Application of Artificial Neural Network for Prediction of Halogenated Refrigerants Vapor Pressure

Abazar Vahdat Azad[1] , Ali Kargari[2]*

[1]Energy Systems Engineering Department, Faculty of Mechanical Engineering, K.N. Toosi University of Technology, Tehran, Iran.

[2]Department of Petrochemical Engineering, Amirkabir University of Technology (Tehran polytechnic), Mahshahr Campus, Mahshahr, Iran.

Aplicación de una red neuronal artificial para la predicción de la presión de vapor de refrigerantes halogenados

Aplicació d'una xarxa neuronal artificial per a la predicció de la pressió de vapor de refrigerants halogenats

Recibido: 3 de febrero de 2009; revisado: 27 de enero de 2010; aceptado: 7 de febrero de 2010

RESUMEN

Se presenta la aplicación de una red neuronal artificial (RNA) para modelar la presión de vapor de algunos refrigerantes halogenados (metanos y etanos halogenados). La estructura de la red neuronal es de tipo no recurrente con algoritmo de retropropagación. Se determina el número óptimo de capas ocultas y de neuronas entre capas mediante un procedimiento de ensayo y error. Los parámetros de la red neuronal se obtienen mediante una fase de aprendizaje empleando el algoritmo de Levenberg-Marquardt. Los valores de presión de vapor a diferentes temperaturas obtenidos a partir de bibliografía de acceso abierto se consideran el objetivo del modelo neuronal. Las predicciones RNA de la presión de vapor son más precisas para un intervalo de temperaturas más amplio. La modelización RNA reduce el error medio para los refrigerantes de 0,69% a 0,31% para el intervalo de temperatura baja, y de 1,39% a 0,99% para el intervalo de temperatura alta. Finalmente, la modelización RNA reduce el error medio en comparación con la ecuación de Antoine en un 47,88% y un 32,18% para los intervalos de temperatura baja y alta, respectivamente.

Palabras clave: Red neuronal, refrigerante, predicción, presión de vapor.

SUMMARY

Application of Artificial Neural Network (ANN) for modeling of vapor pressure for some halogenated refrigerants (halogenated methanes and ethanes) is presented. Neural network training structure was feedforward with back-propagation algorithm.. The optimized number of hidden layer and neurons between layers were determined by a trial and error procedure. Neural network parameters were obtained through a learning phase by Levenberg-Marquardt algorithm. The vapor pressure at different temperatures obtained from open literatures was considered as the neural model target. ANN predictions of vapor pressure are more accurate for a wider range of temperature. The ANN modeling reduced the average error for the refrigerants from 0.69% to 0.31% for low temperature range and from 1.39% to 0.99% for high temperature range. Finally, ANN modeling reduced the average error in comparison to the Antoine equation by 47.88% and 32.18% for low and high temperature range, respectively.

Keyword: Neural network, Refrigerant, Prediction, Vapor pressure.

RESUM

Es presenta l'aplicació d'una xarxa neuronal artificial (XNA) per modelar la pressió de vapor d'alguns refrigerants halogenats (metans i etans halogenats). L'estructura de la xarxa neuronal és de tipus no recurrent amb algorisme de retropropagació. Es determina el número òptim de capes ocultes i de neurones entre capes mitjançant un procediment d'assaig i error. Els paràmetres de la xarxa neuronal s'obtenen mitjançant una fase d'aprenentatge emprant l'algorisme de Levenberg-Marquardt. Els valors de pressió de vapor a diferents temperatures obtinguts a partir de bibliografia d'accés obert es consideren l'objectiu del model neuronal. Les prediccions XNA de la pressió de vapor són més precises per a un interval de temperatures més ampli. La modelització XNA redueix l'error mitjà per als refrigerants de 0,69% a 0,31% per al interval de temperatura baixa, i d'1,39% a 0,99% per al interval de temperatura alta. Finalment, la modelització XNA redueix l'error mitjà en comparació amb l'equació d'Antoine en un 47,88% i un 32,18% per als intervals de temperatura baixa i alta, respectivament.

Mots clau: Xarxa neuronal, refrigerant, predicció, pressió de vapor.

