HORA 2019
www.horacongress.com

International Congress on Human-Computer Interaction, Optimization and Robotic Applications
July 5-7, 2019, Urgup, Nevşehir, Turkey

SETSCI Conference Proceedings
www.set-science.com
8: 152-157, 2019

An Efficient Medical Diagnosis Algorithm Based on a Hybrid Neural Network with a Variable Adaptive Momentum and PSO Algorithm

Alaa Ali Hameed[1], Naim Ajlouni[2], Adem Özyavaş[3], Zeynep Orman[4] and Ali Güneş[5]

[1]Department of Computer Engineering, Istanbul University-Cerrahpaşa, Turkey
Email: dr.alaa85@yahoo.com
[2]Department of Software Engineering, Istanbul Aydin University, Turkey
Email: naimajlouni@aydin.edu.tr
[3]Department of Computer Engineering, Istanbul Aydin University, Turkey
Email: ademozyavas@aydin.edu.tr
[4]Department of Computer Engineering, Istanbul University-Cerrahpaşa, Turkey
Email: ormanz@istanbul.edu.tr
[5]Department of Computer Engineering, Istanbul Aydin University, Turkey
Email: aligunes@aydin.edu.tr

Abstract – Neural Network has been used in a number of scientific fields including medical diagnosis. It is clear that classification is one of the main challenges in any medical diagnosis. However, the drawbacks of conventional neural network classifier are summarized as slow convergence and tendency to be trapped in local minima. The aim of this study is to present a hybrid Back-Propagation algorithm with a variable adaptive momentum (BPVAM) and particle swarm optimization (PSO) (BPVAM-PSO). The PSO is an efficient training algorithm as it does not require any complex calculations. The BPVAM variable momentum, in this case, will start with a high value, which increases the convergence rate; the momentum value will decrease as the error rate is decreased. The best weights of the BPVAM are passed to the PSO. The PSO uses the weights to determine the best set of parameters and as a result, the number of hidden neurons is reduced. This will improve performance further. To verify the efficiency of the BPVAM-PSO algorithm it will be used for the classification of a number of different medical datasets. The results of the proposed algorithm are compared against the performance of the conventional BP Neural Networks and BPVAM algorithm.

Keywords – Neural Networks, Particle Swarm, Adaptive momentum, Optimization, Medical Diagnosis

I. Introduction

With the increased variations in the stream of medical data flowing into the system with time, it becomes important for medical experts to utilize some of the newly developed special-purpose Intelligent Algorithms [1]-[3]. The use of such tools and algorithms will increase the accuracy and efficiency of the prediction and or diagnosis process [4]-[5]. There are two such techniques available for use these are machine learning (ML) and predictive modelling [6]-[7]. In this work we will use ML to achieve the task in hand. ML is an area of computer science which uses cognitive learning methods to program their systems without the need of being explicitly programmed [9].

Back-propagation Neural Network (BPNN) has been researched and applied in many scientific fields, including medical and nitration applications, to solve different problems [10]-[14]. However, conventional BPNN techniques suffer from some inherent problems which affect the performance of the algorithm which includes, slow convergence, and high steady-state error (SSE) [15].

The addition of some momentum to the adjustment expression is used to overcome the problem of slow convergence and high SSE. In this case, the momentum term is used to smooth out the descent path by removing changes in the gradient resulted from local anomalies. It suppresses any oscillations ace due to changes in the slope of the error [16]-[17].

Different versions of varied momentum algorithms have been proposed for different applications by researchers. In this study, the methodology employed by the recently developed Robust Back Propagation with Varied Adaptive Momentum RA-BPVAM will be employed [18].

PSO is applied as an optimization tool in numerous areas coupled with other existing algorithms such as neural networks and fuzzy logic [19]-[21]. The PSO performs the search to obtain the optimal solution referred to as particles, the trajectories are adjusted by a stochastic and deterministic component. Particles are influenced by its 'best' achieved position and group 'best' position otherwise it moves randomly. Particle i is expressed as both a position vectorx_i and velocity vector v_i [22]. The advantages of PSO parameter tuning method over other known methods is that it requires fewer parameters to achieve its acceptable solutions [23].

