AN ARTIFICIAL NEURAL NETWORK MODEL FOR THE PREDICTION OF COMPRESIBILITYFACTOR FOR GENERALIZED EQUATION OF STATE

M.Ravi Kumar[1], Ermias Girma Aklilu[2]

[1]Department of Chemical Engineering, College of Engineering And Technology, Samara University, Samara, Afar, Ethiopi

[2]Department of Chemical Engineering, College of Engineering And Technology, Samara University, Samara, Afar, Ethiopi

Abstract

In recent years, modeling using Artificial Neural Networks is attracting a lot of attention among scientists and engineers, and is being hailed as one of the greatest computational tools ever developed. This is due to the apparent ability of Neural Networks to emulate the brains ability to learn by examples, which in turn enables the networks to make decisions and draw conclusions when presented with complex, noisy and/or incomplete information. The functional relationship among reduced pressure, compressibility, accentric factor, and reduced temperature of a gas is known as generalized equation of state (EOS). Though several generalized equations of state are available in literature each equation has its inherent disadvantages. There is no single generalized equation of state, which is mathematically friendly and can satisfy the experimental compressibility data with engineering accuracy. Hence a two parameter of generalized EOS model is developed using Artificial Neural Network in the present work. A feed forward network with back propagation of errors algorithm is used in training the network. In this work the supervised learning, sum squared error is used as lapnov function. The model is found to be in good agreement with the experimental results.

Keywords: *Artificial Neural Network, Compressibility Factor, Sum Squared Error, Acentric Factor.*

--***--

1. INTRODUCTION

An Artificial Neural Network is a computational structure made up of a number of simple, highly interconnected nodes or processing elements, which processes information by its dynamic state response to external inputs.

The goal of Artificial Neural Network is to map a set of input patterns on to a corresponding set of output patterns. The network accomplishes this mapping by first learning from a series of examples defining sets of input and output correspondences for the given system. The network then applies the learning over past experience to predict the output for the new input also which is not identical with any past experience but has got similarity with some of the past inputs. Hence ANN (Artificial Neural Network) is said to be intelligent. These systems are quite interesting due, not only to their ability for both learning automatically and working with inaccurate information or with failures in their components.

An ANN is a massively parallel-distributed information-processing system that has certain performance characteristics resembling biological neural networks of the human brain. ANN has been developed as a generalization of mathematical models of human cognition and neural biology. Their development is based on the rules that.

1. Information processing occurs at many elements called nodes, also referred to as units, cells or neurons.
2. Signals are passed between nodes through connection links.
3. Each connection link has an associated weight that represents its connection strength.
4. Each node typically applies a nonlinear transformation called activation function to its net input to determine its output signal.

ANN is a sort of parallel computer/processor designed to imitate the way the brain accomplishes a certain task [Willis, et al, 1991]. The smallest processing element of ANN is a neuron (also called node), which performs simple calculations. Using the nodes collectively with massive connections among them results in a network that is able to process and store information for mapping the network inputs to its outputs. With this capability, there are widespread interests due to on-going and potential applications in solving complex problems particularly in the fields of pattern recognition (especially in speech and image processing), classification, control, forecasting, systems identification and optimization. As the term artificial neural networks implies, early work in the field of neural networks centered on modeling the behavior of neurons found in the human brain. Engineering systems are considerably less complex than the brain, hence from an engineering view-point ANN can be viewed as nonlinear empirical models that are especially useful in representing input-output data, making predictions in time, classifying data, and recognizing patterns.

In the present work, a two parameter of generalized equation of state model is developed using Artificial Neural Network. the supervised learning, sum squared error is used as lapnov function. The model is found to be in good agreement with the experimental results.

1.1 Compressibility Factor (Z) Theory

1.1.1 Virial Equation

A useful auxiliary thermodynamic property is defined by the equation.

$$Z \equiv \frac{PV}{RT} \tag{1}$$

This dimensionless ratio is called *compressibility factor*.

$$Z = 1 + B'P + C'P^2 + D^{'P^3} + \ldots. \tag{2}$$

An alternative expression for Z is also in common use.

$$Z = 1 + \frac{B}{V} + \frac{C}{V^2} + \frac{D}{V^3} + \ldots \tag{3}$$

Both of these equations are known as *virial expansions*, and the parameter B', C', D', etc., and B, C, D, etc., are called *virial coefficients*. Parameters B' and B are *second* virial coefficients; C' and C are *third* virial coefficients; etc, For a given gas the virial coefficients are functions of temperature only.

