Improvement of radial basis function neural network with accelerated particle swarm optimization for corrosion rate prediction of 3C steel in seawater environment

Qingping Jian

Chengdu Vocational & Technical College of Industry, Equipment Manufacturing College, Chengdu 610218, China

675429404@qq.com

ABSTRACT. *For accurately prediction of 3C steel corrosion rate in seawater environment, this paper establishes a radial basis function neural network (RBFNN) and improves it with accelerated particle swarm optimization (APSO). Specifically, the centers, spreads and connection weights of each radial basis function (RBF) were automatically tuned by the APSO, and the number of RBFs in the RBFNN was minimized by choosing a special fitness function. The APSO-optimized RBFNN was proved through a case study to have good prediction accuracy and self-learning ability. The research findings provide an accurate, adaptive and easily-to-train prediction model for 3C steel corrosion rate in the seawater environment.*

RÉSUMÉ. *Cet article établit un réseau de neurones à fonction de base radiale (RBFNN) et l'améliore avec l'optimisation accélérée de l'essaim de particules (APSO). Plus précisément, les centres, les spreads et les poids de connexion de chaque fonction de base radiale (RBF) ont été automatiquement ajustés par l'APSO et le nombre de RBF dans la RBFNN a été réduit au minimum en choisissant une fonction de fitness spécifique. Une étude de cas a prouvé que la RBFNN optimisée pour APSO avait une bonne précision de prédiction et une capacité d'autoapprentissage. Les résultats de la recherche fournissent un modèle de prévision précis, adaptable et facile à entraîner pour le taux de corrosion de l'acier 3C dans l'environnement de l'eau de mer.*

KEYWORDS: *radial basis function neural network (RBFNN), seawater environment, accelerated particle swarm optimization (APSO), prediction model, corrosion rate.*

MOTS-CLÉS: *réseau neuronal à fonction de base radiale (RBFNN), environnement d'eau de mer, optimisation accélérée des essaims de particules (APSO), modèle de prédiction, taux de corrosion.*

DOI:10.3166/ACSM.42.577-587

1. Introduction

The 3C steel, a kind of carbon steel with excellent operational and process properties, has been widely adopted in such fields as offshore engineering, chemical processing and oil production. With the extensive application of the steel in the seawater environment, the corrosion prevention has become a research hotspot (Srisuwan *et al.*, 2008; Caceres *et al.*, 2007; Samide *et al.*, 2008; Song *et al.*, 2012; Paik *et al.*, 2004; Liu *et al.*, 2008; Hajeeh, 2003; Liu *et al.*, 2005; Liu *et al.*, 2008; Cao *et al.*, 2012). Seawater is one of the most corrosive natural electrolytes. In the seawater environment, carbon steel is gradually eroded by chemicals or through electrochemical reactions.

Over the years, many models have been developed to simulate the corrosion of carbon steel, namely, empirical formula, numerical simulation, general linear model, and artificial neural networks (ANNs). The ANNs stand out thanks to the effectiveness in the prediction of carbon steel corrosion. However, it is difficult to optimize the connection weights in the ANNs, not to mention their slow convergence and proneness to the local optimum trap.

These defects give birth to the radial basis function neural network (RBFNN) (Bishop, 1995), which is a feedforward neural network with only three layers (i.e. the input layer, the hidden layer and the output layer). The generation of the RBFNN is equivalent to curve fitting in a high-dimensional space, or the interpolation of the training data into a multi-dimensional surface (Schwenker *et al.*, 2001). As a function approximator, the RBFNN boasts a faster training speed than a multilayer perceptron. This network has been successfully applied in many areas, especially in functional approximation and nonlinear control.

However, there are still some difficulties with the design of the RBFNN, including identifying the number of radial basis functions (RBFs). The current identification approaches are generally empirical and time-consuming, as they are trial-and-error in nature. Another key difficulty in RBFNN design is the selection of the free parameters (i.e. the centers, spreads and connection weights) for the RBFs. The two available selection strategies each has its shortcomings. The expert selection strategy is too inaccurate to handle complex and ill-defined problems, while the gradient-type learning strategy is inapplicable in complex, high-dimensional problems.

