratory method [21]. In GMDH-type neural networks, structural parameters such as the number of layers, the number of neurons in each layer, usable input variables, and the desired neuron architectures are automatically and autonomously determined.

In contrast to feedforward neural networks, GMDH-type neural networks do not require prearranging or setting up their structure and can be described by a brief polynomial set [22-24]. GMDH networks are multilayered, with each layer containing neurons with a transfer function g, which is a short polynomial. For example, a linear polynomial is:

$$y = g(u_1, u_2) := w_0 + w_1 u_1 + w_2 u_2 \tag{1}$$

where u_1 and u_2 are the variables, and w_0 ، w_1 ، w_2 are the weights or coefficients of the polynomial.

3.2.1. Training GMDH Networks

Let us assume **X** as an $n \times m$ input data matrix, consisting of *n* training samples, each with *m* features, and $\boldsymbol{y^0}$ as a target vector:

$y^0 = (y_1^0, \dots\dots, y_n^0)^T$, $y_i^0 \epsilon \{0,1\}$ This represents a dataset $D = (X, y^0)$.

The transfer function of the neurons is a short polynomial of two variables, u_1 and u_2. In the case of nonlinearity, the transfer function g includes the corresponding nonlinear term. For example:

$$y = g(u, w) := w_0 + w_1 u_1 + w_2 u_2 + w_{12} u_1 u_2 \quad (2)$$

where $\mathbf{u} = (1, u_1, u_2, u_1 u_2)$ is the input vector, and $\mathbf{w} = w_0, w_1, w_2, w_{12}$ are the polynomial coefficients or weight vector. In the first layer ($r = 1$), the neurons connected to the input node are determined by pairs of m variables $x_1, \dots, x_m$, as follows:

$$u = (1, x_{i1}, x_{i2}, x_{i1} x_{i2}), i_1 \neq i_2 = 1, \dots\dots\dots, m \quad (3)$$

In subsequent layers ($r = 2,3, \dots$), the inputs of the neurons connected to the outputs y_{i1} and y_{i2} are the neurons from the previous layer (r−1), as follows:

$$u = (1, y_{i1}, y_{i2}, y_{i1} y_{i2}), i_1 \neq i_2 = 1, \dots\dots\dots, F \quad (4)$$

Note that *F* is the number of best neurons selected from the previous layer. Users typically define this number in advance, with $F < 0.4L_1$. Therefore, given the weight vector *w* and the *K*-th example for the input *u(K)*, we can calculate the output *y* from the neurons as follows:

$$y = g\left(u^{(K)}, w\right), K = 1, \dots\dots\dots, n. \quad (5)$$

To select the F best neurons, GMDH uses an external criterion calculated on unseen examples, which are not used for fitting the neuron weights. The unseen examples are obtained by dividing the dataset DDD into two non-overlapping subsets: $D_A = (X_A, y_A^0)$ and $D_B = (X_B, y_B^0)$, referred to as the training and examining (testing) datasets, respectively. Users define the sizes of these subsets as n_A and n_B, typically with $n_A = n_B$ and $n_A + n_B = n$.

Let us find a weight vector w^* that minimizes the sum of squared errors e of the neuron computed on subset $\boldsymbol{D}_A$:

$$e = \sum_K (g(u^K, w) - y_0^k)^2, K = 1, \dots\dots\dots, n_A \quad (6)$$

To find an optimal minimum, GMDH fits the neuron weights for a subset $\boldsymbol{D}_A$ using the least squares method (LSM). Note that LSM can provide an effective evaluation of the weights if the data follows a Gaussian distribution.