This work will investigate the efficiency of utilizing ML algorithms namely Robust Back Propagation with Variable Adaptive Momentum "RA-BPVAM" coupled with a well-known optimization method namely Particle Swarm

DOI: 10.36287/setsci.4.5.030

Optimization "PSO" to generate an efficient medical Diagnosis Algorithm.

II. BP ALGORITHM

D. E. Rumelhart et al [24], introduced the traditional BP algorithm, which was considered as a significant contribution to the neural network. The initial W_0 in the weight formula, in the iterative increment formula for the weights takes the form

$$\Delta w(n+1) = \eta \delta_o y(n) \qquad n = 0,1,\ldots \tag{1}$$

Where η >0 is the learning rate which indicates how far to go along the negative direction of the gradient. However, in BP the convergence speed is very slow due to the saturation behavior of the activation function in the network, which, is usually worse for a network with multi-hidden layer networks. This is due to the fact that output unit saturates with small decent gradient value, even in the case large output error, resulting in no significant progress in the weight adjustment. The second disadvantage is the difficulty in the choice of learning rate η to achieve both fast learning while and maintaining stable the learning procedure [16].

The basic requirement of BP is to minimize the overall output error during the learning process. The BP input layer training sets are estimated iteratively to predict correct outputs. The BP process is divided into two sub-processes the forward and backward process. In the forward process there are a number of parameters which includes the input x_i which are the inputs to the neural network with i neurons, w_{ij} are the weights of interconnections between inputs and hidden layers, and j neurons for the hidden layer. Assuming that the hidden transfer function is a sigmoid function, and the output transfer function is a linear activation function.

$$f(H_j) = \frac{1}{1 + e^{-(b_{in} + \sum_{i=1}^{N} x_i w_{ij})}}, \tag{2}$$

Where b_{in} is a bias input layer; hidden layer will pass through the activation function (f). After calculating the overall output by multiplying the output of the hidden layer neurons with the hidden layer weights w_{jk}, the results, then, pass through a sigmoid function (called threshold) as shown below in equation (3).

$$y_k = b_n + \sum_{j=1}^{m} w_{jk} f(H_j) \tag{3}$$

Where b_n is the bias of hidden layer and k output neurons.

To predict the correct output it is intended to obtain minimum error E for each pattern (p) by subtracting the overall output o from the target t; as shown in equation (4).

$$E = \frac{1}{2} \sum_{j=1}^{p} \left(t_j - o_j\right)^2 \tag{4}$$

In the backward process, weights on the connections between all layers will be updated to minimize the error between the target (or desired) and output until finding the optimum weights with minimum E.

Now by adding the learning and momentum coefficients to equation (1), the hidden layer equation with the updated weights will be defined by the following equation

$$\Delta w_{ji}(n+1) = \eta \delta_y x_i(n) + \alpha \Delta w_{ji}(n) \tag{5}$$

And the output layer equation with the momentum coefficient will be defined as

$$\Delta w_{kj}(n+1) = \eta \delta_o y_i(n) + \alpha \Delta w_{kj}(n), \quad n = 0,1,\ldots \tag{6}$$

Where n represents the number of iterations and η is the learning rate.

III. BPVAM ALGORITHM

The BPVAM was introduced by Hameed et al [18], where α (the adaptive momentum) is controlled by the learning rate parameter η. The BPVAM proposes a variable momentum which is expressed as:

$$\alpha(n) = \frac{\lambda}{1 - \beta^n} \tag{7}$$

Where $\lambda < \frac{2-2\beta}{\text{max eigenvalue of } R_{xx}}$ and this case β is the forgetting factor $(0 \ll \beta < 1)$,

The initial value of β is expected to be large. This will lead to the term $1 - \beta^n$ being close to unity. Therefore, the initial $\alpha(n)$ will be relatively large. However, its estimated that fast convergence of the weights update can be achieved through fewer number of iterations. This will further enhanced as the value of λ become small enough, as a result it provides low performance error for the weights update in (5) and (6). This means that equations (5) and (6) can be rewritten as

$$\Delta w_{ji}(n+1) = \eta \delta_y x_i(n) + \left(\frac{\lambda}{1 - \beta^n}\right) \Delta w_{ji}(n) \tag{8}$$

$$\Delta w_{kj}(n+1) = \eta \delta_o y_i(n) + \left(\frac{\lambda}{1 - \beta^n}\right) \Delta w_{kj}(n), \quad n = 0,1,\ldots \tag{9}$$

Where n represents the number of iterations and Δw is defined as updating the weights.