The two sets of coefficients is 2 and 3 are related as follows;

$$B' = \frac{B}{RT} C' = \frac{C - B^2}{(RT)^2} D' = \frac{D - 3BC + 2B^2}{(RT)^3} \qquad \text{etc.}$$

1.1.2 Ideal Gas

Since, the terms B/V, C/V^2, etc., of the virial expansion [Eq. (3)] arise on account of molecular interactions, the virial coefficient B, C, etc., would be zero if no such interactions existed. The virial expansion would then reduce to :

Z = 1 or PV = RT

For a real gas, molecular interactions do exist, and exert an influence on the observed behavior of the gas. As the pressure of a real gas is reduced at constant temperature, V increases and the contributions of the terms B/V, C/V^2, etc., decrease. For a pressure approaching zero, Z approaches unity, not because of any change in the Virial coefficient, but because V becomes infinite. Thus in the limit as the pressure approaches zero, the equation of state assumes the same simple form as for the hypothetical case of B = C = …….. = 0; i.e., Z = 1 or PV = RT

Differentiation of Eq.(2) for a given temperature gives:

$$\left(\frac{\partial Z}{\partial P}\right)_T = B' + 2C'P + 3D^{'P^2} + \ldots. \tag{4}$$

From which, $\left(\frac{\partial Z}{\partial P}\right)_{T;P=0} = B'$

Thus the equation of tangent line is:Z = 1 + BP'a result also given by truncating the Eq. 2 to two terms. A more common form of this equation results from the substitution, B' = B/RT;

$$Z = \frac{PV}{RT} = 1 + \frac{BP}{RT} \tag{5}$$

Equation (3) may also be truncated to two terms for application at low pressure:

$$Z = \frac{PV}{RT} = 1 + \frac{B}{V} \tag{6}$$

and virial equation is truncated to three terms, the appropriate form is:

$$Z = \frac{PV}{RT} = 1 + \frac{B}{V} + \frac{C}{V^2} \tag{7}$$

1.2 Pitzer Correlations for the Compressibility Factor

The correlation for Z takes the form:

$$Z = Z^0 + \omega Z^1 \tag{8}$$

Where Z^0 and Z^1 are functions of both T_r and P_r. When ω = 0, as is the case for the simple fluids, the second term disappears, and Z^0 becomes identical with Z. Thus a generalized correlation for Z as a function of T_r and P_r based on data for just argon, krypton, and xenon provides the relationship $Z^0 = F^0(T_r, P_r)$. By itself, this represents a two-parameter corresponding states correlation for Z. Since the second term of Eq. (8) is a relatively small correlation to this correlation, its omission does not introduce large errors, and a correlation for Z^0 may be used alone for quick but less accurate estimates of Z than are obtained from a three parameters correlation.

Equation (8) is simple linear relation between Z and ω for given values of T_r and P_r do indeed yield approximately straight line, and their slopes provide values for Z^1 from which the generalized function $Z^1 = F^1(T_r, P_r)$ can be constructed.

The simplest form of the Virial equation has validity only to low to moderate pressures where Z is linear in pressure. The generalized Virial-coefficient correlation is therefore useful only where Z^0 and Z^1 are at least approximately linear functions of reduced pressure.

Levenberg Marquardt Method

This ANN design model uses Levenberg Marquardt method as multi variable optimization technique to predict the sum of squared errors for the designed target value.

Neural Network Architecture

A neural network is characterized by its architecture that represents the pattern of connection between the nodes, its method of determining connection weights and the activation function. A typical ANN consists of a number of nodes that are organized according to a particular arrangement. One way of classifying a neural network is by the number of layers.