Suppose we have found an optimal weight vector w^* that minimizes the error (6) for all candidate neurons L*r* in layer *r* over a subset. Then, we can compute the external criterion values CR_i over the subset $\boldsymbol{D}_B$, which was not used for fitting the weights.

$$CR_i = \sum_K (g(u^K, w^*) - y_0^k)^2, K = 1, \dots\dots\dots, n_B, i = 1, \dots\dots, L_r \quad (7)$$

We can observe that the computed value CR_i depends on the behavior of the i-th candidate neuron over the unseen samples of subset $\boldsymbol{D}_B$. The calculated CR_ivalues in the r-th layer are sorted in ascending and increasing order:

$$CR_{i1} \leq CR_{i2} \leq \cdots \leq CR_F \leq \cdots \leq CR_L \quad (8)$$

Thus, F denotes the first best neurons. For each layer r, a minimum value of CR_m, corresponding to the best neuron, is defined. For example, $\boldsymbol{CR_m^{(r)} = CR_{i1}}$, which represents the first F best neurons, and these are then used in the subsequent layer r+1. Training and selection of this layer are performed iteratively using equations (6), (7), and

(8). The value of $CR_m^{(r)}$ gradually decreases as the number of layers *r* increases, and the network grows. When the value of CR reaches a minimum point and then starts to increase, for example, refer to Figure 3.

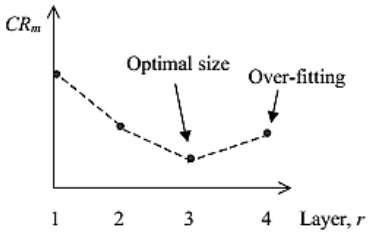


Fig. 3. The calculated value of the CR_m criterion for the growth of networks.

In Figure 3, the value of CR_m in the third layer of the polynomial network has decreased, with the example $\boldsymbol{CR_m^{(1)} > CR_m^{(2)} > CR_m^{(3)}}$. In layer r=3, the value of CR_m reaches its minimum. In the next layer, r=4, the value of CR_m increases, indicating that the network has overfitted according to the external criterion. Since the minimum CR was obtained in the previous layer, the training algorithm is stopped, and we conclude that the optimal polynomial network has grown at layer r=3.

4. COMPARISON OF THE PERFORMANCE OF THE BACKPROPAGATION ALGORITHM AND THE DATA SWARM GROUPING METHOD

In this section, the Data Swarm Grouping method has been simulated using MATLAB software. Both methods were tested and their results evaluated using the breast cancer dataset within this software. The simulation used the Wisconsin breast cancer database, which contains nine features as follows: tumor thickness, uniformity of cell size, uniformity of cell shape, abnormality of border adhesion, single-cell epithelial tissue size, simple cell nucleus, chromatin texture, normal cell nucleus, and mitotic rate. The output is classified into two categories: benign or malignant breast cancer. The objective of this section is to obtain the results of the Data Swarm Grouping method and compare its accuracy with the backpropagation algorithm.

Finally, by examining the averages for each method in MATLAB software, the best method can be identified after execution. In this section, two files from the Wisconsin dataset were prepared, with the training set consisting of 90% of the data and the test set consisting of 10% of the data, selected randomly from the Wisconsin dataset and configured in the software. Since the training set method uses the entire dataset for training, it typically achieves the best accuracy in most algorithms. In this paper, we present the corresponding results for each data entry.

Table 1: Results of the training data for benign and malignant breast cancer

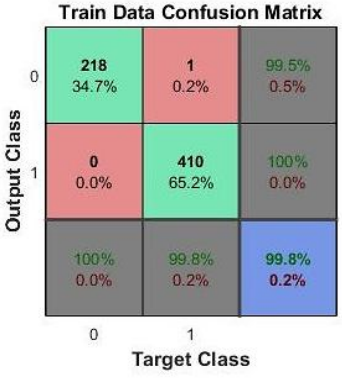


Table 2. Test Data Results for Benign and Malignant Breast Cancer Using the Data Swarm Grouping Method

