



AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

Aqua Site Classification Using Neural Network Models

N. Deepa¹, K. Ganesan²

¹ School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India.

² TIFAC-CORE in Automotive Infotronics, VIT University, Vellore, Tamil Nadu, India.

Abstract

India being one of the major producers of fish contributes 5.5 percent of global fish production and ranks second in the world after China. The production of aquaculture mainly depends on the quality of land selected for aqua farming. Neural Network algorithms have been applied to classify the aquaculture sites based on 6 input variables viz., water, soil, support, infrastructure, input and risk factor. An artificial neural network (ANN) consists of huge number of interconnected elements called neurons that work together to solve a specific problem. An Artificial Neural network can be used for classification, prediction, pattern recognition etc., through a learning process. In this paper, the models were constructed using three Neural Network algorithms viz., Back Propagation Network (BPN), Radial Basis Function (RBF) and Linear Vector Quantization (LVQ). The models classify each aquaculture site into 3 classes viz., suitable, moderate and unsuitable. From the results of the three models, it has been found that Radial Basis Function model not only gives accurate results but also time taken for training the dataset is less when compared with the other two Neural Network models. The results obtained from the neural network models were validated with the results of the fuzzy model.

Keywords

Neural Networks, Aquaculture, Land Classification, Back Propagation Network, Radial Basis Function, Linear Vector Quantization.

Deepa, N. and Ganesan, K. (2016) "Aqua Site Classification Using Neural Network Models", *AGRIS on-line Papers in Economics and Informatics*, Vol. 8, No. 4, pp. 51 - 58. ISSN 1804-1930, DOI 10.7160/aol.2016.080405.

Introduction

Due to protein shortage in many countries, Aquaculture development has become significant important nowadays. The quality of the aqua site chosen for the aquaculture development leads to the success of the projects (Boyd and Clay, 1998). Multiple variables were used for the identification and classification of aqua sites which leads to the success of aquaculture farming projects (McKindsey et al., 2006) A tool called DMTIOLA was developed for making decision on selection of best site for aquaculture farming using a mathematical function. The mathematical function was derived by the integration of few multi criteria decision making approaches (Mahalakshmi et al., 2012). A decision making model using Fuzzy logic was suggested for the classification of aquaculture sites based on six input parameters (Mahalakshmi and Ganesan, 2015).

The Neural Network Model consists of interconnected neurons. They exchange

the activation signals based on the network topology in the form of activation function. Learning process is done by generating and adjusting weights in neural networks. Neural Network models are developed not only for parallel processing of data but can also be used for classification (Konecny et al., 2010). Neural Network algorithms were also used for prediction in the Knowledge-management (Svoboda, 2007)

A Radial Basis Function based neural network model was developed for the prediction of water temperature in the aquaculture ponds of sea cucumber in which it was shown that RBF produced high accuracy in shorter learning time (Min Sun et al., 2012). Freshness of silver carp was determined using BPN based model (Zhang et al., 2009). An Artificial Neural Network model was used to analyse biological growth state based on various water quality parameters in order to forecast the actual yield (Changhui et al., 2010). Many neural network algorithms are used in economic and statistical analyses.

In this study, three neural network models suitable for aquaculture classification were identified viz., BPN and RBF which comes under supervised Neural network models and LVQ which comes under unsupervised neural network model. The purpose of the models is to classify aqua sites into three classes viz., suitable, moderate and unsuitable (Mahalakshmi and Ganesan, 2013).

Materials and methods

Data sets

The essential parameters to be taken into account for aquaculture land suitability are the good quality water, quality of soil, salinity, pollution, threats to flood, temperature, pollution, infrastructural facilities, access to markets and necessary input (Nath et al., 2000). Twenty four variables (Hajek and Boyd, 1994; Mahalakshmi and Ganesan 2009) viz., water (ten sub-variables), Soil (five sub-variables), support (two sub-variables), infrastructure (three sub-variables), input (one sub variable) and risk factor (three sub-variables) listed in Table 1 were used for the construction of Neural Network models.