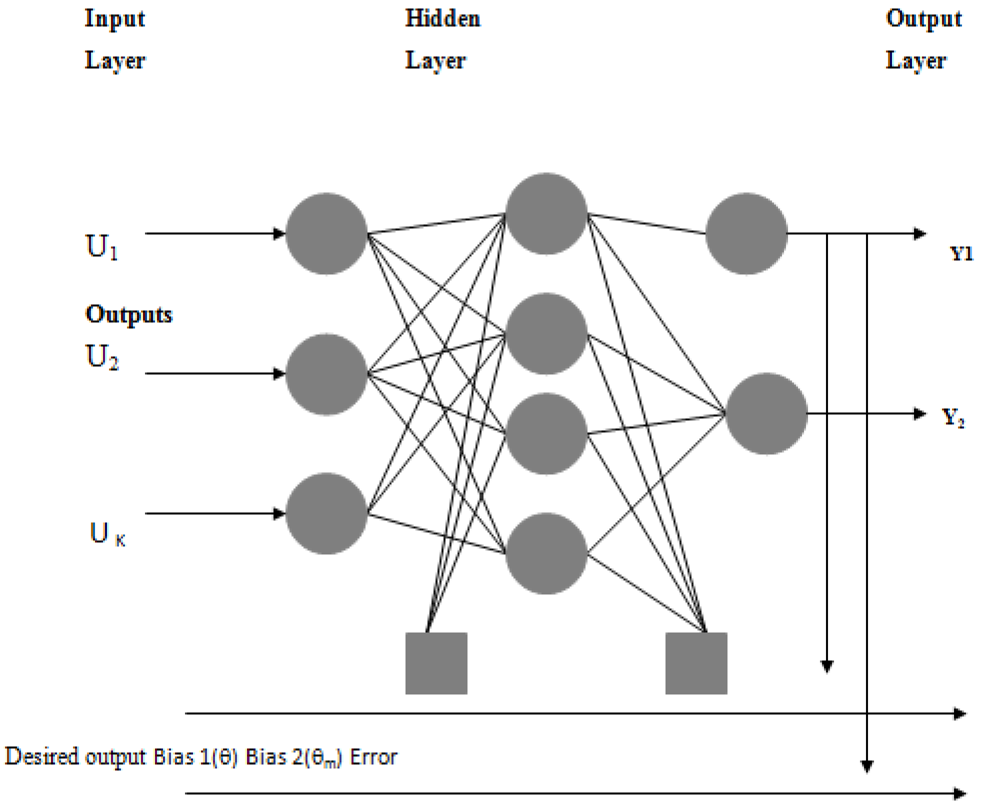
1. Single layer – Hop field et al.
2. Bilayer – Grosberg adaptive resonance network.
3. Multi layer – most back propagation networks.

Input layer: The input layer receives the input variables for the problem at hand. This consists of all the quantities that can influence the output. The input layer is thus transparent and is a means of providing information to the network.

Hidden layer: The layers processing information are in between the input and output layers, which has any number of nodes.

Output layer: This layer gives the output of the network to external receptor.

Each layer consists of certain number of nodes, which are connected by links. No two nodes in the same layer will be having any links. A synaptic weight is assigned to each link to represent the relative connection strength of two nodes at both ends in predicting the input-output relationship.



Fundamentals of Neural Computations

A node/neuron/processing element is an abstraction for a computational procedure, consists of three steps.

1. Calculating the weighted sum of input (net input) signals.
2. Applying a threshold (the value which excites or inhibits the nodes firing) on the weighted sum of input signals.
3. Calculating the output according to particular output function called transfer function or activation function.

Weight factor is nothing but the strength of an interconnection between two neurons expressed numerically.

Transfer functionis nothing but the mapping relation between the output of node and its threshold applied weighted sum of input signals.

Examples of threshold functions are

- Radial basis transfer function: f(x) = $e^{\frac{-x^2}{2}}$
- Logarithmic sigmoid transfer function: (log sig) : f(x) = $\frac{1}{(1+e^{-x})}$

Normalizing the input-output data: The normalization of input and output data is very critical for successful training of network. If input and output variables do not have same order of magnitude some variables may appear to have more significance than they actually do. The training algorithm has to compensate for this and this is not very effective in many training algorithms. And also transfer functions like sigmoid and hyperbolic tangent are insensitive as argument (threshold input) is greater than 2. Hence three commonly used normalizations are given here.

$$X_{1\ norm=}\left(\frac{X_i}{X_{i\ max}}\right);\ X_{i\ norm} = \frac{(X_i - X_{i\ min})}{(X_{i\ max} - X_{i\ min})};$$

Zero mean Normalization method: $X_{i\ norm} = \frac{(X_i - X_{i\ avg})}{R_{i\ max}};$

Where $R_{i\ max}$= Max ($X_{i\ max}$ – $X_{i\ avg}$, $X_{i\ avg}$ - $X_{i\ min}$); $X_{i\ avg} = \frac{\sum_1^n X_i}{n};$

Initializing the weight distribution: -One broad suggestion for initializing weights is take random weight factors in between –1/n to 1/n, where n is number of neurons in input layer.