IV. PSO ALGORITHM

The PSO algorithm is initialized with a group of random particles (candidate solutions). This is followed by a search for an optimal solution by updating its individual variables. Figure 1 shows the updating procedure of a particle by vectorial representation. In each generation, each particle is updated based on 2 special particles: $pbest_{i,j}(n)$ is the personal best solution of each particle found so far, and $gbest_j(n)$ is the global best solution found so far by any particle in the swarm (population). The detailed algorithms of various methods are described below for completeness.

$$v_{i,j}(n+1) = w.v_{i,j}(n) + c_1 r_1 \left(pbest_{i,j}(n) - x_{i,j}(n)\right) + c_2 r_2 \left(gbest_j(n) - x_{i,j}(n)\right) \tag{10}$$

$$x_{i,j}(n+1) = x_{i,j}(n) + v_{i,j}(n+1) \quad (11)$$

The $v_{i,j}(n)$ and $x_{i,j}(n)$ variables in Figure 1 are respectively the j^{th} velocity component and the j^{th} $(j = 1,2,\ldots,D)$ position component of the i^{th} $(i = 1,2,3,\ldots,N)$ particle at generation . N is the size of the initial population (swarm). D is the dimension size of the search space. w is the inertial weight used to balance the global and local search. For the basic PSO, calculating the velocity updating and the position updating by Equations (10) and (11), respectively. $r1$ and $r2$ are two randomly generated numbers at each iteration ranging from 0 to 1. $c1$ and $c2$ are the acceleration constants with positive values.

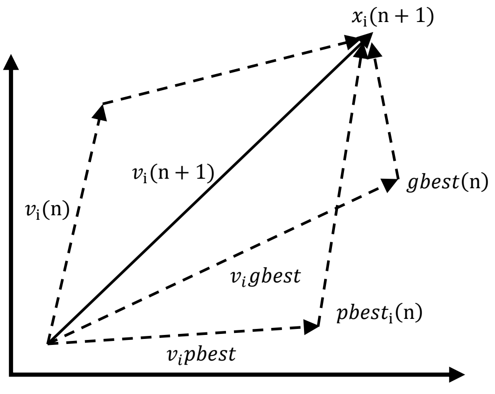

Fig. 1: the velocity and position updating of a particle

V. BPVAM-PSO ALGORITHM

The BPVAM-PSO is an optimization algorithm combining both the BPVAM and PSO into one algorithm. Both the BPVAM and the PSO algorithm have a high convergence speed ability and global search capabilities, hence the combination of the two complement each other. There are common features betwwen BPVAM and PSO, hence by using the PSO to optimize the weights of the BPVAM can further improve the learning and convergence abilities.

The fundamental idea for this hybrid algorithm is that at the initial stage the BPVAM algorithm will run. Due to the Adaptive momentum, it will converge very quickly and obtain acceptable weights with high accuracy and low SSE. These related weights are then passed to the PSO algorithm to find optimal weights values based on the values received from the BPVAM algorithm. Once the new and optimal weights are found by the PSO, the new values are fedback to the BPVAM to further improve its output results. The process is implemented in this manner to guarantee that even if the weights found by the PSO cause divergence rather than convergence, the BPVAM will overcome this problem with its adaptive momentum. This process is highlighted in the flow chart in Fig. 2. As it is seen the BPVAM-PSO algorithm's searching process is initialized by the BPVAM weights rather than random particles. All the particles are then updated according to the Equations (12) and (13).