Sl No.	Main variables	Sub variables
1	Water	pH, Salinity, Total alkalinity, Dissolved Oxygen, Total hardness/ Total alkalinity, Free NH ₃ -N, Total hardness, H ₂ S, Temperature, Transparency.
2	Soil	Salinity, pH, Clay content, Available Phosphorus, Organic Carbon
3	Support	Distance to university/college, Distance to NGO
4	Infrastructure	Distance to processing plants, Distance to Local Markets, Distance to natural fry
5	Input	Animal wastes
6	Risk factors	Winter rain, Flood and cyclone, Pollution

Source: own processing

Table 1: Aquaculture land classification of main and sub variables.

Study area

Kallamandal located in West Godavari district of Andhra Pradesh state, India was selected as study area. This study area was selected because coastal aquaculture is considered as major commercial occupation in this area. The soil, water, support, infrastructure, input, and risk related data used in this paper were collected from 40 randomly chosen aqua farms located at Mogalthur (A)

(25 sites), Kallamandal (B) (15 sites) (Mahalakshmi and Ganesan 2013). Data gathered from A was used for training the BPN, RBF neural network models and data from B was used to test the BPN, RBF models. LVQ being unsupervised neural network requires data for training only. Dataset B was used for developing LVQ model. And the results of neural network models were validated with the results of Fuzzy model (Mahalakshmi and Ganesan 2013).

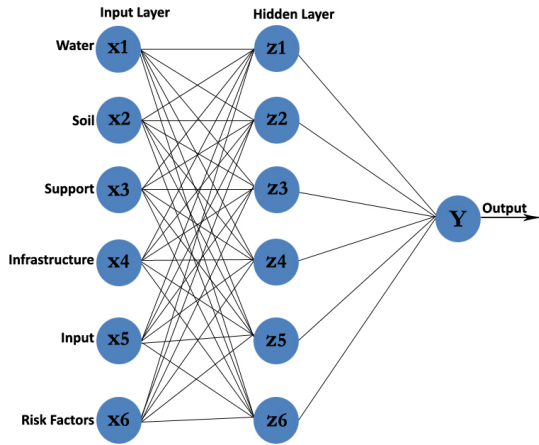
The prime set of criteria identified has been applied to DMTIOLA tool developed by Mahalakshmi. This tool has converted the data A and B to its relative closeness values using various Multi Criteria Decision Making Methods such as Rank sum weight, Analytical Hierarchical Process and TOPSIS method (Mahalakshmi and Ganesan, 2013).

Implementation

Matlab software was used to write programs for developing the three neural network models for aqua site classification. The software (<http://www.mathworks.com>) has many in-built mathematical functions which will be useful for coding. It is easy to code and understand.

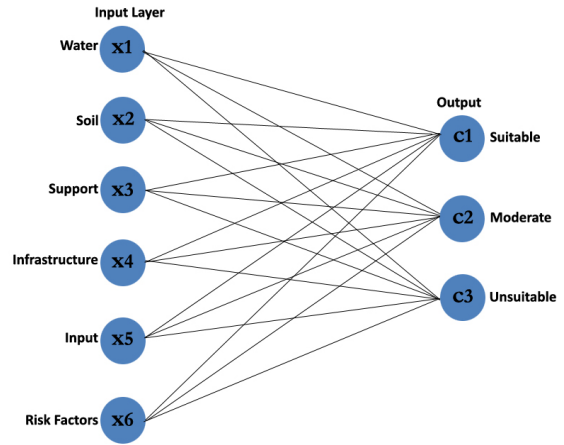
Neural Network and training algorithm

An artificial neural network is an data processing method which works similar to the human brain. It consists of huge number of interconnected elements called neurons that work together to solve a specific problem. Supervised learning is the process of providing the network with some sample inputs and outputs for training purpose whereas unsupervised learning will be provided only with input data. Supervised learning algorithms compare the calculated output with the expected response whereas unsupervised learning algorithm will group the input data into clusters by identifying some unique pattern. In this paper two supervised learning algorithms namely Back Propagation Network (BPN) algorithm, Radial Basis Function (RBF) algorithm and one unsupervised algorithm namely Linear Vector Quantization (LVQ) are used to classify the aquaculture lands. The proposed supervised neural network models for BPN and RBF is shown in Figure 1 and unsupervised Neural Network Model for LVQ is shown in Figure 2.



Source: own processing

Figure 1. Proposed supervised Neural Network Model.



Source: own processing

Figure 2. Proposed unsupervised Neural Network Model.

