Korean J. Chem. Eng., **30**(12), 2197-2204 (2013)
DOI: 10.1007/s11814-013-0170-0

pISSN: 0256-1115
eISSN: 1975-7220

Modeling of corrosion reaction data in inhibited acid environment using regressions and artificial neural networks

Anees Abdullah Khadom†

Department of Chemical Engineering, College of Engineering, Diyala University,
Baquba City 32001, Diyala Governorate, Iraq

(*Received 7 March 2013 • accepted 4 September 2013*)

Abstract−This paper reports the results of mass loss measurements in the corrosion inhibition of mild steel in different concentrations of H_3PO_4 in the temperature range 30-60 °C using potassium iodide as an inhibitor. The present work is focused on determining the optimum mathematical equation and the ANN architecture in order to gain good prediction properties. Three mathematical equations and three ANN architectures are suggested. Computer aided program was used for developing these models. The results show that the polynomial mathematical equation and multi-layer perception are able to accurately predict the measured data with high correlation coefficients.

Key words: Corrosion, Computer Program, Neural Network, Mild Steel, Acid, Regression

INTRODUCTION

Mild steel can be protected against corrosion can be achieved by adding organic or inorganic compounds in small concentrations to the environment [1,2]. Many researchers have concentrated on the corrosion inhibition mechanism, adsorption isotherms, activation parameters, quantum chemistry, etc. [3-5]. Few of them reported the use of mathematical and statistical modeling. Mathematical modeling has already proven to be a very useful and powerful method in determining the relation between dependent and independent variables [6]. Another interesting technique for developing an input-output relationship is the artificial neural network [7-10]. Artificial neural networks (ANN) represent one of the fastest developing fields of artificial intelligence due to their ability to resemble (to a certain extent) the human problem solving characteristic, which is difficult to simulate using the logical, analytical techniques of expert system and standard software technologies [11,12]. The wide applicability of ANNs stems from their flexibility and ability to model linear and nonlinear systems without prior knowledge of an empirical model. This gives ANNs an advantage over traditional fitting methods for some chemical applications [13]. The ANN eliminates the limitations of the classical approaches by extracting the desired information using the input data. Applying ANN to a system requires sufficient input and output data instead of a mathematical equation. ANN can be trained using input and output data to adapt to the system. Also, ANN can be used to deal with the problems with incomplete and imprecise input data [14]. In the present work, the corrosion reaction of mild steel in 0.5, 1.5 and 2.5 M H_3PO_4, at 30, 40, 50, and 60 °C, in the presence of 0.02, 0.03, 0.04 and 0.05 M potassium iodide (KI) as a corrosion inhibitor was studied. The corrosion rate data were used as a source of mathematical and statistical regression.

†To whom correspondence should be addressed.
E-mail: aneesdr@gmail.com

EXPERIMENTAL WORK

The corrosion reaction of mild steel in 0.5, 1.5 and 2.5 M H_3PO_4 (pH of 2.2, 2, and 1.8), at 30, 40, 50, and 60 °C, in the presence of 0.02, 0.03, 0.04 and 0.05 M potassium iodide (KI) as a corrosion inhibitor was studied. The mild steel working electrode specimens have the following chemical compositions, C 0.041 wt%, Mn 0.311%, P 0.05%, S 0.007% and the remainder is iron. Specimens of rectangular shape with dimensions of 1 cm width and a length of 3 cm of mild steel were used. The specimens were first degreased with analar benzene and acetone, and then annealed in a vacuum at 600 °C for 1 hour, and cooled to room temperature. Samples were abraded in sequence under running tap water using emery paper of grade number 220, 320, 400 and 600, then washed with running tap water followed by distilled water, dried with a clean tissue, immersed in acetone and benzene, kept in desiccators over the silica gel bed until use. The dimensions of each sample were measured with a vernier to 2nd decimal of millimeter and accurately weighted to the 4th decimal of a gram. The metal samples were completely immersed each in 250 cm^3 solution of the corrodant contained in a conical flask. They were exposed for a period of 3 h at the desired temperature, acid concentration, and inhibitor concentration. Then the metal samples were cleaned, washed with running tap water followed by distilled water, dried with clean tissue and then immersed in acetone and benzene and dried again. Corrosion rates in $g \cdot m^{-2} \cdot day^{-1}$ (gmd) were determined in the presence and absence of inhibitor.

RESULTS AND DISCUSSION

1. Corrosion Rate Data

Table 1 summarizes the results of 60 test runs using weight loss technique of mild steel corrosion as a function of temperature and acid concentration in the absence and presence of potassium iodide as corrosion inhibitor. The value of corrosion rate was calculated from the following equation:

Table 1. Corrosion rate data (Y_{Exp} and Y_{ANN}) at different inhibitor concentration (X_1), different temperatures (X_2) and different acid concentration (X_3)