* Corresponding author:
Tel: +98-65223-43645; Fax: +98-65223-43646;
email: kargari@aut.ac.ir; ali_kargari@yahoo.com

1. INTRODUCTION

Refrigerant selection for industrial applications is a complex and not straightforward process. Thermodynamic data on refrigerant fluids are of the utmost interest for the refrigeration Industry and, in particular, to design and optimize refrigeration equipment such as heat exchangers and compressors. There is an increasing need of new thermodynamic data such as vapor pressure in the field of refrigeration, also in the insulation industry (Mc Linden and Didion, 1989; Watanabe and Sato, 1992). Experimental data are essential during the screening phase, where candidate efficiencies should be compared. The saturated vapor pressure curve for each substance is unique, but all exhibit a general characteristic curve. Most of the common methods for linearzing the vapor pressure curve stem from the Clausius–Clapeyron equation, which relates the slope of the vapor pressure curve to heat of vaporization (Treybal, 1980). Another expression used for predicting vapor pressure data is the Antoine equation (Kabeel, 2005) and is given by

$$\log_{10}(P^{sat}) = A - \frac{B}{T - 273.15 + C} \quad (1)$$

Where A, B, and C are constants. P is saturated vapour pressure in (bar) and 'T' is temperature in Kelvin. The Antoine equation does not reproduce the correct shape of the vapor pressure curve beyond the temperature range indicated.

Artificial neural networks (ANN) are inspired by the biological neural system and its ability to learn through example. A great advantage of ANN models is that it is not necessary to know the mathematical relationship between the input and output variables. Instead they figure out these relationships through successive trainings. They learn from examples by constructing an input–output mapping without explicit derivation of the model equation. The current interest in artificial neural networks is largely due to their ability to mimic natural intelligence in its learning from experience (Wasserman, 1993). ANNs have been used in a broad range of applications including: pattern classification (Lippmann, 1987; Bishop, 1996), function approximation, optimization, prediction and automatic control (Pham and Liu, 1995). Additionally, ANNs have been used for atmospheric vapor pressure prediction (Potukuchi and Wexler, 1997). From ANN initiation there is hundreds of ANN architecture developed, however, some are more popular and have found numerous applications. Details have been dealt in elsewhere (Bishop, 1996 and 1994). Several authors have reported application of ANN for estimation of thermodynamic properties such as viscosity, density, compressibility factor and VLE. An ANN model for estimation of vapor pressure from aerosol composition, relative humidity and temperature has been reported (Potukuchi and Wexler, 1997). An ANN model has been used for estimating the compressibility factor for the liquid and vapor phase as a function of temperature and pressure for several refrigerants (Chouai et al., 2002). ANN has also been used for estimating the shape factors as a function of temperature and density for a number of refrigerants that can be used in the extended corresponding state model (Scalabrin, Piazza and Richon, 2002; Scalabrin, Piazza and Cristofoli, 2002). Laugier and Richon, (2003) have used ANN model for estimation of compressibility factor and density as a function of pressure and temperature for some refrigerants. They have proposed the use of an artificial feedforward neural networks based model in order to estimate the vapor pressure. ANN have been developed using the data available in the open literature. Detailed description of the multilayer feedforward neural networks and the backpropagation algorithm may be found in Haykin (1994). The most commonly used ANN architecture is the multilayer backpropagation neural network. Backpropagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions (Golden, 1996). In this work we first introduce the neural network which is appropriate for vapor pressure modeling, and then try to create the structure of layers and neurons that lead to Antoine equation. Therefore by using minimum number of hidden layers and neurons in layers, the resulted correlation is less complex.

2. METHODOLOGY

Artificial neural networks (ANN) consist of large numbers of computational units connected in a massively parallel structure. The processing units (neurons) from each layer "k" are linked to all of the other processing units appearing in layer "k+1" by weighted connections. Collectively, these connections (as well as the transfer functions of the processing units) form more or less good distributed representations of relationships between input and output data. ANNs have so far mainly been used in process modeling, process control (Chouai et al., 2000), fault diagnosis, error detection, data reconciliation and process analysis (Bulsari, 1995). Numerous papers have shown that a feedforward network is potentially able to approximate any non-linear function (Funahashi, 1989). More details about neural networks are given in Laugier et al. (1996).

A neuron with a single R-element input vector is shown in Figure 1. Here the individual element inputs $p_1, p_2, \cdots p_R$ are multiplied by weights $W_{1,1}, W_{1,2}, \ldots W_{1,R}$ and the weighted values are fed to the summing junction. Their sum is simply Wp, the dot product of the (single row) matrix W and the vector p.