The structure of the Pi particle is given by Eq. (13).

$$MSE = \frac{1}{2S}\sum_{i=0}^{S} e_i^2 \quad 12$$

$$P_i = [w1_{11}^{i}\ w1_{12}^{i} \ldots \Theta_{11}^{i} \ldots w2_{11}^{i}\ w2_{12}^{i} \ldots \Theta2_{11}^{i} \ldots .\ w3_{11}^{i} \ldots \Theta_{11}^{i}] \quad 13$$

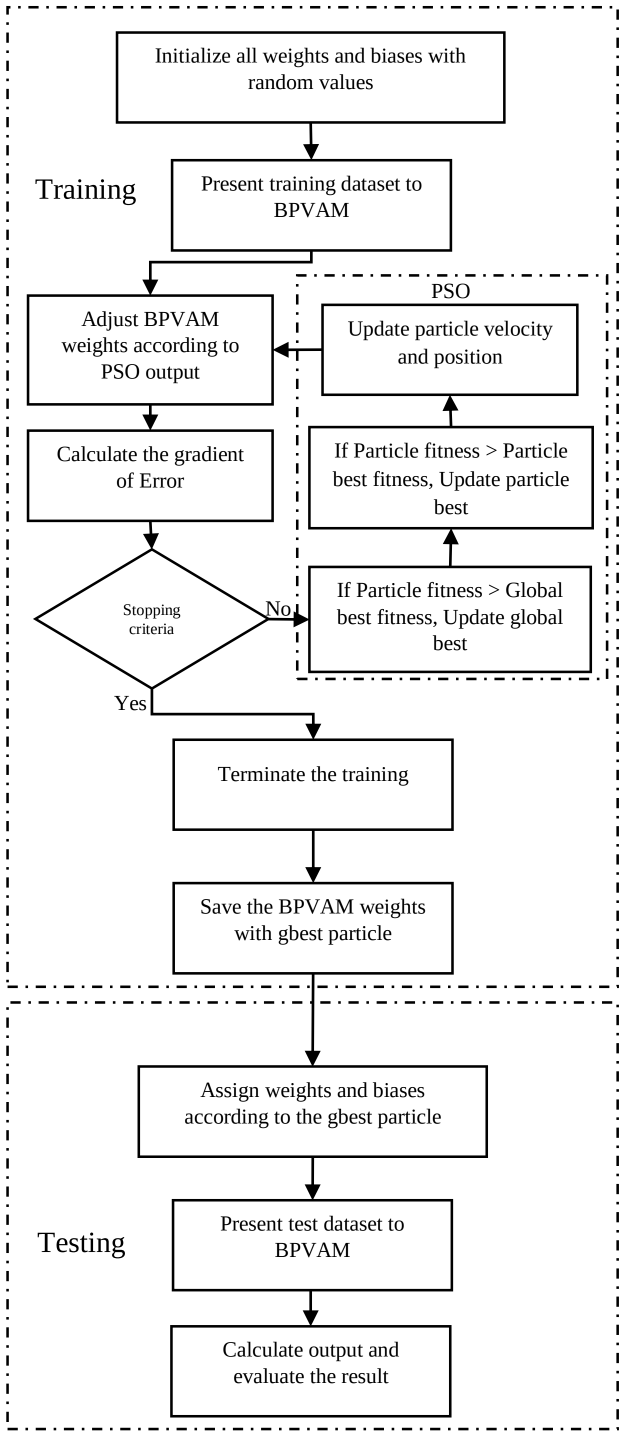

Fig. 2 BPVAM-PSO Flowchart of the testing and training process

The flowchart given in Figure 2 shows the training and testing processes of the BPVAM-PSO. Artificial neural networks begin the training operation with the initialization of random weights and biases. This indicates the numerical values associated with the layers of the artificial neural network. These weights and biases are individuals to each particle, as given in Eq. (13). The number of connections among layers indicates the population size or search space dimension. A stopping criterion that is selected as the maximum number of generations or fitness value is the target of the gbest particles as in Figure 2.

VI. Results and Simulations

In this study, it is proposed to carry out the test and simulation on two different data sets which are Diabetes and Lung cancer data sets. The performance of the proposed method is compared to the conventional BP and the BP with Variable Adaptive Momentum algorithms. The testing and training process is described by the flow chart as in Figure 3 below.

In this study, a normalization process was carried out on the data sets using a Min-Max normalization between 0 and 1. All the simulation codes were written and executed using Matlab on a Dell computer Intel core i7, 2.10 GHz.