Run no.	X_1 (M)	X_2 (K)	X_3 (M)	Y_{Exp} (gmd)	Y_{ANN} (gmd)	Set (Train\Test)	Absolut error (%)
1	0	303	0.5	250.115	249.811	Test	0.121
2	0	313	0.5	1044.945	1048.398	Train	0.330
3	0	323	0.5	2091.705	2092.562	Test	0.041
4	0	333	0.5	3348.035	3368.558	Train	0.613
5	0.02	303	0.5	15.73	15.800	Train	0.445
6	0.03	313	0.5	29.16	30.643	Test	5.086
7	0.04	323	0.5	42.25	40.678	Test	3.721
8	0.05	333	0.5	95.185	97.617	Test	2.555
9	0.02	303	0.5	13.165	13.800	Train	4.823
10	0.03	313	0.5	16.31	16.643	Train	2.041
11	0.04	323	0.5	19.32	18.678	Test	3.322
12	0.05	333	0.5	56.125	58.617	Train	4.441
13	0.02	303	0.5	9.395	9.800	Test	4.311
14	0.03	313	0.5	9.91	9.643	Train	2.694
15	0.04	323	0.5	17.435	16.178	Train	7.209
16	0.05	333	0.5	54.3	55.617	Train	2.425
17	0.02	303	0.5	7.33	7.800	Test	6.412
18	0.03	313	0.5	7.9575	8.643	Test	8.614
19	0.04	323	0.5	8.485	9.678	Train	14.061
20	0.05	333	0.5	28.7	29.617	Train	3.195
21	0	303	1.5	350.161	347.155	Test	0.858
22	0	313	1.5	1462.923	1463.212	Test	0.019
23	0	323	1.5	2928.387	2900.725	Train	0.944
24	0	333	1.5	4687.249	4678.997	Test	0.176
25	0.02	303	1.5	22.022	21.366	Train	2.978
26	0.03	313	1.5	40.824	41.331	Test	1.241
27	0.04	323	1.5	59.15	61.191	Train	3.450
28	0.05	333	1.5	133.259	131.126	Train	1.601
29	0.02	303	1.5	18.431	17.366	Test	5.778
30	0.03	313	1.5	22.834	21.331	Test	6.582
31	0.04	323	1.5	27.048	28.191	Test	4.225
32	0.05	333	1.5	78.575	77.126	Test	1.844
33	0.02	303	1.5	13.153	12.066	Train	8.264
34	0.03	313	1.5	13.874	14.331	Train	3.293
35	0.04	323	1.5	24.409	23.191	Train	4.989
36	0.05	333	1.5	76.02	74.126	Train	2.491
37	0.02	303	1.5	10.262	9.366	Train	8.731
38	0.03	313	1.5	11.1405	11.331	Train	1.701
39	0.04	323	1.5	11.879	11.691	Train	1.582
40	0.05	333	1.5	40.18	41.126	Train	2.354
41	0	303	2.5	488.23	491.519	Train	0.673
42	0	313	2.5	1999.89	2007.518	Test	0.381
43	0	323	2.5	4003.41	4072.099	Test	1.715
44	0	333	2.5	6233.07	6113.978	Test	1.910
45	0.02	303	2.5	31.46	32.669	Test	3.842
46	0.03	313	2.5	58.32	56.256	Train	3.539
47	0.04	323	2.5	84.50	86.493	Train	2.358
48	0.05	333	2.5	190.37	193.545	Train	1.667
49	0.02	303	2.5	26.33	25.669	Train	2.510
50	0.03	313	2.5	32.62	33.256	Test	1.949
51	0.04	323	2.5	38.64	37.493	Test	2.968

Table 1. Continued

Run no.	X_1 (M)	X_2 (K)	X_3 (M)	Y_{Exp} (gmd)	Y_{ANN} (gmd)	Set (Train\Test)	Absolut error (%)
52	0.05	333	2.5	112.25	111.545	Train	0.628
53	0.02	303	2.5	18.79	19.069	Test	1.484
54	0.03	313	2.5	19.82	21.156	Test	6.740
55	0.04	323	2.5	34.87	35.493	Test	1.786
56	0.05	333	2.5	108.60	109.545	Test	0.870
57	0.02	303	2.5	14.66	13.669	Test	6.759
58	0.03	313	2.5	15.915	15.256	Test	4.140
59	0.04	323	2.5	16.97	16.493	Test	2.810
60	0.05	333	2.5	57.40	58.545	Train	1.994

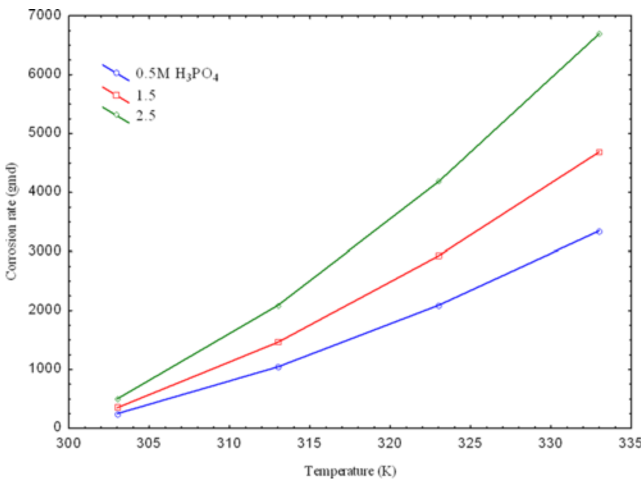

Fig. 1. Variation of corrosion rate with temperature at different acid concentration and absence of KI.