To overcome these difficulties, this paper puts forward an efficient accelerated particle swarm optimization (APSO) learning algorithm for RBF design. This algorithm helps to avoid trial-and-error and local optimum trap, and enables the self-generation of centers, spreads and connection weights of each RBF. The resulting RBFNN can approach the desired system response in a robust and automatic manner. Then, the APSO was adopted to set up a rational RBFNN for the prediction of 3C steel corrosion rate in the seawater environment, and determine the proper number of RBFs.

The APSO is a variant of the intelligent optimization algorithm named particle swarm optimization (PSO). The PSO can obtain the global optimal solution through

information exchange and sharing between individual particles and the swarm. The APSO outshines the PSO in convergence speed, because it only uses the global optimal solution to guide the particle update process (Kennedy and Eberhart, 1995). The fast convergence, strong robustness and independence of specific solving model have made the APSO a desirable optimization tool for model parameters. Hence, this paper develops a RBFNN model optimized by the APSO (hereinafter referred to as the APSO-RBFNN) for the prediction of 3C steel corrosion rate in seawater environment. The model was proved to have good accuracy and generalization ability in corrosion rate prediction.

The remainder of this paper is organized as follows: Section 2 introduces the structure of the RBFNN; Section 3 provides the details on the APSO; Section 4 sets up the APSO-RBFNN and verifies its effectiveness through a case study; Section 5 wraps up this paper with several conclusions.

2. RBFNN structure

As shown in Figure 1, a RBFNN generally has three layers, that is, an input layer, a hidden layer and an output layer. The selection of hidden layer RBFs is the key to the RBFNN design. The most popular RBF is the Gaussian basis function:

$$h(x, c_i, \delta_i) = \exp\left[-\left(\frac{\|x - c_i\|^2}{2\delta_i^2}\right)\right] \tag{1}$$

where $\|x - c_i\|$ is the Euclidean distance between an input vector x and a center c_i; δ_i is the error of the i-th RBF. The RBFNN output can be calculated by the weighted average method:

$$f(x) = \sum_{i=1}^{m} w_i \cdot h_i(x) \tag{2}$$

where w_i is the i-th weight between the hidden and output layers; m is the number of hidden nodes; $h_i(x)$ is the output of the i-th hidden node.

The above description shows that the RBFNN is determined by the contour of the RBF h_i(x), which is in turn defined by parameters $\{c_{i1}, \delta_{i1}; c_{i2}; \delta_{i2}, c_{in}; \delta_{in}\}$ and connection weights w_i. The RBFNN performance varies with the parameter sets $R = \{c_{i1}, c_{i2}, \ldots, c_{in;}\delta_i; w_i, 1 \le i \le m\}$. Let m be the number of RBFs to be constructed. Then, a total of $m(n+l)$ parameters should be selected to design the optimal RBFNN. In this paper, the parameter selection is considered as a search problem, and the parameter set R is found in the search space by an APSO evolutional learning method.

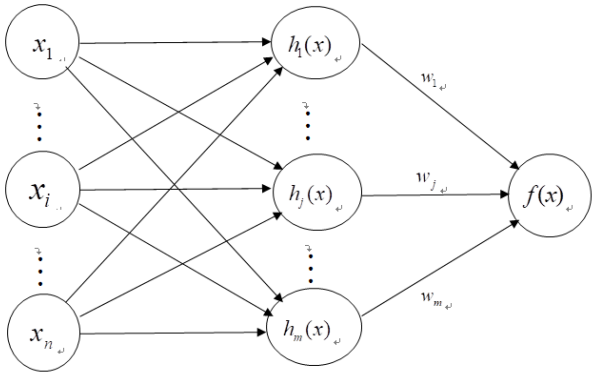

Figure 1. RBFNN structure

3. The APSO

To obtain the optimal solution, the original PSO firstly initializes a set of random particles (solutions) and then looks for the optimal solution iteratively in the search space of moving particles. During the iteration, the speed and position of each particle are updated according to the best-known individual and global solutions until the optimal solution is discovered. The update mechanisms for particle speed and position can be expressed as (Yang, 2010):