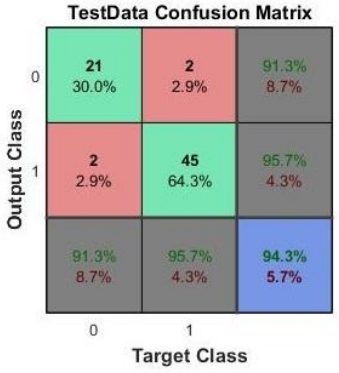
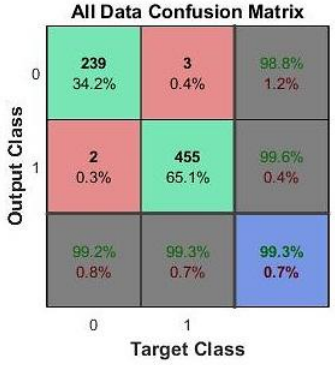


Table 3. Overall Results for Benign and Malignant Breast Cancer Using the Data Swarm Grouping Method



For comparison, the results of the backpropagation algorithm are presented, utilizing one input layer, one hidden layer, and one output layer. There are 9 neurons in the input layer, 6 neurons in the hidden layer, and one neuron in the output layer. Random values are assigned to the input weights and biases. 80% of the data is allocated for training, and the remaining 20% is used for testing. The learning rate is set to 0.7, and a maximum of 1000 cycles is considered. The result obtained from this algorithm shows an accuracy rate of 98.99% [25].

Table 4 presents the evaluation of the data swarm grouping method. The evaluation is conducted using confusion matrices for disease detection.

Table 4. Prediction Accuracy of the GMDH Model in Disease Diagnosis

Algorithm	Phase	Correct Prediction (%)	Incorrect Prediction (%)
GMDH	Training	99.8	0.2
GMDH	Testing	94.3	5.7

Based on the tables above and the results related to the backpropagation algorithm, it can be observed that the highest accuracy belongs to the proposed method with an accuracy of 99.3%.

5. CONCLUSION

In this paper, information about breast cancer, its causes, and methods of diagnosis were presented. The results obtained from the simulation performed in this study showed that the accuracy rate for diagnosing whether breast cancer is benign or malignant is 99.3% using a neural network based on the Grouped Data Shuffling Method (GMDH). Therefore, according to the experimental accuracy results of the backpropagation training algorithm, the

Grouped Data Shuffling Method achieves the highest accuracy. This paper leaves considerable room for further work, where the GMDH neural network could be combined with population-based optimization techniques (such as particle swarm optimization, differential evolution, etc.) and applied as a new method for classification.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] American Cancer Society (2013). Breast Cancer: Facts and Figures 2013. Atlanta: ACS.

[2] UCI Machine Learning Repository. (n.d.). Retrieved from http://archive.ics.uci.edu/ml/machine-learningdatabases/

[3] Mitra, S., & Hayashi, Y. (2000). Neuro-fuzzy rule generation: Survey in soft computing framework. IEEE Transactions on Neural Networks, 11(3), 748-768. https://doi.org/10.1109/72.846746

[4] Mitra, S., & Acharya, T. (2005). Data mining: Multimedia, soft computing, and bioinformatics. John Wiley & Sons. https://doi.org/10.1117/1.2179076

[5] Bellazzi, R., & Zupan, B. (2008). Predictive data mining in clinical medicine: Current issues and guidelines. International Journal of Medical Informatics, 77(2), 81-97. https://doi.org/10.1016/j.ijmedinf.2006.11.006

[6] Marcano-Cedeño, A., Quintanilla-Domínguez, J., & Andina, D. (2011). WBCD breast cancer database classification applying artificial metaplasticity neural network. Expert Systems with Applications, 38(8), 9573-9579. https://doi.org/10.1016/j.eswa.2011.01.167

[7] Malmir, H., Farokhi, F., & Sabbaghi-Nadooshan, R. (2013). Optimization of data mining with evolutionary algorithms for cloud computing application. 3rd International Conference on Computer and Knowledge Engineering (ICCKE), 343-347. IEEE. https://doi.org/10.1109/ICCKE.2013.6682821

[8] Hamilton, H. J., Shan, N., & Cercone, N. (1996). RIAC: A rule induction algorithm based on approximate classification. Computer Science Department, University of Regina.