Setting learning rate and momentum coefficient: - Learning rate and momentum coefficient are two very important parameters that control how effectively the back propagation algorithm trains the neural network. The learning rate is a positive constant and that controls the rate at which the new weight factors are adjusted based on the calculated gradient descent correction term. The momentum coefficient is extra weight added on to the weight factors that accelerates the rate that the weight factors are adjusted.

The equation for adjusting the weight factors is:

New weight factor = [old weight factor] + [learning rate] x [gradient descendent correction term] + [momentum coefficient] x previous weight change]

When the learning rate is very slow, the training requires more number of iterations and the effect is very pronounced for complex networks and the training may be trapped at local minima (of obj. fn.) on the contrary if learning rate is very high the minima is never reached. Hence initially reasonably high and safe learning rate should be taken which goes on decreasing with iterations. General suggestion for learning rate and momentum coefficient is 0.3 and 0.4 respectively.

Selecting proper transfer function: Sigmoid and hyperbolic tangent functions are recommended for prediction/modeling and radial basis for fault diagnosis and feature categorization applications.

Selecting training data (Representative data): A representative data set prepared for training should contain information of all peaks, troughs and middle points also. Total number of data points available is divided into 4:1 ratio of training to testing data sets. Testing data sets should also contain some data points that belongs peaks, toughs and middle points.

Back Propagation of Error Algorithm

The back propagation or errors learning algorithm is nothing but changing the weight factors based between layer of interest and its previous layers based or error back propagated from the layer preceded by the layer of interest.

The weight factors in the network between any two layers are calculated as follows for minimizing the sum-squared error using steepest descent method.

Let w_{ij} be the weight factor between i^{th} hidden neurons and j^{th} output neuron. Then the change in w_{ij} i.e., Δw_{ij} is given by$\Delta w_{ij} = \beta_0\ y_i\ e_j$. where β_0 islearning rate for output layer y_i is output of i^{th} neuron in the hidden layer previous to output layer. E_j is error of j^{th} neuron in output layer given by the difference between its actual and network outputs.

$$W_{ij(new)} = w_{ij(old)} + \Delta w_{ij}$$

If T_j is the bias for j^{th} neuron in output layer, $\Delta T_j = \beta_0\ e_j$
If T_h is the bias for j^{th} neuron in hidden layer, $\Delta Th = \beta_0\ e_h$

When momentum correction is applied,

$$\Delta w[i][j]_t = \Delta w[i][j]_t + \alpha\Delta w[i][j]_{(t-1)}$$

where t indicates iteration number and α is the momentum coefficient.

$$E_2 = t_2 - y_2\ ;\ \Delta w_{12} = \beta_0\, x_1\ e_2\ ;\ e_{h2} = y_2\,(1\text{-}y_2)\ [w_{21}\ e_1 + w_{22}\ e_{21}];$$

$$\Delta w_{21} = \beta_0\, x_2\ e_2\ ;\ \Delta w_{12} = \beta_h\, w_{12}\ e_2$$

The back propagation learning algorithm:

```
Step 0: Initialize weights.
Step 1: While stopping condition is false.
        For each pair of set:
        a. Calculate outputs of all neurons in the network
        b. Calculate new weight factors and biases
Step 2: While stopping condition is false (go to step 1) else break
```

2. RESULTS AND DISCUSSIONS

The Program code is run for a target goal of 0.002 and simulated for different number of Neurons as follows:

Target Error = 0.002 No of Neurons =1

A Three dimensional plot of T_r, P_r and Z values for the first run is drawn as shown below.