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_1[(p_{ij} - x_{ij}(t)] + c_2 r_2[p_{gj} - x_{ij}(t)] \quad (3)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (4)$$

where $1 \le i \le n$; $1 \le j \le d$; c_1 and c_2 are two acceleration factors; r_1 and r_2 are two random numbers in [0, 1]; w is the inertia factor. The position changes in the interval of $[-x_{j,max}, x_{j,max}]$ while the speed changes in the interval of $[-v_{j,max}, v_{j,max}]$. If either x_{ij} or v_{ij} exceeds the corresponding interval, the boundary value will be adopted for calculation.

In the original PSO, each particle zigzags during the iteration under the combined impact of the best-known individual and global solutions. To solve the problem, Yang (Feng, 2006) modified the PSO into the APSO, which only adopts the global optimal solution in speed update. Hence, the speed update formula (3) can be simplified as:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r + c_2[p_{gj} - x_{ij}(t)] \tag{5}$$

where r is a random number in [0, 1]. It can be seen that the second term item on the left side of formula (3) was removed, while formula (4) was retained as the position update formula. To further improve the convergence speed, formulas (4) and (5) were combined as the new position update formula:

$$x_{ij}(t+1) = (1-c_2)x_{ij}(t) + c_2 p_{gj} + c_1 r \tag{6}$$

To avoid the local optimum trap, the speed term was replaced with a stochastic term $c_1 r$. For the stability of the optimization algorithm, the value of c_1 needs to decrease with the growing number of iterations:

$$c_1 = \delta^t \tag{7}$$

where δ is a random number in (0, 1) (usually between 0.7~0.9); $t \in [0, t_{max}]$, with t_{max} being the maximum number of iterations. Let L_j and U_j be the lower and upper limits of the number of RBFs of the j-th particle, respectively, and r_j (j=1,2,..., p) be the number of RBFs available for selection. Then, the number of RBFs selected by the j-th particle must fall in $[L_j, U_j]$.

4. Prediction model based on APSO-RBFNN

The APSO-based evolutional learning aims to build an appropriate RBFNN with the fewest number of RBFs. Let $R_j^* = \{r_j, c_{ij1}, c_{ij2}, \ldots, c_{ijn}, \delta_{ij}, w_{ij}; 1 \le i \le m\}$ be the set of parameters (i.e. RBFs) used to form the RBFNN. The proper number of RBFs can be approached with the following fitness function:

$$f(R_j^*) = g_1(m(R_j^*)) * g_2(J(R_j^*)) \tag{8}$$

where $m(R_j^*)$ is the number of RBFs corresponding to the j-th particle parameter set, with R_j^* being acquired by the proposed RBF selection method; $J(R_j^*)$ is the performance measure with respect to the solution of the j-th particle. In view of the desired objectives, the performance of each individual can be evaluated by the following functions:

$$g_1(m(R_j^*)) = \exp(-(m(R_j^*)/\sigma_R) \tag{9}$$

$$g_2(m(R_j^*)) = 1/ERROR(R_j^*) \tag{10}$$

where ERROR(R^*_j) is the error between the desired and actual outputs of the j-th

RBFNN, which is determined by the individual parameter set (R_j^*) corresponding to the same input; σ_R is a user-defined constant. The value of σ_R determines the shape of the fitness function, forming a flexible method to realize any objective. Under the guidance of the proposed fitness function, the APSO can develop the optimal RBFNN capable of achieving high accuracy, that is, a small ERROR, with fewer RBFs. The APSO-based automatic generation of the optimal RBFNN is much more time-efficient than the traditional trial-and-error methods that determines the proper RBFNN parameters. With fewer RBFs, the optimal RBFNN simplifies the computation and reduces the memory load.