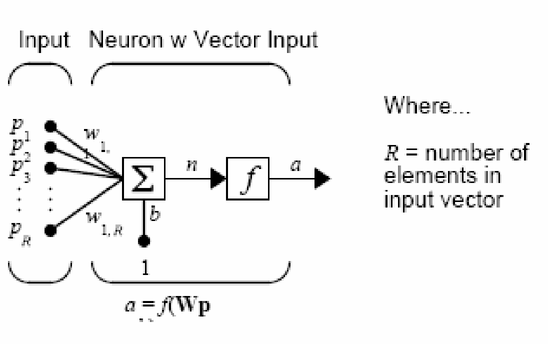

Figure 1- *A neuron with a single R-element input vector.*

The neuron has a bias b, which is summed with the weighted inputs to form the net input n. This sum, n, is the argument of the transfer function f.

$$n = W_{1,1}p_1 + W_{1,2}p_2 + \ldots + W_{1,R}\,p_R + b \quad (2)$$

This expression can, of course, be written in mathematical terms as:

$$n = \mathbf{W} \times \mathbf{p} + \mathbf{b} = \begin{bmatrix} W_{1,1} & W_{1,2} & . & . & . & W_{1,R} \end{bmatrix} \times \begin{bmatrix} p_1 \\ p_2 \\ . \\ . \\ . \\ p_R \end{bmatrix} + \mathbf{b} \qquad (3)$$

A network may have several layers. Each layer has a weight matrix W, a bias vector b, and an output vector a. To distinguish between the weight matrix, output vectors, etc., for each of these layers in our figures, we append the number of the layer as a superscript to the variable of interest. This layer notation in a three-layer network has been shown in Figure 2.

The network shown above has R^1 inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. It is common for different layers to have different numbers of neurons. A constant input 1 is fed to the biases for each neuron.

Note that the outputs of each intermediate layer are the inputs to the following layer. Thus layer 2 can be analyzed as a one-layer network with S^1 inputs, S^2 neurons, and an S^2XS^1 weight matrix W^2. The input to layer 2 is a1; the output is a2. Now that all the vectors and matrixes of layer 2 have been identified, it can be treated as a single-layer network on its own. This approach is taken with any layer of the network.

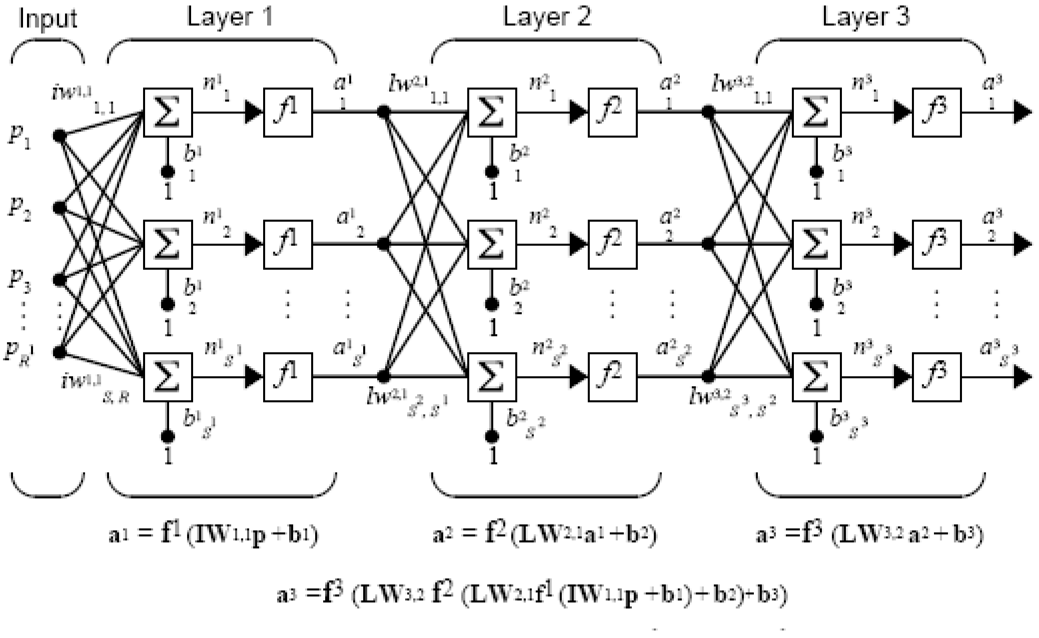

Figure 2- *A feed-forward artificial neural network architecture.*

The layers of a multilayer network play different roles. A layer that produces the network output is called an output layer. All other layers are called hidden layers. The three-layer network shown earlier has one output layer (layer 3) and two hidden layers (layer 1 and layer 2). Some authors refer to the inputs as a fourth layer.