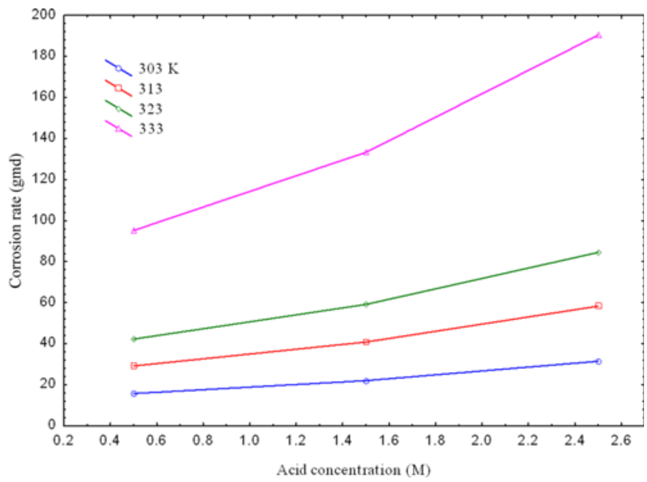

Fig. 3. Variation of corrosion rate with acid concentration at different temperature and 0.02 M inhibitor concentration.

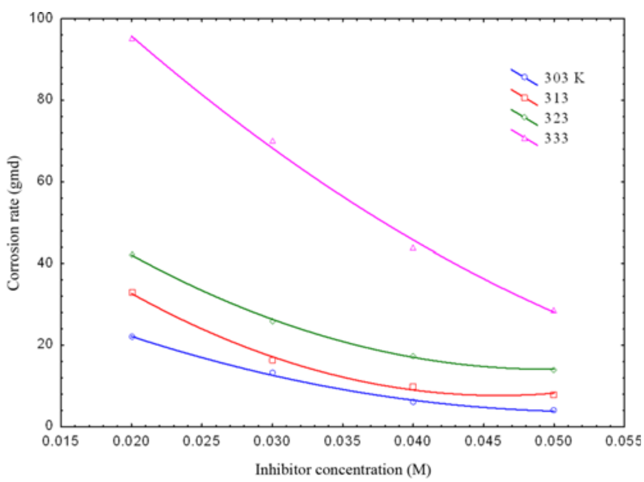

Fig. 2. Variation of corrosion rate with inhibitor concentration at different temperature and 0.5 M acid concentration.

$$\mathrm{CR} = \frac{\text{weight loss (g)}}{\text{area (m}^2) \times \text{time (day)}} \tag{1}$$

Table 1 and Figs. 1, 2 and 3 show the behavior of corrosion rate with the research operation conditions. Corrosion rate increased with temperature and acid concentration. While, it decreased with the addition of inhibitor. In Fig. 1, in the absence of inhibitor, the corrosion rate increased with temperature at different acid concentrations. Similar behavior was observed in the case of addition of KI. Fig. 2 illustrates that the corrosion rate was reduced to a significant level as the inhibitor concentration increased from 0 to 0.05 M at different temperatures and 0.5 M H_3PO_4 concentration. The same figures can be obtained at 1.5 and 2.5 M H_3PO_4. The effect of acid concentration is shown in Fig. 3. The corrosion rate increased with acid concentration at different temperatures and 0.02 M KI. This behavior can be seen at other inhibitor concentrations. For purposes of comparison, Table 1 also lists the corrosion rate values predicted by the application of optimum ANN (Y_{ANN}) against the experimental one (Y_{Exp}), which indicates a good correlation between the results as will be discussed later. The average of percentage absolute error between experimental corrosion rates and predicated by ANN was 3.1%.

2. Mathematical Modeling and Statistical Analysis

Three mathematical models were suggested to represent the corrosion rate data. The kinetic model was derived depending on Mathur and Vasudevan discussions [15] and on the basis that the corrosion rate decreased with inhibitor concentration increasing.

$$Y = a_0 X_1^{-a_1} e^{\frac{-a_2}{X_2}} e^{a_3 X_3} \tag{2}$$

The first is the exponential model of Khadom et al., which was suggested successfully in our previous works [16,17]:

Table 2. The constants of estimated models and correlation coefficients

Model	Equations	*R^2	RMSE	MAPE
Kinetic	$Y=5.339X_1^{-1.837}e^{\frac{-6360.3}{X_2}}e^{0.35X_3}$	0.9952	2.14	1.18
Exponential	$Y=7.998X_1^{-1.84}.e^{\frac{-6361}{X_2}}.X_3^{0.452}$	0.9929	3.33	2.52
Polynomial	$Y=211905-3.2\times10^7X_1-1.15\times10^8X_1^2+1.13\times10^5X_1X_2$ $-2.6\times10^4X_1X_3-1429X_2+32X_2X_3-9335X_3$	0.9961	1.27	0.67
ANN train	MLP3:3-6-1:1	0.9873	4.47	0.032
ANN test	MLP3:3-6-1:1	0.9781	9.21	0.035