Back Propagation Network Model

Back Propagation Network (Arthur and Yu-Chi, 1969) is a multi-layer neural network supervised training algorithm. It uses gradient-descent based delta learning rule which minimizes total mean squared error. The BPN training algorithm is divided into 3 stages namely Feed Forward Stage, Back Propagation of errors and updation of weights and biases. Updated final weights and biases at final epoch are used while testing the algorithm to compare the calculated output with the expected response.

BPN model was developed using the following steps:

Various parameters used in the model:

X – Input vector. In our case there are 6 input parameters and 25 training input patterns available for Aqua farm classification

T – Target vector

δ_k = error at output unit Y_k . Here $k = 1$

δ_j = error at hidden unit Z_j where $j = 1$ to 6

α = Learning rate

V_{0j} = bias at hidden unit where $j = 1$ to 6

Z_j = hidden unit j

W_{0k} = bias at output unit where $k = 1$

Y_k = output unit k

V_{ij} = weights between input and hidden layer. Here $i = 1$ to 6 and $j = 1$ to 6

W_{jk} = weights between hidden and output layer where $j = 1$ to 6 and $k = 1$

1. Initialize weights between input and hidden layer V_{ij} , weights between hidden and output layer W_{jk} , bias at hidden unit V_{0j} and bias

at output unit W_{0k} to random values.

2. For each training vector do the following steps

Feed Forward stage

Find output Y_k by applying bipolar activation function

$$Y_k = f(Y_{-ink}) \text{ where } k = 1 \text{ and } f(x) = \frac{1}{1 + e^{-\lambda x}} \text{ also } \lambda = 1 \quad (1)$$

$$\text{where } Y_{-ink} = W_{0k} + \sum_{j=1}^6 Z_j W_{jk} \text{ where } j = 1$$

to 6 and $k = 1$

$$Z_j = f(Z_{-inj}) \text{ where } j = 1 \text{ to } 6 \text{ and } f(x) = \frac{1}{1 + e^{-\lambda x}} \text{ also } \lambda = 1$$

$$Z_{-inj} = V_{0j} + \sum_{i=1}^6 X_i V_{ij} \text{ where } j = 1 \text{ to } 6$$

3. Back propagation of errors

Error at hidden units is obtained using

$$\Delta_j = \delta_{-inj} = f(Z_{-inj}) \text{ where } j = 1 \text{ to } 6 \quad (2)$$

$$\text{where } \delta_{-inj} = \sum_{k=1}^1 \delta_k W_{jk} \text{ where } j = 1 \text{ to } 6 \text{ and } k = 1.$$

Error term $\delta_k = (T_k - Y_k) f'(Y_{-ink})$ where $k = 1$

$$f'(Y_{-ink}) = \frac{1}{1 + e^{-\lambda x}} \left(1 - \frac{1}{1 + e^{-\lambda x}} \right)$$

where and $x = Y_{-ink}$

4. Updation of weights and Biases

Updation of weights between input and hidden layer are calculated using

$$V_{ij(new)} = V_{ij(old)} + \Delta V_{ij} \quad (3)$$

where $\Delta V_{ij} = \alpha \Delta_j X_i$ where $i = 1$ to 6 and $j = 1$ to 6

Updation of bias at hidden unit is obtained using

$$V_{0j(new)} = V_{0j(old)} + \Delta V_{0j} \quad (4)$$

where $\Delta V_{0j} = \alpha \Delta_j$ where $j = 1$ to 6

Updation of weights between hidden and output layer are calculated using the formula

$$W_{jk(new)} = W_{jk(old)} + \Delta W_{jk} \quad (5)$$

where $\Delta W_{jk} = \alpha \delta_k Z_j$ where $k = 1$ and $j = 1$ to 6

Updation of bias at output unit is obtained using

$$W_{0k(new)} = W_{0k(old)} + \Delta W_{0k} \quad (6)$$

where $W_{0k} = \alpha \delta_k$ where $k = 1$

5. Repeat the steps upto any specified condition to stop the iteration. The condition can be either the number of epochs or error rate.