Given the suitable fitness function *f*(.), the selection problem can be viewed as the search problem below:

$$\underset{R_j^* \in \Re}{MAX} \quad f(R_j^*) \tag{11}$$

where R_j^* is the possible position of the *j*-th particle (RBF) in the search space (*R*); $f(R_j^*)$ is the fitness of the individual *R*.

The corrosion rates of 3C steel (Table 1) in different seawater environments generated by Liu *et al.* [9] were adopted to verify the effectiveness of our prediction model. The dataset covers the corrosion data of 46 3C steel samples in different seawater environments, measured via electrochemical means.

During the training of the APSO-RBFNN prediction model, five attributes (i.e. temperature, dissolved oxygen, salinity, pH value and oxidation–reduction potential) were taken as input variables while the corrosion rate was the output variable. Then, the APSO-RBFNN and the contrastive neural networks were learned and tested with the same training and test samples, which were selected randomly by Liu *et al.* Out of Liu's samples, the samples numbered 7, 10, 14, 19 and 21 were selected as the test samples while the remaining 41 samples were taken as the training samples. All of them were normalized before use.

Assuming that the number of available RBFs falls in the target RBFNN was given 20 RBFs at the start. Hence, 140 parameters $\{c_{i1}, c_{i2}, \ldots, c_{i5}; \delta_i; w_i, 1 \le i \le 20\}$ should be selected efficiently from the solution space. The other parameters were configured as follows: the number of iterations G is 1,000, the scaling factors are c_1=1.5 and c_2=1.5, and the constant for the fitness function σ_R is 100. Then, the fitness function can be defined as:

$$\begin{aligned} f(R_j) &= g_1(m(R_j^*)) * g_2(J(R_j^*)) \\ &= \exp(-m(R_j^*)/\sigma_R) * (MSE(R_j^*)) \end{aligned} \tag{12}$$

where MSE is the mean square error between the actual and the desired outputs. The results of the APSO-RBFNN prediction model were presented in Figures 2 and 3. Specifically, Figure 2 shows the relationship between the MSE and the number of

iterations and Figure 3 displays the relationship between the RFB numbers and the number of iterations.

Table 1. 3C steel corrosion rates in different seawater environments

No.	Temperature (℃)	Dissolved oxygen ($mg \cdot L^{-1}$)	Salinity (ppt)	pH value	Oxidation-reduction potential (mV)	Corrosion rate ($\mu A.cm^{-2}$)
1	25.9	6.71	30.1	5.1	378	16.4
2	29.35	6.09	29	6.3	400	16.9
3	27.9	6.18	31.5	7	363	15.57
4	24	7.95	30.2	8.1	324	13.65
5	28	5.05	31.4	9.2	240	13.24
6	27.32	3.21	29.31	8.2	281	12.91
7	27.87	6.55	31.68	7.2	356	14.06*
8	28.27	6.98	28.2	6.6	384	15.47
9	30.7	7.15	31.74	6.5	401	16.28
10	29.37	6.82	30.12	6.2	414	17.11*
11	24.27	0.8	32.56	8.1	171	3.61
12	27.45	2.6	35.37	7.96	287	7.94
13	27.23	4.2	31.94	7.89	289	9.63
14	27.48	5.9	32.39	7.83	331	10.578*
15	28.75	6.8	32.22	8	340	11.43
16	28.52	8.4	32.1	8.01	345	12.52
17	28.45	9.9	31.95	7.93	309	22.64
18	23.95	7.61	9.17	8.04	231	10.94
19	24.73	6.06	17.33	7.88	321	11.446
20	24.6	7.52	24.42	7.57	210	11.83
21	24.51	7.02	32	8.16	308	12.553*
22	23.65	6.51	41.34	7.67	245	8.402
23	16.74	7.11	33.55	8.25	178	10.85
24	21.11	6.03	33.44	8.03	295	11.448
25	25.57	6.7	32.19	8.09	325	11.872
26	31.16	4.38	33.21	7.94	242	8.924
27	25.62	34.89	5.32	7.9	385	15.966
28	24.95	16.29	6.8	7.82	341	12.12
29	24.5	18.37	5.31	7.93	302	12.07
30	25.59	21	7.04	7.95	244	11.4
31	26.11	34.84	2.82	7.8	335.2	11.288
32	24.96	40	6.32	8.08	254	9.3
33	9.5	32.31	4.26	8.2	195	10.56
34	12.05	32.04	4.95	8.17	232	11.04
35	14.86	32.51	6.3	7.95	198	11.06
36	28.13	34.34	5.14	7.8	362.9	13.93
37	24.17	16.09	7.68	8.04	283.8	11.55
38	23.54	15.04	8.27	8.06	243.8	11.72
39	25.31	15.22	7.59	9.32	246.7	11.39
40	12.55	37.9	6.42	7.49	235.3	10.52
41	16.81	39.49	6.61	7.73	258.7	10.24
42	24.09	36.72	5.59	7.83	281.8	9.93
43	26.34	35.97	3.25	7.98	367.1	14.37
44	25.35	16.94	4.05	8	341.2	15.07
45	26.07	35.34	4.07	7.94	404	18.13
46	26.52	34.48	4.94	7.9	326.1	11.828