Once the network weights and biases have been initialized, the network is ready for training. The training process requires a set of examples of proper network behavior - network inputs as 'p' and target outputs as't'. During training the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function for feedforward networks is mean square error MSE (the average squared error between the network outputs and the target outputs t). Here we describe the training procedure for feedforward networks. These algorithms use the gradient of the performance function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called backpropagation, which involves performing computations backwards through the network.

There are many variations of the backpropagation algorithm. The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative of the gradient. One of the iterations of this algorithm can be written as:

$$x_{k+1} = x_k - \alpha_k g_k \qquad (4)$$

Where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate.

There are two different ways in which this gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all of the inputs are applied to the network before the weights are updated. In batch mode the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases. The proposed ANN operates in the batch mode.

The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feedforward networks), then the Hessian matrix can be approximated as

$$H = J^T J \quad (5)$$

and the gradient can be computed as

$$g = J^T e \quad (6)$$

Where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The J Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix.

The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (7)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced by increase in iteration number. This algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights).

The neurons within the hidden layer perform two tasks: they sum the weighted inputs connected to them and then pass the resulting summations through a non-linear activation function to the output neuron or adjacent neurons of the corresponding hidden layer (in case of more than one hidden neuron layer). Multilayer networks often use the sigmoid transfer function (Figure 3). In this work, the sigmoid function is used in the interval (0, +1) that is:

$$f(x) = \frac{1}{(1 + e^{-x})} \quad (8)$$

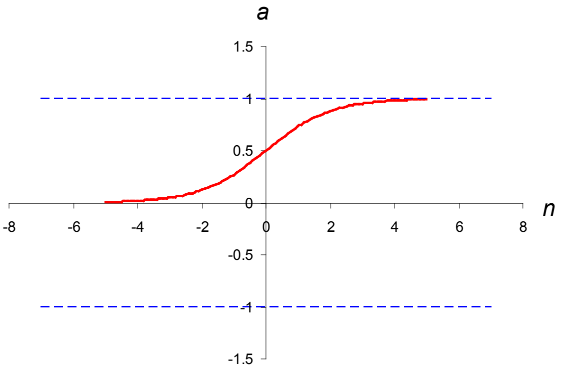

Figure 3- *The Sigmoid transfer function.*

Latest layer has Purelin function that is (Figure 4):

$$f(x) = \text{Purelin}(ax + b) = ax + b \quad (9)$$

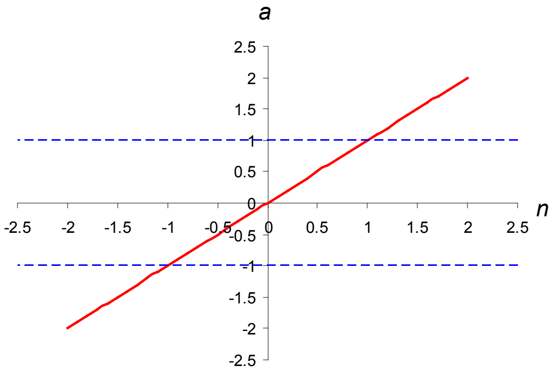

Figure 4- *The Pureline transfer function.*

An important aspect of a neural network is the learning step, based on a set of measured numerical values (the learning database). The aim is to find the optimal structure of neural network by considering the minimum number of layer and neurons in each layer (for simplest formulation from trained neural network structure) and also maximum accuracy of result which formulate the vapor pressure versus temperature in Antoine equation form.

3. NEURAL NETWORK MODEL

The optimized number of hidden neurons has been determined during the learning phase by trial and error procedure. Lower number of hidden layers and neurons in each layer will ease the extraction of the formula from the trained neural network structure. As a result the network must have one hidden layer with one neuron in the layer (Figure 5), also the output must be in term of $\log_{0}(P^{sat})$, where P^{sat} is the saturated vapor pressure. Moreover, input of neural network must be in term of $\frac{1}{T - 273.15 + C}$ where T is temperature in Kelvin and C is the third Antoine constants. Therefore the first and the second Antoine constant as 'A' and 'B' are changed. The experimental vapor pressure data for the halogenated refrigerants adopted from open literatures (Smith and Srivastava, 1986; Raznjevic, 1976; Perry and Green, 1999).