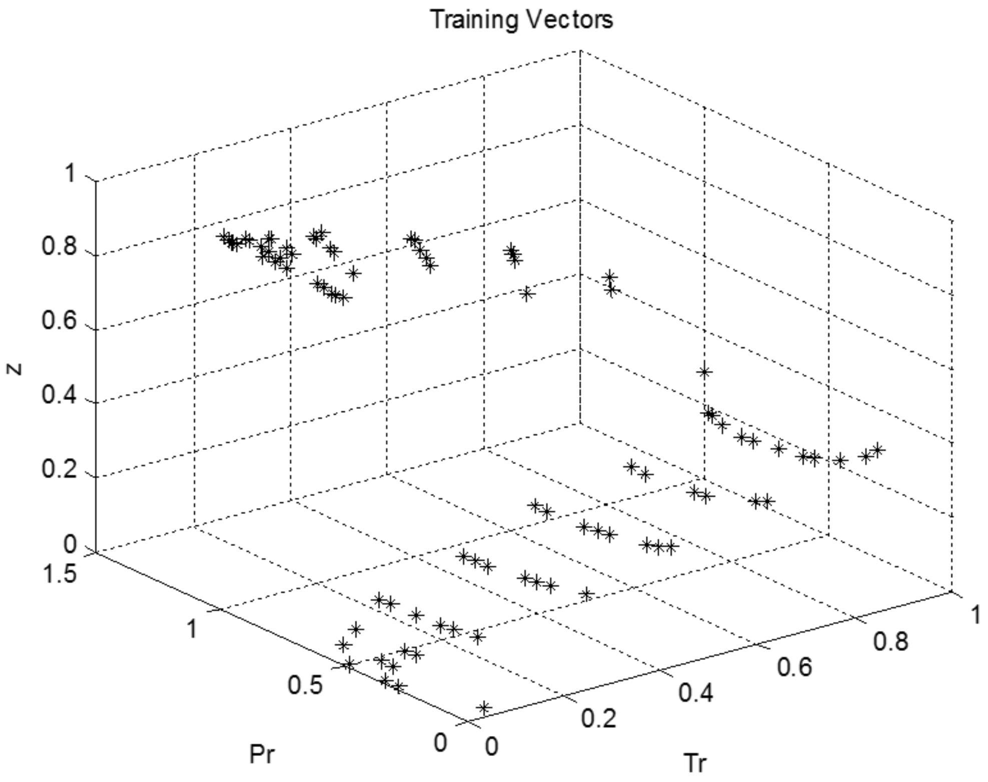


Fig. 1: 3-D Plot between (T_r, P_r, Z)

Sum-squared error goal (**SSE**) = 2.92067.

TRAINLM: Network error did not reach the error goal. Further training may be necessary, or try different initial weights and biases and/or more hidden neurons.

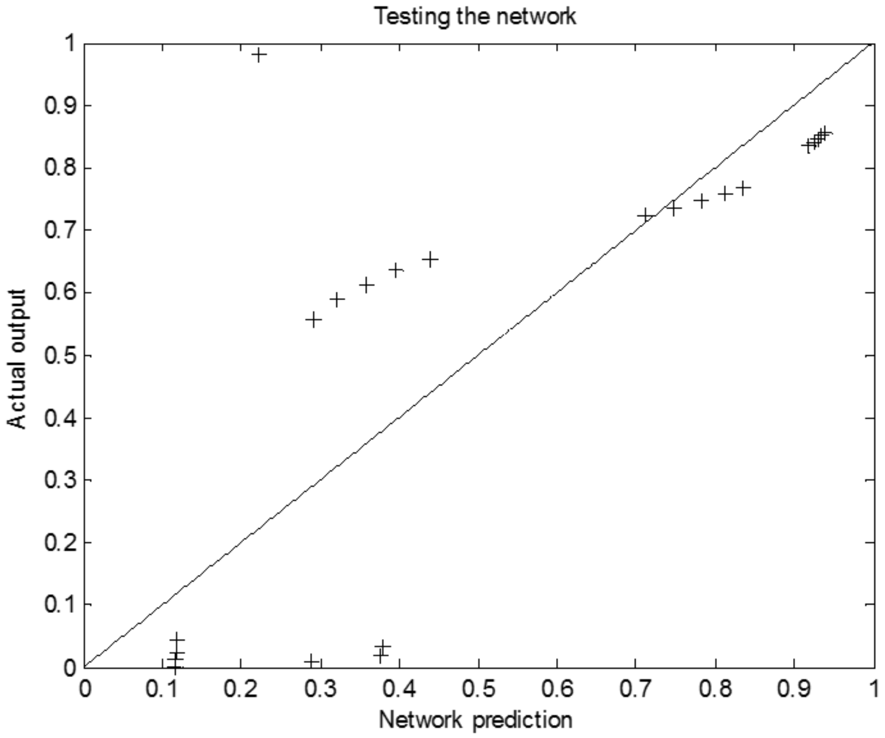


Fig. 2 ANN Prediction VsActual data points in testing for one number of neurons with an error of 0.002