The results of each of the tests will be considered separately and will be tabulated accordingly

A. Lung Cancer Dataset

The Lung Cancer dataset was obtained from the UCI repository "https://archive.ics.uci.edu/ml/datasets/Lung Cancer". The dataset consists of 32 instances, 56 attributes, and 3 classes of pathological lung cancers. In this test, the results of both conventional BP and the modified BPVAM were compared against the performance of the proposed hybrid BPVAM-PSO algorithm. The algorithms related parameters are highlighted in table 1. The test results are as shown in table 2. From the results, it can be seen that the Hybrid BPVAM-PSO algorithm obtained better accuracy than the other two algorithms. It can also be seen that the performance of both the BPVAM and the Hybrid BPVAM obtained much better accuracy than the BP algorithm during at the initial run time (60% accuracy at iteration 100 for both algorithms). However, as the number of iterations is increased the hybrid BPVAM-PSO algorithm outperformed the BPVAM algorithm by obtaining higher accuracy at iteration 300 (accuracy 70% for BPVAM-PSO, accuracy 60% for BPVAM). the results also show that the hybrid BPVAM-PSO outperformed the other two algorithms by obtaining a lower steady-state error SSE. It should be noted at the initial stages the BPVAM obtained a better accuracy than the hybrid BPVAM-PSO (SSE = 0.033 for BPVAM while SSE=0.035 for BPVAM-PSO at iteration 100). This situation changed as the number of iteration increased (SSE=0.001 for BPVAM-PSO and SSE=0.003 for BPVAM at iteration 300). This is due to the fact that the PSO requires a number of extra iterations to obtain optimized weight values to achieve the required task. The SSE behavior of the Lung Cancer Dataset is as shown in Figure 3.

B. Diabetes Dataset

The Diabetes Dataset was obtained from the Kaggle repository "https://www.kaggle.com/uciml/pima-indians-diabetes-database/data". This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The dataset consists of 768 instances, 8 attributes, and 2 classes. In this study, 70% of the data was used for training and 30% for testing. In this test, the results of both conventional BP and the modified BPVAM were compared against the performance of the proposed hybrid BPVAM-PSO algorithm. The algorithms related parameters are highlighted in table 3.

Table 1: Algorithms related parameters

Lung Cancer Dataset Parameters	BP	BPVAM	BPVAM-PSO
Hidden layer neurons	20	20	20
Learning rate (η)	0.7	0.7	0.7
Momentum (α)	0.3	-	-
Speed control (λ)	-	0.05	0.05
Forgetting factor (β)	-	0.97	0.97
No of particle	-	-	20
C1	-	-	1.2
C2	-	-	1.2
Inertia weight (w)	-	-	0.9

Table 2: Lung cancer test results according to the consumed time

Lung Cancer Dataset	Accuracy (%) with 100 Iterations	SSE	Accuracy (%) with 300 Iterations	SSE
BP	40.00	0.05	60.00	0.005
BPVAM	60.00	0.03	60.00	0.003
BPVAM-PSO	60.00	0.035	70.00	0.001

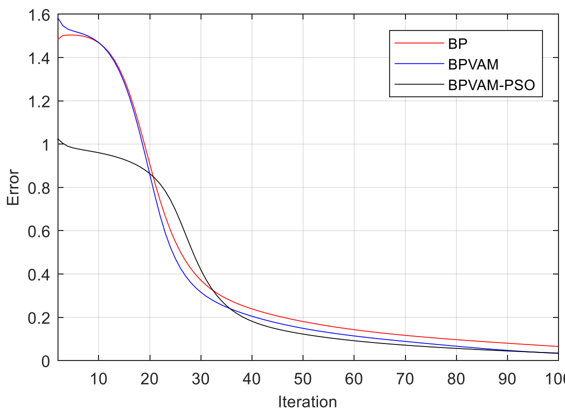

Fig. 3: Lung cancer dataset SEE behavior

The test results are as shown in table 4. From the results, it can be seen that the BPVAM-PSO algorithm obtained better accuracy than the BPVAM and the conventional BP algorithms during the first 300 iterations (BP=79.49%, BPVAM=80.77%, and BPVAM-PSO=82.05). However, as the number of iterations increased both the proposed hybrid

BPVAM-PSO and the BPVAM obtained the same accuracy 82.05% which is much better than the conventional BP at 80.77%. As far as the SEE is concerned the result shows that the hybrid algorithm maintained a better performance all the way through the execution time of the algorithm by attaining a SEE equivalent to 36.26, it was noticed in this case the the proposed algorithm obtained a low SEE rate and maintained that rate for the rest of the execution time, while the other two algorithms showed a variation in the value during the execution of the algorithm. It can be concluded that the proposed algorithm achieved the required accuracy and lower SEE much faster than the other algorithms and maintained this achievement. The SSE behavior of the Diabetes Dataset is as shown in Figure 4.