Take the weight (1, 1) and the bias (1) equal to "A" and "-B", respectively. Due to the structure of the network, the first layer input is $\frac{1}{T - 273.15 + C}$ and the output of the first layer is $A + \frac{(-B)}{T - 273\ 15 + C}$. Because neuron of this layer has a Pureline function, then the output of layer which is ANN output) is $A - \frac{B}{T - 273\ 15 + C}$).

Then the final output of the ANN is $\log_{10}(P^{sat})$

$$\log_{10}(P^{sat}) = A - \frac{B}{T - 273\ 15 + C} \quad (10)$$

The deviation between the calculated and experimental vapor pressures was calculated as follows:

$$\%\ Error = \left| \frac{P_{ANN} - P_{\exp}}{P_{\exp}} \right| \times 100 \quad (11)$$

In which, P_{ANN} is the calculated vapor pressure using equation (10) and P_{exp} is the reported experimental vapor pressure.

Table 1- *Experimental data (Smith and Srivastava, 1986; Raznjevic, 1976; Perry and Green, 1999) and reported Antoine constants (Polinig et al., 2001) for some halogenated refrigerants.*

Substance	No. of exp. data	Temperature Range (K)	Ref.	Antoine constants (Reported)			Temperature Range (K)
				A	B	C	
R10(CCl4)	45	259-556	21	4.10445	1265.632	232.148	259-373.76
R11(CFCl3)	46	233-323	18	4.00905	1043.313	236.95	218.98-317.57
R12(CF2Cl2)	134	203-388	18	4.01171	868.076	246.39	178.7-260.7
R12B1(CBrClF2)	45	199-324	21	3.9585	933.04	240	198-288.26
R13(CF3Cl)	35	143-263	18	3.90353	654.656	249.36	140.61-205.48
R13B1(CBrF3)	44	160-311	21	3.8964	731.31	245.7	158.1-230.85
R14(CF4)	45	107-180	21	3.95894	510.595	257.2	106.2-155-54
R20(CHCl3)	31	252-431	21	3.96288	1106.904	218.552	250.1-356.89
R21(CHCl2F)	38	245-450	21	4.02473	959.934	230.03	210.83-300.91
R22(CHF2Cl)	82	173-301	18	4.13253	835.462	243.46	173.13-247.74
R23(CHF3)	45	144-298	21	4.2214	707.396	249.84	142.79-203.75
R30(CH2Cl2)	41	212-510	21	4.07622	1070.07	223.24	235.2-333.36
R32(CH2F2)	45	191-351	21	4.29712	833.137	245.86	166.22-235.78
R40(CH3Cl)	45	213-333	18	4.16533	920.86	245.58	184.6-265.87
R123(C2HCl2F3)	17	249-456	14	4.21161	1132.447	241.59	223.16-321.15
R124(C2HClF4)	18	233-395	14	4.0536	900.49	234.389	222-286
R134A(C2H2F4)	24	232-374	21	4.11874	850.881	232.99	186.41-263.04
R142B(C2H3ClF2)	28	251-354	21	4.05053	928.645	238.69	195.98-282.13
R143A(C2H3F3)	21	161-346	14	4.068	801.34	244.55	167-241
R150(C2H4Cl2)	35	279-403	21	4.28356	1341.37	230.05	267.4-379.91
R150A(C2H4Cl2)	35	248-493	21	4.1678	1201.05	231.27	246.6-352.49
R160(C2H5Cl)	37	217-416	21	4.09088	1020.63	237.57	211.86-304.89
R160B1(C2H5Br)	39	218-503	21	4.04485	1090.811	231.71	231.35-332.8
R161(C2H5F)	45	178-248	21	4.21998	897.368	250.66	174.1-251.47

Table 1 shows the experimental data characteristics and the reported Antoine constants including the temperature range for each substance. The results of ANN calculations were shown in Table 2. In this table the AANN and BANN are calculated Antoine constants using ANN also R2 which is the square of the Pearson product moment correlation coefficient through data points for known data (experimental data).