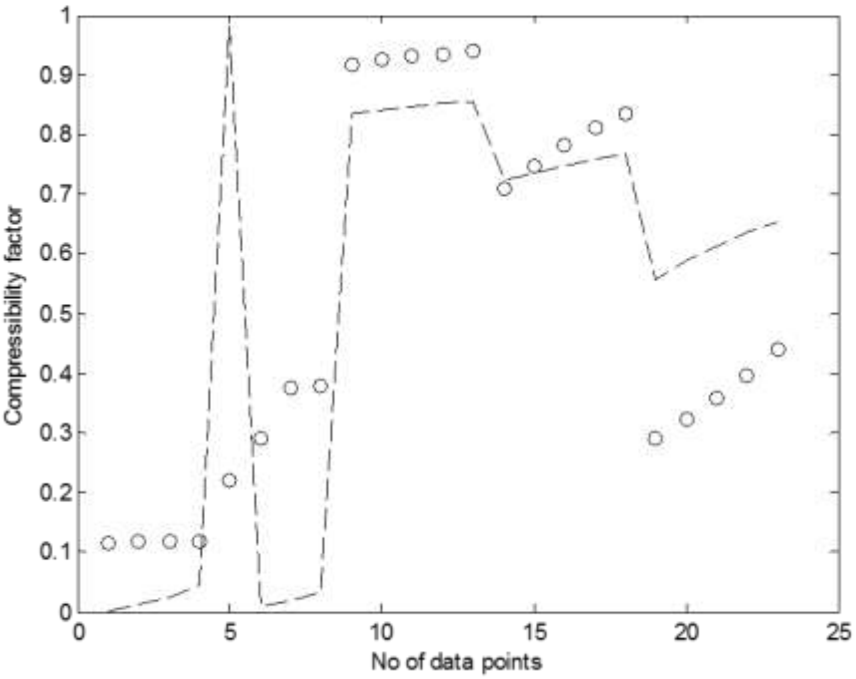


Fig. 3 Experimental Vs Model Predicted for testing Data pointsfor one number of neurons with an error of 0.002

Table1 Experimental Vs Model Values for one number of Neurons with an Error of 0.002

Expt. values	Model values	Expt. values	Model values	Expt. values	Model values	Expt. values	Model values
0.0029	0.1159	0.8206	0.8991	0.1158	0.1139	0.9946	0.9476
0.0130	0.1138	0.6635	0.5256	0.1564	0.1138	0.9768	0.9525
0.0239	0.1187	0.1321	0.1703	0.1904	0.1138	0.9573	0.9531
0.0442	0.1188	0.1614	0.1192	0.2200	0.1138	0.9174	0.9489
0.9804	0.2214	0.8561	0.9389	0.0778	0.1189	0.9965	0.9585
0.0093	0.2903	0.7574	0.8111	0.1109	0.1162	0.9826	0.9586
0.0178	0.3757	0.6138	0.3573	0.1415	0.1149	0.9659	0.9577
0.0344	0.3786	0.1844	0.1557	0.1703	0.1143	0.9322	0.9558
0.9935	0.9313	0.0021	0.1612	0.0611	0.3844	0.0882	0.1149
0.9725	0.9429	0.0110	0.1294	0.0983	0.2393	0.1429	0.1139
0.9528	0.9487	0.0239	0.1187	0.1321	0.1703	0.2084	0.1138
0.9115	0.9445	0.0522	0.1149	0.1664	0.1327	0.2892	0.1138
0.9963	0.9582	0.9922	0.8897	0.8338	0.9176	0.0687	0.1677
0.9821	0.9579	0.9505	0.7136	0.7360	0.7481	0.1063	0.1190
0.9648	0.9573	0.0178	0.3757	0.6138	0.3573	0.1476	0.1143
0.9300	0.9550	0.0371	0.1664	0.2901	0.1717	0.1939	0.1138
0.0825	0.1161	0.9961	0.9575	0.0026	0.1183	0.8059	0.8720
0.1322	0.1140	0.9790	0.9553	0.0119	0.1210	0.1006	0.3903
0.1904	0.1138	0.9528	0.9487	0.0221	0.1244	0.1301	0.1391
0.2604	0.1138	0.8810	0.8975	0.0413	0.1246	0.1626	0.1163
0.0670	0.2363	0.9968	0.9591	0.9849	0.3543	0.8509	0.9349
0.1027	0.1251	0.9832	0.9585	0.9377	0.4825	0.7471	0.7819
0.1415	0.1149	0.9648	0.9573	0.8958	0.6043	0.5887	0.3224
0.1842	0.1139	0.9523	0.9532	0.0336	0.6078	0.1779	0.1494