For each testing vector we do the following steps:

By substituting the weight and bias values obtained during training process, we calculate the output Y. Then compare the output vector Y and target vector T and calculate error term using

$$E = Tk - Yk \quad (7)$$

Radial Basis Function (RBF) Model

The Radial Basis Function (Broomhead and Lowe, 1988) is a feed-forward neural network supervised training Algorithm. Radial Basis Function Networks are used in image processing applications and in many regularization networks. Radial Basis Function uses gaussian function with which it differs from the Back Propagation Network algorithm. RBF networks have three network layers viz., input layer, hidden layer and output layer. Hypothetical connections are formed between input layer and hidden layer and weighted connections are formed between hidden and output layer. Updation of weights are done by RBF training algorithm. In RBF network, the Gaussian activation function is used to compute the output. We calculate the radial basis function by choosing the centres. Here all the input vectors are selected as centres to ensure sufficient sampling.

The following steps were used to develop the RBF model:

RBF Training algorithm

1. Initialize the weights between input and hidden units to some small random values

2. For each Input pattern do the following steps:
3. Each input unit (X_i , $i = 1$ to 6) sends signals to hidden unit
4. Calculate the radial basis function by choosing the centres using the equation (1). Here all the input vectors are selected as centres to ensure sufficient sampling.

$$Mi(X_i) = e^{-\sum_{j=1}^r \left(\frac{1}{\sqrt{2\Pi}} e^{-\frac{1}{2}(X_{ji} - \bar{X}_{ji})^2} \right)} \quad (8)$$

where \bar{X}_{ji} = centre of the RBF unit for input values

$\frac{1}{\sqrt{2\Pi}}$ is width of the RBF unit
 X_{ji} = jth variable of input pattern

5. Calculate weight using

$$W = M' * t \quad (9)$$

where M' = Inverse of radial basis function

t = target units

6. Obtain the calculated output y using

$$y = W' * M \quad (10)$$

7. Calculate the training error using

$$E = \sum (t - y)^2 \quad (11)$$

and repeat the steps

RBF Testing algorithm

For each input pattern do the following steps:

1. Find the output vector y using Gaussian value and weights obtained during training process

$$y = W' * M \quad (12)$$

2. Compare the target vector T and output vector y and calculate testing error term using the formula given in equation (7).

Linear Vector Quantization Model – Unsupervised Neural Network

When learning is done using data without output and error calculation is not done, such network is called unsupervised network. In this, the network may react to one of many output classes on training. Target values are given for input training pattern in Linear Vector Quantization (LVQ) network. After training process, the LVQ net assigns each and every input vector to some output class. In LVQ, some input vectors are used as weights and the other input vectors are used for training

the network. Next step is to identify the winner unit by calculating the Euclidean distance. The input vector with minimum distance is identified as winner. The weights were updated based on the comparison of winner index with the target class. The iterations are repeated by decrementing the learning rate. The architecture diagram for unsupervised neural network LVQ model is shown in Figure 2.

In this neural network model, first 3 input vectors are used as the reference vectors or weight vectors and the 15 site's dataset have been taken for training the network. The following steps were used to develop LVQ model:

Parameters used in LVQ model:

Input vector X_i where $i = 1, 2, \dots, 15$

Weight vector for the j^{th} output neuron $W_j = (W_{1j}, W_{2j}, \dots, W_{6j})$. Here first 3 input vectors are used as weight vectors.

C_j = output class represented by the j^{th} neuron, where $j = 1$ for suitable, 2 for moderate and 3 for unsuitable.

T = Target class for input X

α = Learning rate = 0.7

LVQ Training Algorithm:

1. Initialize weight vectors by assigning the first 3 input vectors to weights.
2. Until stopping condition becomes false, do the following steps from 2 to 6.
3. For each training input vector x , repeat the steps from 3 to 4
4. Find j such that $D(j)$ is minimum.
5. Calculate the Euclidean distance between the input vector and the weight vector of the j^{th} neuron using the formula

$$D(j) = \sqrt{\sum_{i=1}^6 (X_i - W_{ij})^2} \quad (13)$$

6. Update the weights of the j^{th} neuron using the formula:

if $C_j = T$ then

$$W_{j(new)} = W_{j(old)} + \alpha (X - W_{j(old)})$$

(i.e. move the weight vector W towards the input vector X)

if $C_j \neq T$ then

$$W_{j(new)} = W_{j(old)} - \alpha (X - W_{j(old)})$$

(i.e. move W away from X)

7. Reduce learning rate α

8. Test the condition used to stop the iteration: Here stopping condition is the learning rate reaching a small value.

Results and discussion

In this paper, the results of Back Propagation Network, Radial Basis Function Network models and Linear Vector Quantization model have been compared.