*R^2: correlation coefficient, RMSE: root mean square error, MAPE: mean absolute percentage error

$$Y=a_0X_1^{-a_1}.e^{\frac{-a_2}{X_2}}.X_3^{a_3} \quad (3)$$

This model was derived from the basis that the corrosion rate results can be related to temperature by Arrhenius type equation [18], and the corrosion rate decreases as the inhibitor concentration increases and as acid concentration decreases. The second-order polynomial model was also suggested. This model takes into account the individual effect of each variable and the interaction between them.

$$Y=a_0+a_1X_1+a_2X_1^2+a_3X_1X_2+a_4X_1X_3+a_5X_2+a_6X_2^2+a_7X_2X_3+a_8X_3+a_9X_3^2 \quad (4)$$

where Y is corrosion Rate (gmd), X_1 is inhibitor concentration (M), X_2 is absolute temperature (K), X_3 is acid concentration (M) and, a_0, a_1, a_2, … a_9 are constants. STATISTICA 7 software was used to estimate the coefficients of these models. This software is based on the Levenberg-Marquardt non-linear estimation least squares method. The numerical values of these coefficients are shown in Table 2. These equations represent the corrosion rate data with high correlation coefficients R^2. Furthermore, the value of the root mean square error (RMSE) and mean absolute percentage error (MAPE) were acceptable. The smaller the difference between RMSE and MAPE, the smaller the variance in the individual error in the sample. As shown in Fig. 4, the best fitting was obtained with a polynomial model. The coefficient a_2 in Eqs. (2) and (3) corresponds to the factor E/R (activation energy/gas constant) in the Arrhenius equation. The values of a_2 in Eqs. (2) and (3) are 6360.3 and 6361, respectively.

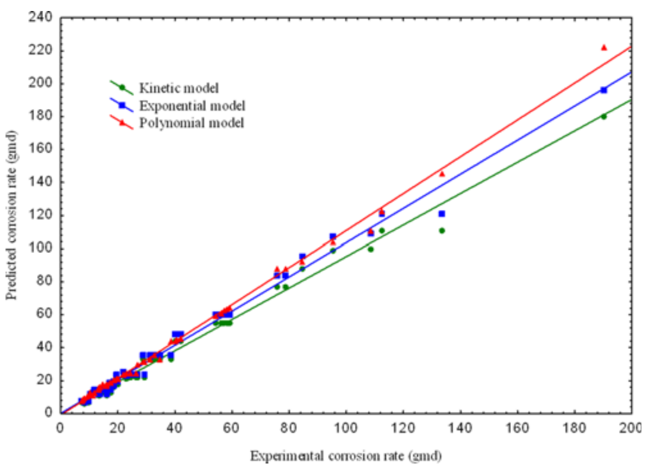

Fig. 4. Experimental against predicted corrosion rate from mathematical regression.

These values yield an activation energy of 52.87 and 52.88 kJ/mol, which agrees with the activation energy values for the corrosion of steel in acidic media [19,20]. The analysis of variance (F-test) can be used for testing the significance of each effect in Eq. (4). The calculations are given in Table 3. An estimate of the variance S_b^2 is obtained by dividing the experimental error variance S_r^2 by the sum of squares of each effect $\sum X^2$, as follows: $S_b^2=S_r^2/\sum X^2$, where, $S_r^2=\sum e^2/\gamma$, and γ=N−n, while F-values=a^2/S_b^2 [17]. The significance of the effects may be estimated by comparing the values of the ratio (a^2/S_b^2) with the critical value of the F-distribution at 95% confidence level ($F_{0.95}$=6.61). If the ratio of a^2/S_b^2>6.61, then the effect is significant. Thus, according to the results shown in Table 3, most of the individual and interaction effects of variables are significant and must be taken into account.

3. Artificial Neural Network (ANN) Analysis

3-1. ANN Data and Description

The corrosion rate data were used as feed for constructing the ANN. An artificial neural network is an intelligent data-driven modeling tool that is able to capture and represent complex and nonlinear input/output relationships. They simulate the learning process of the human brain. Like the brain, the network structure composed of several processing elements is called neurons or nodes [21]. These neurons, which are located in the structure, are highly interconnected with each other according to the type of the neural network. Connections can be made in several ways. Training algorithm and the structure of the network change with the changing of connection types. All ANNs have a similar topological structure or architec-

Table 3. F-test analysis

Effect	$\sum X^2$	Coefficient, a	Variance S_b^2	F-value	$F_{0.95}$=6.61	
		a_0	211905			
X_1	1.68	a_1	-3.2×10^7	25×10^{-8}	1.7×10^{17}	S
X_2	19080	a_2	-1.15×10^8	28×10^{-4}	7.3×10^{17}	S
X_3	90	a_3	1.13×10^{17}	13.4×10^{-4}	1.8×10^{21}	S
X_1X_2	540.24	a_4	-2.6×10^4	8×10^{-5}	32×10^{13}	S
X_1X_3	2.52	a_5	−1429	1×10^{-8}	1.8×10^{15}	S
X_2X_3	28620	a_6	2	0.004	4.7×10^8	S
X_1^2	0.0648	a_7	32	1×10^{-9}	5.9×10^8	S
X_2^2	37×10^7	a_8	−9335	55.335	1.37	NS
X_3^2	8275	a_9	36	0.001	5.33	NS

S: significant, NS: not significant

ture [22]. Generally, there are three layers—an input layer, which receives information from the external world, a hidden layer, which processes the information and an output layer which presents the output to the external world. The arrangement of neurons in each layer is entirely dependent on the user, which depends on the problem to be modeled and studied. The network learns all the patterns at the end of training and then the network is tested for its performance using patterns that were not used for training.