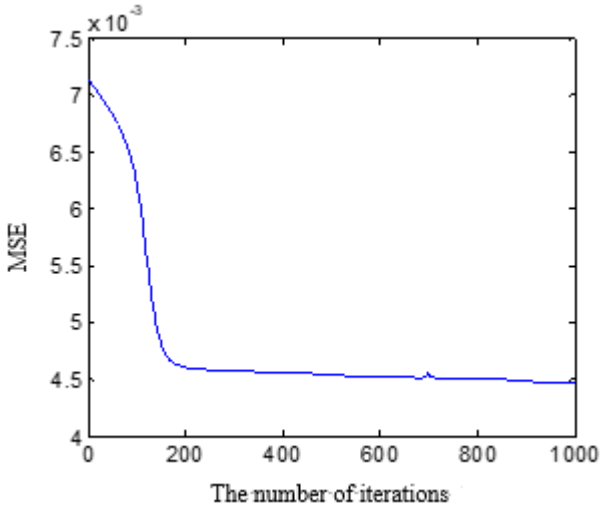

Figure 2. The relationship between the MSE and the number of iterations

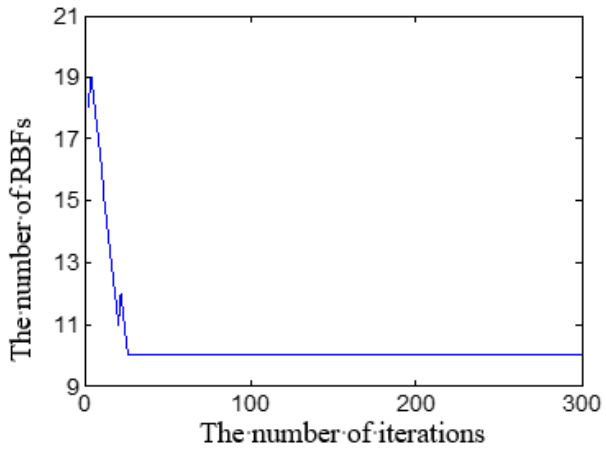

Figure 3. The relationship between the number of RBFs and the number of iterations

The parameter set of the optimal RBFNN is shown in Table 2. Simulation results indicate that 10 RBFs are sufficient for the prediction of 3C steel corrosion rate. Next, APSO-RBFNN was compared with the backpropagation neural network (BPNN) and the genetic algorithm-backpropagation neural network (GA-BPNN) in

terms of the prediction of 3C steel corrosion rate. The comparison results are listed in Table 3.