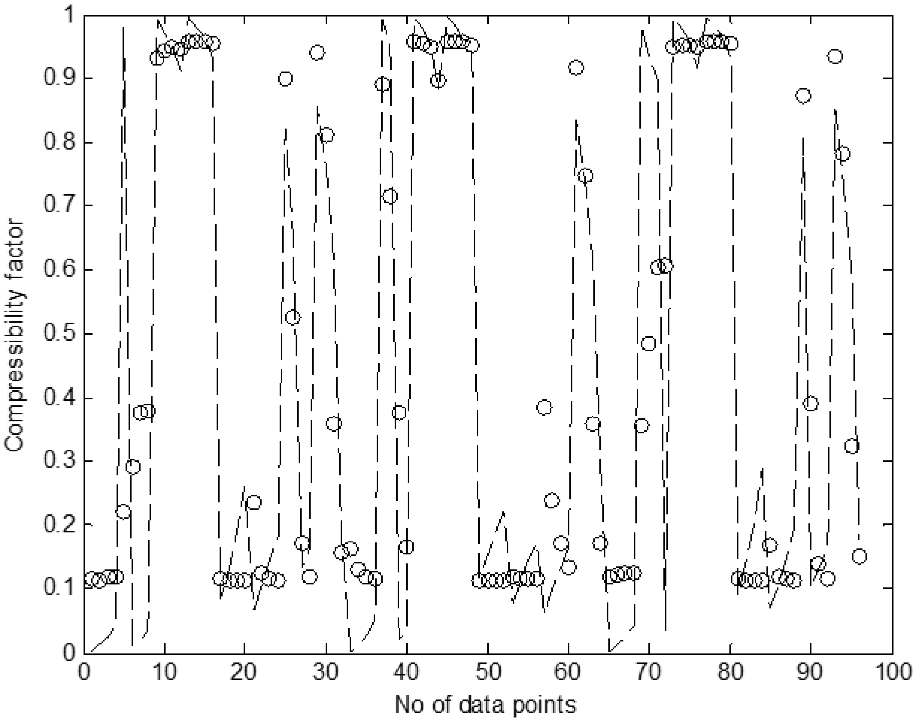


Fig. 4: Experimental Data points Vs Model Predictions (All data sets) for one number of neurons with an error of 0.002

3. CONCLUSION

A chemical engineer has to utilize many number of correlations in his day to day practice in order to calculate the required values. In the present work, as the functional relationship is extremely complex an ANN can be suggested as best method to solve with regard to its accuracy. We suggest an ANN with 37 hidden neurons, 2 input neurons and one output neuron to find compressibility factor from given T_r and P_r using Levenberg Marquardt second order optimization technique. The network proposed with above parameters is found to be very accurate and better than R-K or Vanderwaal two parameter model.

Generalized Equation of State

$f(P,V,T)=0$ is known as Equation of State for a gas where P,V,and T are the pressure, molar volume, and temperature of the gas. Whereas the ratio of pressure to critical pressure is known as reduced pressure. Similarly reduced volume and temperature are defined.

$f(\mathrm{Pr},Z,Tr,\varpi)=0$ is known as three parameter model of generalized equation of state. There are several generalized equations of state. Like

1. Cubic generalized equations of state : Vander Waal's, R-K , and SRK EOS.
2. Pitzer correlation based on Virial EOS.

So on.

The Pitzer correlation is limited to 20 psi pressure and cubic equations of state which are mathematically simpler and valid for both vapor and liquid phases are useful up to 70 psi (approximately) only. Whereas the generalized correlation for a wide range of conditions is too complex to be given a mathematical form. As the functional relationship is a black box several other models like

a) ANN
b) Fuzzy
c) Expert systems
may be considered.