3-2. ANN Models

The simplest form of ANN is the *linear model* (LM). A neural network with no hidden layers, and an output with dot product synaptic function and identity activation function, actually implements a linear model. The weights correspond to the matrix, and the thresholds to the bias vector. When the network is executed, it effectively multiplies the input by the weights matrix and then adds the bias vector. The linear network provides a good benchmark against which

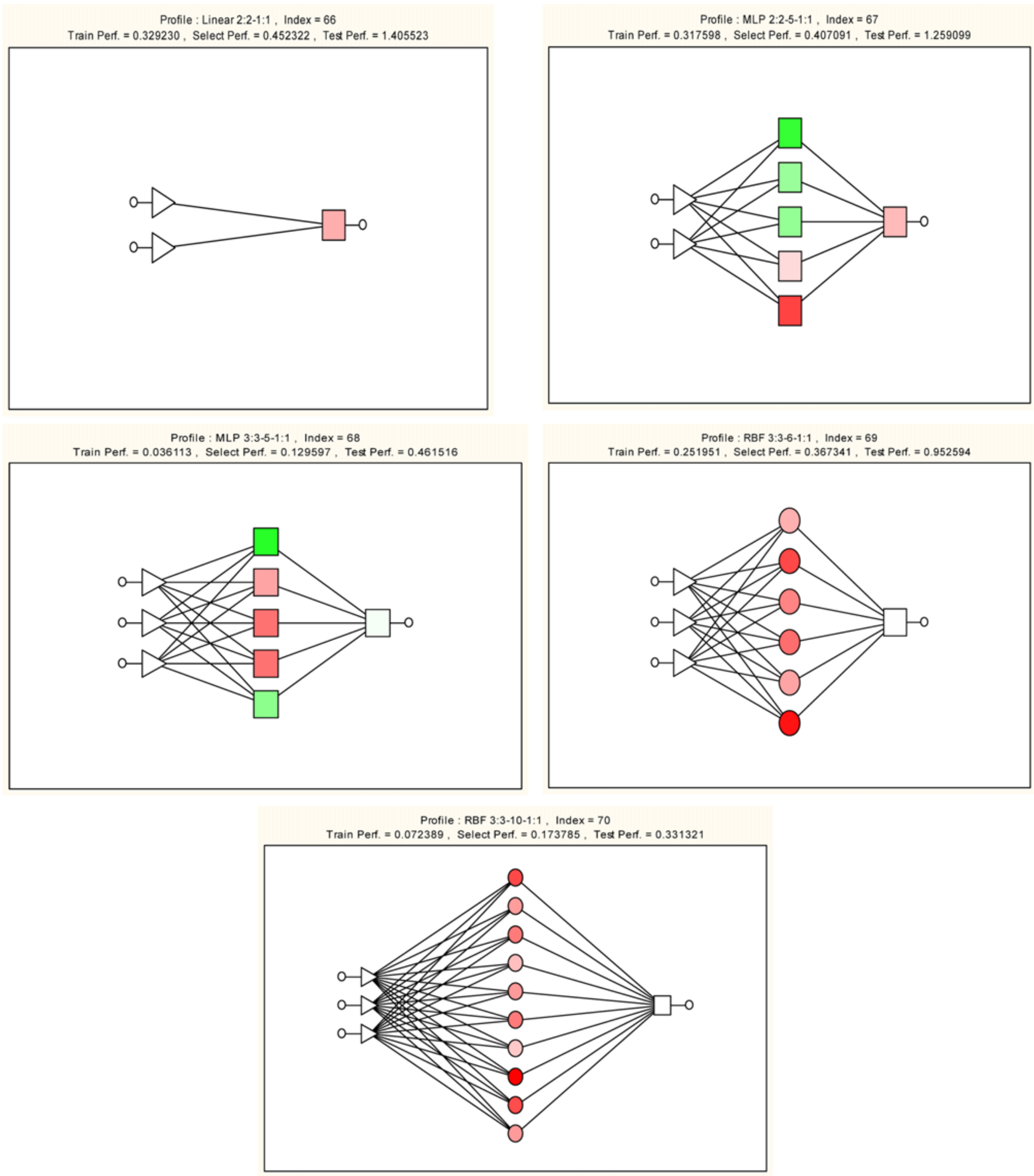

Fig. 5. ANN created by IPS.

to compare the performance of neural networks. The *multi-layer perceptrons* (MLP) network trained using back-propagation (BP) algorithm is a widely used network type and is commonly applied to all kinds of industrial as well as research modeling problems. *A radial basis function (RBF) neural network is* a new class of robust neural network that has been used to a limited extent in modeling various research problems [23].