The two main parameters of Back Propagation network algorithm are learning rate and momentum factor. Learning rate is denoted by α which can vary from 0 to 1. The learning rate is used to control the weight adjustment during training process. It affects the rate of convergence of Back Propagation training algorithm. If the value of learning rate is high, the learning process is fast and if the value of learning rate is low, the learning process is said to be slow. Momentum factor value can be used to increase the speed of training process. BPN algorithm without momentum factor parameter has higher training time and slower convergence. Thereby momentum factor is used to accelerate the learning process. In Table 2 below, we list the results of BPN for various momentum factors and learning rate values for 5000 iterations.

Momentum Factor	Learning Rate	Error	Time
0.21	0.5	0.1382	2.1966
0.21	0.8	0.0553	1.6050
0.5	0.8	0.2443	1.5970
0.1	0.9	0.0480	1.7117

Source: own processing

Table 2: Results of Back Propagation Network with different Momentum factor and Learning rate values for 5000 iterations.

From the first two cases of Table 2 we observe that by increasing the learning rate, error and time taken for testing the data set has been reduced. In the third case, it has been observed that the increase in momentum factor leads to the increase in error rate. Finally from the 4th case, we see that an increase in learning rate and decrease in momentum factor lead to reduced error rate and time. The relationship between the target and calculated output for BPN model is shown in Figure 3. In the figure, target vectors with suitable output class (value 2.32) have been correctly classified as suitable (values 2.17, 2.19) in calculated output and the target vectors with moderate output class (value 1.01) have been incorrectly classified as unsuitable (values 1.81,

0.89) in calculated output. The Total Mean Square Error calculated by BPN algorithm is 0.7970. Time taken for testing the dataset is found to be 5.3 sec.

By applying Back propagation Network model, out of 15 testing sites, 2 sites have been found to be suitable, 5 sites have been classified as moderate and 5 sites have been found to be unsuitable with respect to the Fuzzy model results. These results were validated with fuzzy model results (Mahalakshmi and Ganesan, 2013) and the accuracy rate is found to be 80%.

From the results of RBF model, we found that out of 15 testing sites dataset, one site have been correctly classified as suitable, six sites have been accurately found to be moderate and six sites have been correctly classified as unsuitable with respect to the results of Fuzzy model. The results of RBF model are found to be 87% accurate with respect to fuzzy model results.

The relationship between the target and calculated output for RBF model are shown in Figure 4. In the diagram, target vector with suitable output class (value 2.32) has been erroneously classified as moderate output class (value 1.80) in calculated output and target vector with moderate output class (value 1.81) has been incorrectly classified as suitable output class (value 2.18) in calculated output. The Total Mean Square Error calculated by RBF algorithm is 0.3121. Time taken for testing the dataset is found to be 0.02 sec.

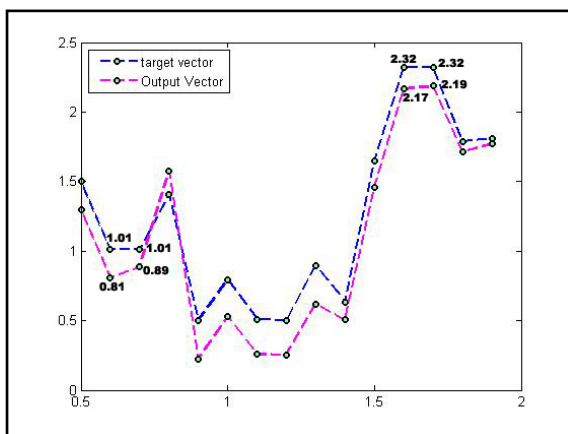
In this neural network model, first 3 input vectors from Table 1 are used as the reference vectors or weight vectors, and the 15 sites dataset (including first 3 datasets) have been taken for training the network. The learning rate is

initialized to 0.7. Out of 15 dataset, 11 sites were clustered to correct output category using LVQ model thereby achieving 73% of accuracy and the results are validated with respect to the fuzzy model results.