Table 2- *Calculated Antoine constants using ANN structure.*

Substance	A_{ANN}	B_{ANN}	C	Temperature Range (K)		R^2
				Tmin	Tmax	
R10	4.099	1264.702	232.148	259	556	0.99957
R11	4.021	1046.633	236.95	233	323.15	0.99999
R12	4.01677	868.8362	246.39	203.15	388.65	0.99949
R12B1	3.9563	932.3444	240	199	324	0.99999
R13	3.9164	656.1323	249.36	143.15	263.15	0.99993
R13B1	3.896	729.8489	245.7	160	311	0.99980
R14	3.9605	510.7252	257.2	107	180	0.99999
R20	3.9353	1099.421	218.552	252	431	0.99980
R21	4.0413	963.6388	230.03	245	450	0.99944
R22	4.1428	838.2166	243.46	173.15	301.15	0.99998
R23	4.23	708.412	249.84	144	298	0.99955
R30	4.0873	1073.005	223.24	235	323	0.99972
R32	4.3057	834.9511	245.86	191	351	0.99935
R40	4.1511	919.5913	245.58	213.15	333.15	0.99998
R123	4.2415	1141.7571	241.59	249.49	456.83	0.99975
R124	4.05506	900.209	234.389	233.15	395.65	0.99941
R134A	4.1184	850.9421	232.99	169.85	374.18	0.99928
R142B	4.0304	921.7775	238.69	251	354	0.99991
R143A	4.0707	802.1276	244.55	161.82	346.75	0.99931
R150	4.2925	1344.3099	230.05	279	403	0.99999
R150A	4.155514	1197.7113	231.27	248	493	0.99958
R160	4.085028	1019.2620	237.57	217	460	0.99986
R160B1	4.0565	1093.7197	231.71	218	503	0.99281
R161	4.2222	897.7961	250.66	169	375	0.99965

The artificial neural network trained on overall range of actual data point. This range is higher than temperature range for reported Antoine equation (Table 1). We will refer to the Antoine equation temperature range as "low range temperature" and that for the experimental data as "high range temperature". A comparison between the results of vapor pressure predictions by Antoine equation and ANN model over the low range and high range temperatures were shown in Table 3. Figures 6 and 7 show the error percent for the vapor pressure prediction using Antoine equation and ANN model both for low and high range temperature ranges, respectively. As see ANN prediction is more accurate than Antoine equation. Table 4 shows the percent of error reduction by ANN modeling in comparison to the Antoine equation.

Table 3- Comparison between error of Antoine and ANN at low and high temperature interval.

Substance	LOW RANGE TEMPERATURE			HIGH RANGE TEMPERATURE		
	Temperature range	% Error		Temperature range	% Error	
		ANTOINE	ANN		ANTOINE	ANN
R10	259-373	0.62	0.62	259-556	1.58	1.28
R11	233-317	0.52	0.18	233-323	0.51	0.17
R12	203-260	0.40	0.11	203-388	0.43	0.34
R12B1	199-284	0.28	0.02	199-324	0.35	0.14
R13	143-203	0.57	0.38	143-263	0.98	0.40
R13B1	160-231	1.81	1.23	160-311	1.80	1.17
R14	107-155	0.10	0.01	107-180	0.20	0.09
R20	252-352	0.79	0.41	252-431	1.61	0.76
R21	245-301	0.21	0.38	245-450	2.28	1.49
R22	173-247	1.23	0.30	173-301	0.90	0.28
R23	144-202	0.38	0.16	144-298	0.99	0.83
R30	235-332	0.66	0.38	212-510	2.04	1.65
R32	191-234	0.21	0.09	191-351	1.72	1.41
R40	213-265	1.98	0.20	213-333	2.11	0.27
R123	249-321	1.12	0.42	249-456	0.83	0.49
R124	233-283	0.82	0.38	233-395	1.85	1.46
R134A	232-263	0.16	0.08	232-374	2.10	2.15
R142B	251-281	2.55	0.89	251-354	1.34	0.87
R143A	161-240	0.45	0.09	161-346	1.47	1.32
R150	279-378	0.36	0.25	279-403	0.34	0.28
R150A	248-349	0.44	0.38	248-493	2.96	2.49
R160	217-305	0.36	0.22	217-416	0.64	0.62
R160B1	233-330	0.35	0.12	218-503	3.11	2.86
R161	169-375	0.16	0.14	178-248	0.99	0.89
Average		0.69	0.31		1.39	0.99