Table 2: SSE values for different number of neurons with target goal of 0.002

Number of Neurons	Sum-squared error (SSE)	Is it a desired SSE
1	2.92067	No
10	0.0182752	No
20	0.0178607	No
30	0.0218734	No
40	**0.0019945**	**Yes**

As shown in table 2 the predicted SSE is matches with desired target value for forty (40) number of neurons which is the optimum number of neurons for the designed ANN for equation of state model.

Table 3: SSE values for different number of neurons with target goal of 0.001

Number of Neurons	Sum-squared error (SSE)	Is it a desired SSE
1	2.92067	No
15	0.0258809	No
30	0.00528496	No
45	**0.000951105**	**Yes**

As shown in table 3 the predicted SSE is matches with desired target value for forty five (45) number of neurons which is the optimum number of neurons for the designed ANN for equation of state model.

REFERENCES

[1] A neural network weight determination model designed uniquely for small data set learning.

[2] *Expert Systems with Applications,Volume 36, Issue 6, August 2009, Pages 9853-9858, Der-Chiang Li, Chiao-Wen Liu.*

[3] Neural Networks in Bioprocessing and Chemical Engineering by D.R. Baughman and Y.A.Liu, Academic Press, (1995).

[4] An ordinal optimization theory-based algorithm for a class of simulation optimization problems and application. *Expert Systems with Applications, Volume 36, Issue 5, July 2009, Pages 9340-9349. Shih-Cheng Horng, Shieh-Shing Lin.*

[5] An artificial neural network approach to compressor performance prediction. *Applied Energy, Volume 86, Issues 7-8, July-August 2009, Pages 1210-1221, K. Ghorbanian, M. Gholamrezaei*

[6] Artificial neural networks to predict daylight illuminance in office buildings. *Building and Environment, Volume 44, Issue 8, August 2009, Pages 1751-1757, Tuğçe Kazanasmaz, Murat Günaydin, Selcen Binol.*

[7] Adaptive neuro-fuzzy based inferential sensor model for estimating the average air temperature in space heating systems, *Building and Environment, Volume 44, Issue 8, August 2009, Pages 1609-1616, S. Jassar, Z. Liao, L. Zhao*

[8] Flow forecast by SWAT model and ANN in Pracana basin, Portugal. *Advances in Engineering Software, Volume 40, Issue 7, July 2009, Pages 467-473, Mehmet C. Demirel, Anabela Venancio, Ercan Kahya*

[9] Heat transfer of a helical double-pipe vertical evaporator: Theoretical analysis and experimental validation. *Applied Energy, Volume 86, Issues 7-8, July-August 2009, Pages 1144-1153. D. Colorado-Garrido, E. Santoyo-Castelazo, J.A. Hernández, O. García-Valladares, J. Siqueiros, D. Juarez-Romero*

[10] Modeling the characteristics of turbo compressors for fuel cell systems using hybrid method based on moving least squares. *Applied Energy, Volume 86, Issues 7-8, July-August 2009, Pages 1283-1289. R. Tirnovan, S. Giurgea, A. Miraoui, M. Cirrincione*

[11] Artificial intelligence approaches to achieve strategic control over project cash flows. *Automation in Construction, Volume 18, Issue 4, July 2009, Pages 386-393. Min-Yuan Cheng, Hsing-Chih Tsai, Chih-Lung Liu.*

[12] A particle swarm optimization-aided fuzzy cloud classifier applied for plant numerical taxonomy based on attribute similarity. *Expert Systems with Applications, Volume 36, Issue 5, July 2009, Pages 9388-9397. Hongfei Lu, Erxu Pi, Qiufa Peng, Lanlan Wang, Changjiang Zhang.*

[13] Neural Computing: A Technology Handbook for Professional II/PLUS and Neural Works Explorer.

[14] Himmelblau, D.M., "Use of Artificial Neural Networks to Monitor Faults and for Troubleshooting in the Process Industries"

[15] Hoskins,J.C. "Artificial Neural Networks for knowledge Representation in Chemical Engineering.

[16] Zhang, J. and A.J.Morris, "On-line process fault Diagnosis Using Fuzzy Neural Networks"

[17] Introduction to Artificial Neural Systems, by Jacek M. Zurada. *West Publishing company.*