Using the three neural network models, the test datasets have been classified and the classification results are compared with the results of Fuzzy model (Mahalakshmi & Ganesan, 2013) which is specified in the last column of Table 4. From the above comparison, the results have been validated and it has been assessed that neural network model using BPN produced 80% accurate results and RBF has produced 87% correct results whereas LVQ has given 73% correct results. Table 3 shows the results of neural network models using BPN, RBF and LVQ.

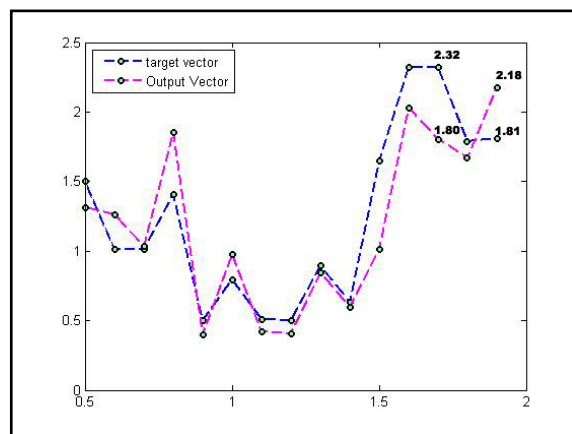
The performance of the test data sets are shown graphically with target vector and output vector plotted for BPN and RBF Network in Figure 3 and Figure 4. The LVQ algorithm provides less accurate results with reduced time compared to BPN and RBF. The time taken for LVQ algorithm is 0.0066 sec whereas BPN takes 5.3 sec and RBF takes 0.02 sec.

From the results of MATLAB programs, it has been identified that Back propagation model performance is dependent more on the parameters. It consumes more time when compared with the other two models. Radial Basis Function gives accurate results and time taken for training the dataset is also very low. Even though Linear Vector Quantization takes less time compared to BPN and RBF, it does not classify all the input dataset correctly. The results show that Radial Basis Function Network are faster



Source: own processing

Figure 3: Relationship between target and calculated output for BPN model.



Source: own processing

Figure 4: Relationship between target and calculated output for RBF model.

Land suitability Classification	Neural Network Models			Fuzzy Model Results (Mahalakshmi & Ganesan, 2013)
	BPN	RBF	LVQ	
Suitable	2/15	1/15	1/15	2/15
Moderate	5/15	6/15	4/15	7/15
Unsuitable	5/15	6/15	6/15	6/15
Accuracy	80%	87%	73%	
Time in sec	5.3	0.02	0.0066	

Source: own processing

Table 3: Results of BPN, RBF and LVQ Models.

and simpler as compared to Back Propagation Network, Linear Vector Quantization and can be used for classification problems.

Conclusion

In this paper, three artificial neural network models viz., Back propagation network, Radial Basis function network and Linear Vector Quantization have been used for aquaculture site classification. The results obtained from test datasets are compared and are validated. It has been found that the output seems to be in close correlation with the experimental results produced by fuzzy model (Mahalakshmi and Ganesan, 2013) and aquaculture experts results. The error rate, testing time of three algorithms were compared. LVQ testing time seems to be lower compared to BPN and RBF. The RBF model has classified the dataset quite faster with better accuracy. The error goal was reached by RBF in only one epoch while BPN took nearly 5000 epochs. LVQ has achieved 73% correct results within 10 epochs and the algorithm is simple to implement when

compared to BPN and RBF. An important point to be noted is that BPN selects number of hidden layers by trial and error method while RBF has used only one hidden layer with growing number of neurons. The results of BPN concluded that the learning rate and momentum factor are the parameters which affect the performance of BPN model in aqua site classification and it was shown by varying momentum factor and learning rate. It has been concluded that RBF model is suitable for classification problems which consumes less time with higher accuracy compared to BPN and LVQ models.

Acknowledgements

The authors would like to thank TIFAC-CORE in Automotive Infotonics, VIT University, Vellore, Tamil Nadu, India for providing necessary hardware and software support. This work forms part of the R&D activities of TIFAC-CORE funded by the Department of Science and Technology, Government of India.

Corresponding author:

N. Deepa

School of Information Technology and Engineering, VIT University, Vellore - 632 014, Tamil Nadu, India.