Table 3: Algorithms related parameters

Diabetes Dataset Parameters	**BP**	**BPVAM**	**BPVAM-PSO**
Hidden layer neurons	8	8	8
Learning rate (η)	0.01	0.01	0.01
Momentum (α)	0.01	-	-
Speed control (λ)	-	0.1	0.1
Forgetting factor (β)	-	0.8	0.8
No of particle	-	-	9
C1	-	-	0.2×10^{-10}
C2	-	-	0.2×10^{-10}
Inertia weight (w)	-	-	0.9

Table 2: Diabetes test results according to the consumed time

Diabetes Dataset	**Accuracy (%) with 300 Iterations**	**SSE**	**Accuracy (%) with 500 Iterations**	**SSE**
BP	79.49	40.28	80.77	36.63
BPVAM	80.77	39.07	82.05	36.46
BPVAM-PSO	82.05	38.13	82.05	36.26

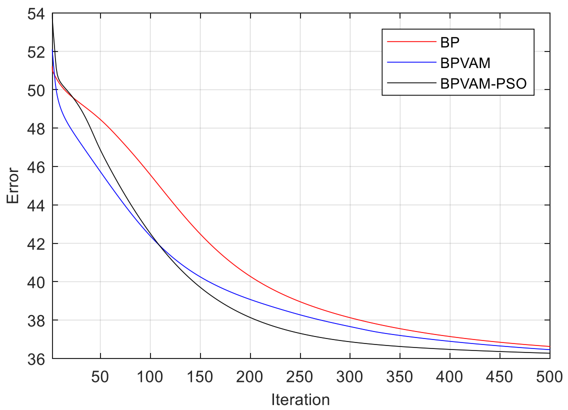

Fig. 4: Diabetes dataset SEE behaviour

VII. CONCLUSION

In this work, the proposed hybrid BPVAM-PSO and both the conventional and BPVAM algorithms were tested under the same conditions. The Results showed that the BPVAM-PSO performed more efficiently than both the conventional and BPVAM algorithms in both tests using two different medical datasets namely lung cancer and diabetes datasets obtained from UCI and Kaggle data repositories respectively.

It was noticed that the BPVAM algorithm due to the initial high momentum rate performed as expected with high efficiency. However, as the number of iterations increased the proposed hybrid BPVAM-PSO algorithm made a good recovery and managed to outperform the BPVAM. This is expected as the PSO is initialized with the best weights of the BPVAM to further improve the results. It can be concluded that the hybrid algorithm shows good enough performance which needs to be investigated further against big datasets before a final conclusion can be made.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered.

REFERENCES

[1] J. Xu, P. Wu, Y. Chen, Q. Meng, H. Dawood and M. M. Khan, "A Novel Deep Flexible Neural Forest Model for Classification of Cancer Subtypes Based on Gene Expression Data", *IEEE Access*, vol. 7, pp. 22086-22095, 2019.

[2] D. P. Aldryan and A. Annisa, "Cancer Detection Based on Microarray Data Classification with Ant Colony Optimization and Modified Backpropagation Conjugate Gradient Polak-Ribiére", *In 2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 13-16, 2018.

[3] S. Reis, P. Gazinska, J. H. Hipwell, T. Mertzanidou, K. Naidoo, N. Williams, S. Pinder and D. J. Hawkes, "Automated classification of breast cancer stroma maturity from histological images", *IEEE Transactions on Biomedical Engineering*, vol. 64, issue 10, pp. 2344-2352, 2017.

[4] F. F. Ting and K. S. Sim, "Self-regulated multilayer perceptron neural network for breast cancer classification", *In 2017 International Conference on Robotics, Automation and Sciences (ICORAS)*, pp. 1-5, 2017.