3-3. ANN Construction by *Intelligent Problem Solver*

The neural network model presented in this study was created using *STATISTICA 7 software package*, which is a comprehensive, state-of-the-art, powerful, and extremely fast neural network data analysis package. This feature has different options and sub-software. One of them is an *intelligent problem solver* (IPS). It is a unique wizard-style that can be used as a guide through a step by step procedure of creating a variety of different networks and choosing the network with the best performance. Fig. 5 shows the networks created by IPS. Unit activation levels are (by default) displayed in color: red for positive activation levels, green for negative. *Triangles* pointing to the right indicate input neurons. These neurons perform no processing and simply introduce the input values to the network. *Squares* indicate dot product synaptic function units (e.g, as found in MLP). *Circles* indicate radial synaptic function units. Small open circles that represent input and output variables are illustrated using a small open circle joined to the corresponding input or output neuron. In some circumstances (nominal variables and time series inputs) a number of neurons are joined to a single input or output variable. These networks were constructed using different activation functions, such as sigmoid, hyperbolic, exponential, step, ramp, sine, square root, etc. IPS selects the best activation function for network building. Artificial neuron receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection that has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the activation of the neuron (also known as the post-synaptic potential, or PSP, of the neuron). The activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron. If the step activation function is used (i.e., the neuron's output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0), then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Actually, the step function is rarely used in artificial neural networks. It is also noticeable that weights can be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain. If a network is to be of any use, there must be inputs (which carry the values of variables of interest in the outside world) and outputs (which form predictions, or control signals). Inputs and outputs correspond to sensory and motor nerves such as those coming from the eyes and leading to the hands. However, there also can be hidden neurons that play an internal role in the network. The input, hidden, and output neurons need to be connected together. The key issue here is feedback [24]. A simple network has a feedforward structure: signals flow from inputs, forward through any hidden units, eventually reaching the output units. Such a structure has stable behavior. However, if the network is recurrent (contains connections back from later to earlier neurons), it can be unstable and have very complex dynamics. Recurrent networks are very interesting to researchers in neural networks, but so far it is the feedforward structures that have proven most useful in solving real problems, and it is these types of neural networks that present work models. A typical feedforward network has neurons arranged in a distinct layered topology. The input layer is not really neural at all: these units simply introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better; thus this point was taken into account. When the network is executed (used), the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

3-4. ANN Selection

Among the decisions that the neural network designer must make, which of the available variables to use as inputs to the neural network (independent variables) is one of the most difficult. The only method that is guaranteed to select the best input set is to train networks with all possible input sets and all possible architectures, and to select the best. In practice, this is impossible for any significant number of candidate inputs. Some highly sophisticated algorithms are contained to select input variables, and for most users this will provide a reasonable analysis. Table 4 shows the analysis of each net ANN. Index represents a unique life-long number assigned to each neural network when it is created. The indices are assigned in chronological order. Profile is the most useful summary statistic, packing a great deal of information into a short piece of text. It tells us the network type, the number of input and output variables, the num-

Table 4. ANN models summary

Index	Profile	Train perf.	Select perf.	Test perf.	Train error	Select error	Test error
66	Linear 2:2-1:1	0.32923	0.45232	1.40552	0.07795	0.21899	0.09196
67	MLP2:2-5-1:1	0.31759	0.40709	1.25909	0.07520	0.19830	0.08107
68	MLP3:3-6-1:1	0.03611	0.12959	0.46151	0.00855	0.06186	0.02952
69	RBF3:3-6-1:1	0.25195	0.36734	0.95259	0.00024	0.00071	0.00025
70	RBF3:3-10-1:1	0.07238	0.17378	0.33132	0.00007	0.00034	0.00008

ber of layers, and the number of neurons in each layer. The format is <type> <inputs>:<layer1>-<layer2>-<layer3>:<outputs>, where the number of layers may vary. For example, the profile MLP 2:2-5-1:1 signifies a multilayer perceptron with two input variables and one output variable, and three layers of 2, 5, and 1 units, respectively. For simple networks the number of input variables and output variables may match the number of neurons in the input and output layers, but this is not always so, as we shall see. Train Perf/Select Perf/Test Perf: these columns give the performance of the networks on the training, selection, and test subsets, respectively. One should not give too much credence to the performance rate reported on the training set, which is often deceptively good (indicating over-learning). Also, one should avoid using the test set performance to select models, as that defeats the object of having it, which is to maintain some data not used for training or model selection, so that a dispassionate final assessment of performance can be made. The performance measure on the selection subset should be used to discriminate between, and choose between, networks. The meaning of the performance measure depends on the network type. It is the ratio of the prediction to observation standard deviations. Train Error/Select Error/Test Error columns report the error rates on the subsets. The error rate is less directly interpretable than the performance measure, but is of more significance to the training algorithms themselves. Fig. 6 shows the predicted corrosion rate against experimental one; the best results are obtained by MLP 3:3-5-1:1. Furthermore, Table 5 shows some statistics for ANN. Maximum correlation coefficient and minimum error standard deviation, absolute error and standard deviation ratio were with the network of index 68 (i.e., MLP 3:3-5-1:1).