Table 2. The structure parameters of the RBFNN

i	c_{i1}	c_{i2}	c_{i3}	c_{i4}	c_{i5}	δ_i	w_i
1	0.889	0.278	0.753	0.663	0.547	0.049	0.813
2	0.846	0.358	0.875	0.300	0.705	0.205	0.599
3	0.553	0.659	0.681	-0.059	1.336	0.793	0.548
4	0.146	0.867	0.652	0.471	0.662	0.152	0.719
5	0.535	0.621	0.541	0.677	0.517	0.114	0.498
6	1.005	0.709	0.238	0.143	0.331	0.695	-0.566
7	0.368	0.494	0.397	0.719	0.622	0.097	0.357
8	0.836	0.281	0.561	0.801	0.305	0.232	0.884
9	0.538	0.389	0.183	0.704	1.275	1.227	0.813
10	0.857	-0.198	0.691	0.647	0.477	0.447	-0.541

Table 3. Performance comparison between different prediction models

No.	Experimental rate ($\mu A.cm^{-2}$)	BPNN		GA-BPNN		APSO-RBFNN	
		Predicted rate ($\mu A.cm^{-2}$)	Relative error (%)	Predicted rate ($\mu A.cm^{-2}$)	Relative error (%)	Predicted rate ($\mu A.cm^{-2}$)	Relative error (%)
7	14.06	15.081	7.26	13.981	0.56	14.23	-1.2
10	17.11	16.533	-3.37	17.048	0.36	17.35	-1.4
14	10.578	10.116	-4.37	10.961	0.62	11.45	-8.2
21	12.553	12.786	1.86	12.776	1.78	12.67	-0.87

From Table 3 and Figure 4, it can be seen that the APSO-RBFNN achieved a smaller relative error than the BPNN under the sample training and test conditions. Despite having a smaller relative error than our model, the GA-BPNN cannot determine the number of hidden layers and nodes in the BPNN without experienced modeler or repeated experiments, while our model can adjust the number of hidden layer nodes with the special fitness function and optimize the model parameters. That is why our model realized high accuracy with the fewest hidden layer nodes. The comparison proves that the proposed APSO-RBFNN is a feasible way to adaptively learn model parameters and determine the network structure.

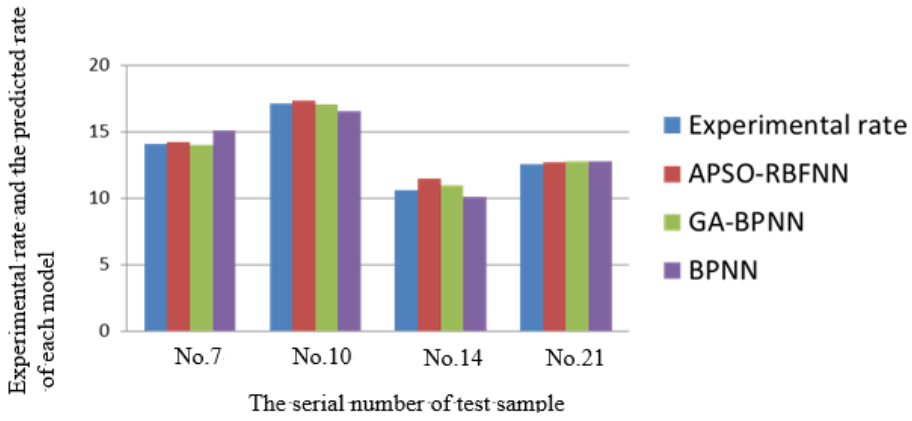

Figure 4. Comparison between the experimental rate and the predicted rate of each model

5. Conclusions

This paper puts forward an evolutional RBFNN model based on the APSO to predict 3C steel corrosion rate under seawater environment. The design strategy can automatically adjust the number of hidden layer nodes and the related parameters (centers, spreads and connection weights) in the RBFNN at the same time. In this way, the APSO-RBFNN model eliminates the need for experienced modeler or repeated experiments, which are required in BPNN prediction models. In addition, our design strategy accelerates the speed of network construction. The comparative analysis of experimental data shows the APSO-RBFNN outperformed the traditional neural network prediction approaches in prediction accuracy and adaptive modelling ability.