[5] N. Yukinawa, S. Oba, K. Kato and S. Ishii, "Optimal aggregation of binary classifiers for multiclass cancer diagnosis using gene expression profiles", *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, issue 2, pp. 333-343, 2008.

[6] R. Jafari-Marandi, S. Davarzani, M. S. Gharibdousti and B. K. Smith, "An optimum ANN-based breast cancer diagnosis: Bridging gaps between ANN learning and decision-making goals", *Applied Soft Computing*, vol. 72, pp. 108-120, 2018.

[7] F. Zhanga, Z. Li, B. Zhang, H. Du, B. Wang and X. Zhang, Multi-modal Deep Learning Model for Auxiliary Diagnosis of Alzheimer's Disease. Neurocomputing, 2019.

[8] S. E. Lacy, S. L. Smith and M. A. Lones, "Using echo state networks for classification: A case study in Parkinson's disease diagnosis", *Artificial intelligence in medicine*, vol. 86, pp. 53-59, 2018.

[9] A. L. Samuel, "Some studies in machine learning using the game of checkers. II—recent progress", *In Computer Games I*, pp. 366-400. Springer, New York, NY, 1988.

[10] Y. Fei, G. Pengdong and L. Yongquan, "Bolt force prediction using simplified finite element model and back propagation neural networks", *In 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pp. 520-523, 2016.

[11] Y. Zhang and Z. Wang, "A hybrid model for blood pressure prediction from a PPG signal based on MIV and GA-BP neural network", *In 2017 13th International Conference on Natural*

Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 1989-1993, 2017.

[12] Y. Liu, W. Jing and L. Xu, "Cascading model based back propagation neural network in enabling precise classification", *In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 7-11, 2016.

[13] Y. Fang, M. Pang and B. Wang, "A course control system of unmanned surface vehicle (USV) using back-propagation neural network (BPNN) and artificial bee colony (ABC) algorithm", *Procedia computer science*, vol. 111, pp. 361-366, 2017.

[14] N. Bisoyi, H. Gupta, N. P. Padhy and G. J. Chakrapani, "Prediction of daily sediment discharge using a back propagation neural network training algorithm: A case study of the Narmada River, India", *International journal of sediment research*, vol. 34, issue 2, pp. 125-135, 2019.

[15] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors", *Cognitive modeling*, vol. 5, issue 3, 1, 1988.

[16] H. Shao and G. Zheng, "A new BP algorithm with adaptive momentum for FNNs training", *In 2009 WRI Global Congress on Intelligent Systems,* vol. 4, pp. 16-20, 2009.

[17] H. Shao and G. Zheng, "Convergence analysis of a back-propagation algorithm with adaptive momentum", *Neurocomputing*, vol. 74, issue 5, 749-752, 2011.

[18] A. A. Hameed, B. Karlik and M. S. Salman, "Back-propagation algorithm with variable adaptive momentum", *Knowledge-Based Systems*, vol. 114, pp. 79-87, 2016.

[19] Y. Ding, W. Zhang, L. Yu and K. Lu, "The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis", *Energy*, vol. 176, pp. 582-588, 2019.

[20] H. Hamdi, C. B. Regaya and A. Zaafouri, "Real-time study of a photovoltaic system with boost converter using the PSO-RBF neural network algorithms in a MyRio controller", *Solar Energy*, vol. 183, pp. 1-16, 2019.

[21] H. Liang, J. Zou, Z. Li, M. J. Khan and Y. Lu, "Dynamic evaluation of drilling leakage risk based on fuzzy theory and PSO-SVR algorithm", *Future Generation Computer Systems*, vol. 95, pp. 454-466, 2019.

[22] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Networks*, 1995.

[23] H. Cui, M. Shu, M. Song and Y. Wang, "Parameter selection and performance comparison of particle swarm optimization in sensor networks localization", *Sensors*, vol. 17, issue 3, 487, 2017.

[24] D.E. Rumelhart, J. L. McClelland and the PDP Research Group. Parallel Distributed Processing Explorations in the Microstructure of Cognition. Mass.: MIT Press, 1986.