[9] Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 4, 77-909. https://doi.org/10.1613/jair.279

[10] Pena-Reyes, C. A., & Sipper, M. (1999). A fuzzy-genetic approach to breast cancer diagnosis. Artificial Intelligence in Medicine, 17(2), 131-155. https://doi.org/10.1016/S0933-3657(99)00019-6

[11] Abonyi, J., & Szeifert, F. (2003). Supervised fuzzy clustering for the identification of fuzzy classifiers. Pattern Recognition Letters, 24(14), 2195-2207. https://doi.org/10.1016/S0167-8655(03)00047-3

[12] Paulin, F. (2011). Classification of breast cancer by comparing backpropagation training algorithm. International Journal on Computer Science and Engineering (IJCSE).

[13] Lavanya, D., & Rani, D. K. U. (2011). Analysis of feature selection with classification: Breast cancer datasets. Indian Journal of Computer Science and Engineering (IJCSE), 2(5), 756-763.

[14] Bhardwaj, A., & Tiwari, A. (2015). Breast cancer diagnosis using genetically optimized neural network model. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2015.01.065

[15] Zadeh, H. G., Haddadnia, J., Seryasat, O. R., & Isfahani, S. M. M. (2016). Segmenting breast cancerous regions in thermal images using fuzzy active contours. EXCLI Journal, 15, 532.

[16] Seryasat, O. R., & Haddadnia, J. (2018). Evaluation of a new ensemble learning framework for mass classification in mammograms. Clinical Breast Cancer, 18(3), e407-e420. https://doi.org/10.1016/j.clbc.2017.05.009

[17] Rahmani-Seryasat, O., Haddadnia, J., & Ghayoumi-Zadeh, H. (2015). A new method to classify breast cancer tumors and their fractionation. Ciência e Natura, 37(4), 51-57. https://doi.org/10.5902/2179460X19428

[18] Rahmani Seryasat, O., Kor, I., Ghayoumi Zadeh, H., & Shams Taleghani, A. (2021). Predicting the number of comments on Facebook posts using an ensemble regression model. *International Journal of Nonlinear Analysis and Applications*, *12*(Special Issue), 49-62.

[19] Kondo, T., & Ueno, J. (2007). Logistic GMDH-type neural network and its application to identification of X-ray film characteristic curve. Journal of Advanced Computational Intelligence and Intelligent Informatics, 11(3), 312-318. https://doi.org/10.20965/jaciii.2007.p0312

[20] Kondo, T. (1998). GMDH neural network algorithm using the heuristic self-organization method and its application to the pattern identification problem. Proceedings of the 37th SICE Annual Conference, 1143-1148. https://doi.org/10.1109/SICE.1998.742993

[21] Ivakhnenko, A. G. (1970). Heuristic self-organization in problems of engineering cybernetics. Automatica, 6(2), 207-219. https://doi.org/10.1016/0005-1098(70)90092-0

[22] Farlow, S. J. (Ed.). (1984). Self-organizing methods in modeling: GMDH type algorithms. Marcel Dekker Inc.

[23] Madala, H. R., & Ivakhnenko, A. G. (1994). Inductive learning algorithms for complex systems modeling. CRC Press Inc.

[24] Müller, J.-A., Lemke, F., & Ivakhnenko, A. G. (1998). GMDH algorithms for complex systems modeling. Mathematical and Computer Modeling of Dynamical Systems, 4, 275-315. https://doi.org/10.1080/13873959808837083

[25] Paulin, F., & Santhakumaran, A. (2011). Classification of breast cancer by comparing backpropagation training algorithms. International Journal on Computer Science and Engineering (IJCS). https://doi.org/10.5120/656-923