References

Bishop C. M. (1995). Neural networks for pattern recognition. *Oxford: Oxford University Press*. https://doi.org/10.1007/3-540-44732-6_7

Caceres L., Vargas T., Herrera L. (2007). Determination of electrochemical parameters and corrosion rate for carbon steel in un-buffered sodium chloride solutions using a superposition model. *Corrosion Science*, Vol. 49, No. 8, pp. 3168-3184. https://doi.org/10.1016/j.corsci.2007.03.003

Cao C. L., Cheng X. W., Zhang G. Y. (2012). Study on improving prediction accuracy of 3C steel corrosion rate. *China Well and Rock Salt*, Vol. 43, No. 5, pp. 25-28.

Feng H. M. (2006). Self-generation RBFNs using evolutional PSO learning. *Neurocomputing*, Vol. 70, No. 1, pp. 241-251. https://doi.org/10.1016/j.neucom.2006.03.007

Hajeeh M. (2003). Estimating corrosion: a statistical approach. *Materials and Design*, Vol. 24, No. 7, pp. 509-518. https://doi.org/10.1016/S0261-3069(03)00110-9

Kennedy J., Eberhart R. C. (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

Liu J. J., Lin Y. Z., Li X. Y. (2008). Numerical simulation for carbon steel flow-induced corrosion in high-velocity flow seawater. *Anti-Corrosion Methods and Materials*, Vol. 55, No. 2, pp. 66-72. https://doi.org/10.1108/00035590810859430

Liu X. Q., Tang X., Wang J. (2005). Correlation between seawater environmental factors and marine corrosion rate using artificial neural network analysis. *J. Chin. Soc. Corros. Prot*., Vol. 25, No. 2005, pp. 11–14.

Liu Y. X., Gao X. C., Zhang G. Y., Guo H. H. (2008). BP neural networks used in prediction and analyses of 3C steel corrosion function. *Journal of Materials Science and Engineering*, Vol. 26, No. 2008, pp. 94–97. https://doi.org/10.1080/00207540801918588

Paik J. K., Thayamballi A. K., Park Y. I., Hwang T. S. (2004). A time-dependent corrosion wastage model for seawater ballast tank structures of ships. *Corrosion Science*, Vol. 46, No. 2, pp. 471-486. https://doi.org/10.1016/S0010-938X(03)00145-8

Samide A. P., Bibicu I., Agiu M., Preda M. (2008). Mossbauer spectroscopy study on the corrosion inhibition of carbon steel in hydrochloric acid solution. *Materials Letters*, Vol. 62, No. 2, pp. 320-322. https://doi.org/10.1016/j.matlet.2007.05.025

Schwenker F., Kestler H. A., Palm G. (2001). Three learning phases for radial-basis-function networks. *Neural Networks*, Vol. 14, No. 4-5, pp. 439-458. https://doi.org/10.1007/s00213-006-0306-6

Song W. W., Dong C. C., Zhang B. (2012). Application of artificial neural network to metal corrosion in seawater. *Corrosion & Protection*, Vol. 33, No. 8, pp. 668-671.

Srisuwan N., Ochoa N., Pebere N., Tribollet B. (2008). Variation of carbon steel corrosion rate with flow conditions in the presence of an inhibitive formulation. *Corrosion Science*, Vol. 50, No. 5, pp. 1245-1250. https://doi.org/10.1016/j.corsci.2008.01.029

Yang X. S. (2010). Engineering optimization: an introduction with metaheuristic appplications. *John Wiley & Sons.* https://doi.org/10.1002/9780470640425