3-5. ANN Sensitivity Analysis

The sensitivity analysis gives some information about the relative importance of the variables used in a neural network. In sensitivity analysis, the *IPS* tests how the neural network would cope if each of its input variables were unavailable. In sensitivity analysis, the data set is submitted to the network repeatedly, with each variable in turn treated as missing, and the resulting network error is recorded. If an important variable is removed in this fashion, the error will increase a great deal; if an unimportant variable is removed, the error will not increase very much. Table 6 indicates the sensitivity of the model to each variable. The sensitivity is reported in two rows: the *Ratio* and the *Rank*. The basic sensitivity figure is the *Ratio*. For each variable, the network is executed as if that variable is "unavailable." Unavailability of a variable used by the model will presumably cause some deterioration in its performance. The ratio reported is the ratio of the error with the variable unavailable to the ratio with its availability. Important variables have a high ratio, indicating that the network performance deteriorates badly if they are not present. If the *Ratio* is one or lower, then making the variable "unavailable" either has no effect on the performance of the network, or actually enhances it. The *Rank* lists the variables in order of importance (i.e. order of descending *Ratio*), and is provided for convenience in interpreting the sensitivities. In the present work data, the X_1 (inhibitor concentration) is the most sensitive variables, with lower effect for X_2 (tem-

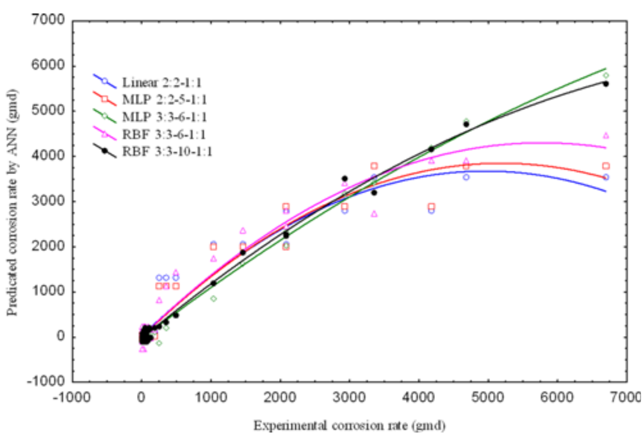

Fig. 6. Experimental corrosion rate against predicted by ANN.

Table 5. ANN sensitivity analysis

	X_1 (M)	X_2 (K)	X_3 (M)
Ratio. 66	2.43	2.03	
Rank. 66	1	2	
Ratio. 67	2.61	2.12	
Rank. 67	1	2	
Ratio. 68	9.1	7.41	2.64
Rank. 68	1	2	3
Ratio. 69	3.16	2.57	1.25
Rank. 69	1	2	3
Ratio. 70	6.99	5.48	2.18
Rank. 70	1	2	3

Table 6. Statistical and regression data

	Y. 66	Y. 67	Y. 68	Y. 69	Y. 70
Data mean	525.47	525.47	525.47	525.47	525.47
Data S. D.	1291.49	1291.49	1291.49	1291.49	1291.49
Error mean	–12.25	–22.01	–16.68	22.81	–7.09
Error S. D.	546.55	498.86	144.36	423.67	184.41
Abs. error mean	220.11	203.65	72.57	243.38	95.49
S. D. ratio	0.42	0.386	0.112	0.328	0.143
Correlation coff.	0.913	0.928	0.994	0.946	0.990

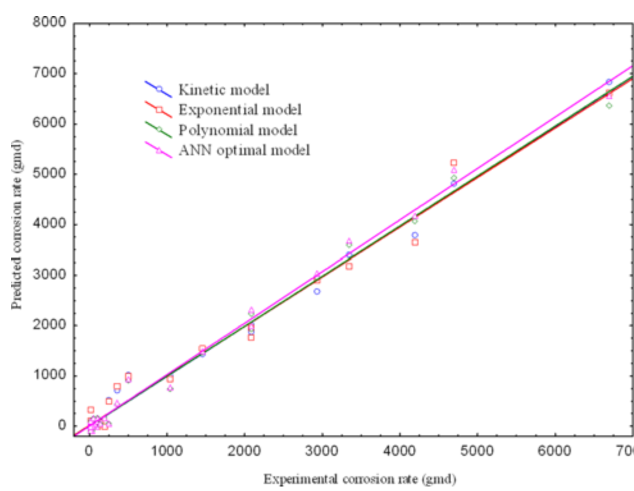

Fig. 7. Experimental against predicted corrosion rate from kinetics, exponential, polynomial and ANN optimal model.

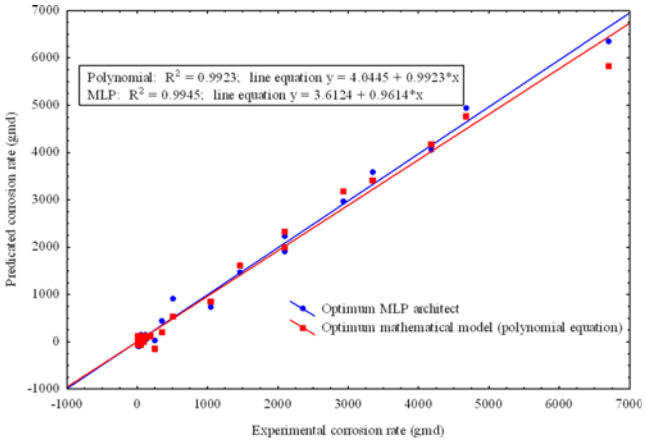

Fig. 8. Experimental against predicted corrosion rate from mathematical optimum model and ANN optimum model.

perature) and X_3 (acid concentration), respectively. The same behavior was observed with mathematical polynomial model (Table 1); the effects of X_1 coefficients were more significant than the others.