Phone: +91 9597058880, E-mail: deepa.rajesh@vit.ac.in

References

- [1] Boyd, C. E. and Clay, J. W. (1998) "Shrimp aquaculture and the environment", *Scientific American*, Vol. 278, No. 6, pp. 58-65. ISSN 0036-8733.
- [2] Broomhead, D. S., and Lowe, D. (1988) "Multi-variable functional interpolation and adaptive networks", *Complex Systems*, Vol. 2, No. 3, pp. 321-355. ISSN 0891-2513.
- [3] Bryson, A. E. and Ho, Y.-Ch. (1969) "*Applied Optimal Control*", Blaisdell Publishing Co., USA. ISBN 0891162283.
- [4] Changhui, D., Yanping, G., Jun, G., Xinying, M. and Songsong, L. (2010) "*Research on the Growth Model of Aquaculture organisms based on Neural Network Expert System*", Sixth International Conference on Natural Computation, pp. 1812-1815. ISBN 978-1-4244-5961-2.

- [5] Hajek, F. and Boyd, C. E. (1994) "Rating soil and water information for aquaculture", *Aquaculture Engineering*, Vol. 13, No. 2, pp. 115-128. ISSN 0144-8609. DOI: 10.1016/0144-8609(94)90009-4.
- [6] Konecny, V., Trenz, O. and Svobodova, E. (2010) "Classification of companies with the assistance of self-learning neural networks", *Agricultural Economics*, Vol. 56, No.2, pp. 51-58. ISSN 1805-9295.
- [7] Mahalakshmi, P. and Ganesan, K. (2009) "Mahalanobis Taguchi System based variables selection for shrimp aquaculture development", *Computers and Electronics in Agriculture*, Vol. 65, No. 2, pp. 192-197. ISSN 0168-1699. DOI 10.1016/j.compag.2008.09.003.
- [8] Mahalakshmi, P. and Ganesan, K. (2013) "Application of rank sum, TOPSIS and pair-wise comparison methods for prioritising aquaculture sites", *Indian Journal of Fisheries*, Vol. 60, No. 3, pp. 55-58. ISSN 0970-6011.
- [9] Mahalakshmi, P. and Ganesan, K. (2013) "*Decision Making Models for aquaculture farming development*", Today and Tomorrow's Printers and Publishers, New Delhi. ISBN 9788170194699.
- [10] Mahalakshmi, P. and Ganesan, K. (2015) "Mamdani fuzzy rule based model to classify sites for aquaculture development", *Indian Journal of Fisheries*, Vol. 62, No. 1, pp. 110-115. ISSN 0970-6011.
- [11] McKindsey, C. W., Thetmeyer, H., Landry, T. and Silvert, W. (2006) "Review of recent carrying capacity models for bivalve culture and recommendations for research and management", *Aquaculture*, Vol. 261, No. 2, pp. 451-462. ISSN 0044-8486. DOI 10.1016/j.aquaculture.2006.06.044.
- [12] Min, S., Ji, Ch., Daoliang, L. (2012) "Water Temperature Prediction in Sea Cucumber Aquaculture Ponds by RBF Neural Network Model", *Proceedings of the International Conference on Systems and Informatics*, pp. 1154-1159. ISBN 978-1-4673-0197-8.
- [13] Nath, S. S., Bolte, J. P., Ross, L. G. and Aguilar-Manjarrez, J. (2000) "Applications of geographical information systems (GIS) for spatial decision support in aquaculture", *Aquaculture Engineering*, Vol. 23, No. 1-3, pp. 233-278. ISSN 0144-8609. DOI: 10.1016/S0144-8609(00)00051-0.
- [14] Svoboda, E. (2007) "Knowledge-management in managerial work of business management", *Agricultural Economics*, Vol. 53, No. 7, pp. 298-303. ISSN 1805-9295.
- [15] Zhang J., Li, X. Y., Wang, W., Zhou, Z. (2009) "Determination of Freshness of Freshwater Fish based on BP-ANN and Bio-impedance Characteristics", *Global Congress on Intelligent Systems*, IEEE, pp. 68-71. ISSN 2155-6083. ISBN 978-0-7695-3571-5. DOI 10.1109/GCIS.2009.39.