OPTIMUM REGRESSION AND ESTIMATED MODEL

Mathematical statistical and ANN analysis showed that both techniques represent the corrosion rate data in a powerful and successful way. Fig. 7 compares the three mathematical models with optimum ANN model. While Fig. 8 shows the optimum mathematical polynomial model as compared with optimum ANN model. This indicates that both polynomial equations (from mathematical analysis) and MLP with three input layers, five hidden layers and one output layer (from ANN analysis) predicted the corrosion rate values with higher correlation coefficients.

CONCLUSIONS

The following conclusions can be drawn from the current study:

- The results obtained from weight loss measurements indicate that corrosion of mild steel in phosphoric acid solution decreased with inhibitor concentration increase and increased with temperature and acid concentration increase.
- An attempt has been made to use mathematical regression, statistical analysis, and ANNs to model the corrosion processes of mild steel as a function of inhibitor concentration, temperature and acid concentration.
- For mathematical studies, all suggested equations successfully represented the corrosion rate data.
- In ANN studies, MLP with three input layer, five hidden layers and one output layer was the better architect than LM and RBF models.

ACKNOWLEDGEMENT

This work was supported by Baghdad University, Chemical Engineering Department, which is gratefully acknowledged. Also special thanks to Mrs. Areej S. Dawood for her assistance during this study.

REFERENCES

1. A. A. Khadom, A. Y. Musa, A. H. Kadhum, A. B. Mohamad and M. S. Takriff, *Portugaliae Electrochim. Acta*, **28**, 221 (2010).
2. A. S. Yaro and A. A. Khadom, *Inter. J. Surf. Sci. Eng.*, **4**, 429 (2010).
3. A. A. Khadom, A. S. Yaro and A. H. Kadhum, *J. Chilean Chem. Soc.*, **55**, 150 (2010).
4. A. A. Khadom, A. S. Yaro, A. S. Altaie and A. H. Kadhum, *Portugalia Electrochem. Acta*, **27**, 699 (2009).
5. A. Y. Musa, A. H. Kadhum, A. B. Mohamad, M. S. Takriff, A. R. Daud and S. K. Kamarudin, *Corros. Sci.*, **52**, 526 (2010).
6. A. A. Khadom, A. S. Yaro, A. S. Altaie and A. H. Kadhum, *J. Appl. Sci.*, **9**, 2457 (2009).
7. M. Tatlier, H. K. Cigizoglu and A. Erdem-Şenatalar, *Comput. Chem. Eng.*, **30**, 137 (2005).
8. V. Dua, *Comput. Chem. Eng.*, **35**, 545 (2011).
9. A. M. Rashidi, *J. Mater. Sci. Technol.*, **28**, 1071 (2012).
10. K. Song, J. Xing, Q. Dong, P. Liu, B. Tian and X. Cao, *Mater. Des.*, **26**, 337 (2005).
11. H. Moral, A. Aksoy and C. F. Gokcay, *Comput. Chem. Eng.*, **32**, 2471 (2008).
12. I. Fahmi and S. Cremaschi, *Comput. Chem. Eng.*, **46**, 105 (2012).
13. A. L. Ahmad, I. A. Azid, A. R. Yusof and K. N. Seetharamu, *Comput. Chem. Eng.*, **28**, 2709 (2004).
14. S. V. Sousa, F. G. Martins, M. C. Alvim-Ferraz and M. C. Pereira, *Environ. Model. Software.*, **22**, 97 (2007).
15. P. B. Mathur and T. Vasudevan, *Corrosion*, **38**, 171 (1982).
16. A. A. Khadom, A. S. Yaro and A. H. Kadhum, *J. Taiwan Inst. Chem. Eng.*, **41**, 126 (2010).
17. A. S. Yaro, H. Al-Jendeel and A. A. Khadom, *Desalination*, **270**, 193 (2011).
18. I. B. Obot and N. O. Obi-Egbedi, *Corros. Sci.*, **52**, 198 (2010).
19. A. Kosari, M. Momeni, R. Parvizi, M. Zakeri, M. H. Moayed, A. Davoodi and H. Eshghi, *Corros. Sci.*, **53**, 3058 (2011).
20. T. Poornima, J. Nayak and A. Nityananda Shetty, *Corros. Sci.*, **53**, 3688 (2011).
21. N. Birbilis, M. K. Cavanaugh, A. D. Sudholz, S. M. Zhu, M. A. Easton and M. A. Gibson, *Corros. Sci.*, **53**, 168 (2011).
22. Z. Zhang and K. Friedrich, *Compos. Sci. Technol.*, **63**, 2029 (2003).
23. J. A. Platt, *Neural Comput.*, **3**, 213 (1991).
24. S. Haykin, *Neural networks: A comprehensive foundation*, Macmillan Publishing, New York (1994).