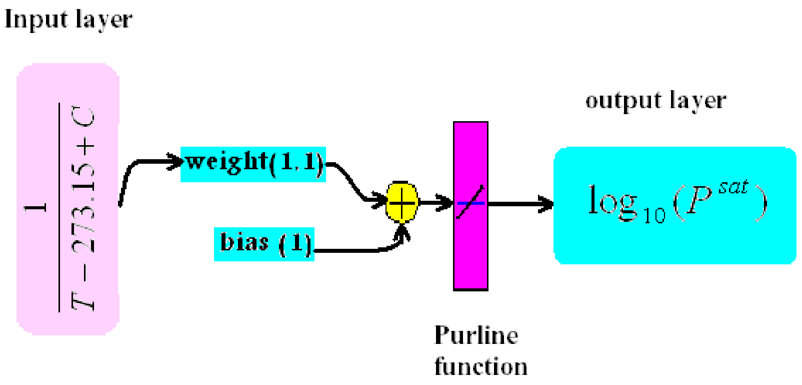

Figure 5 - *A one-layer feed forward ANN.*

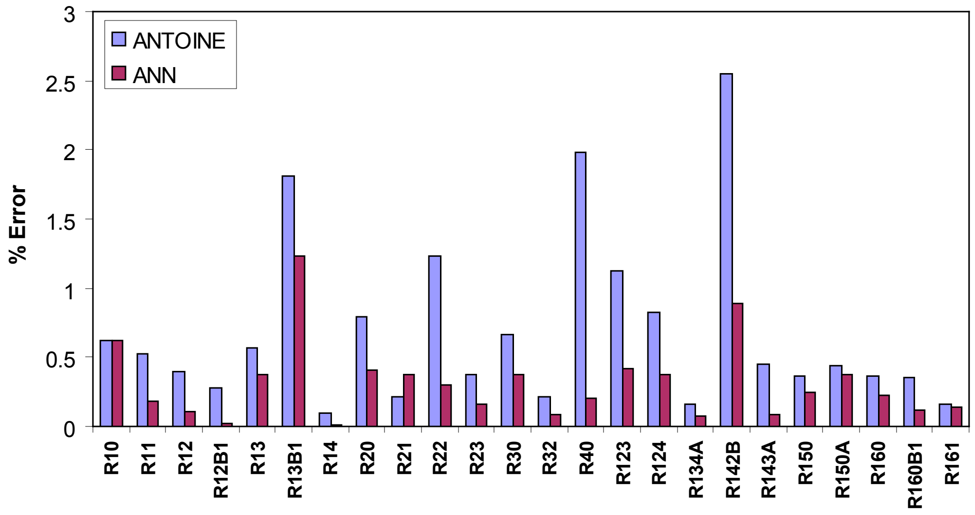

Figure 6- *Error percent for vapor pressure prediction by Antoine equation and ANN modeling for some halogenated refrigerants at low range temperature.*

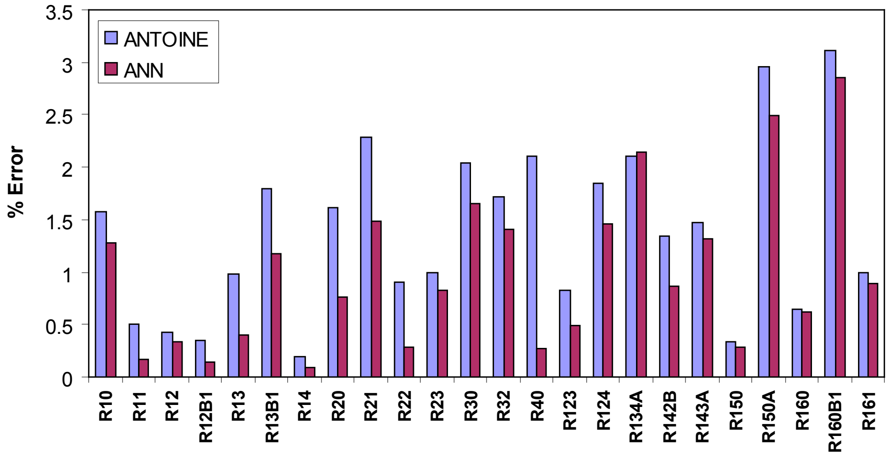

Figure 7- *Error percent for vapor pressure prediction by Antoine equation and ANN modeling for some halogenated refrigerants at low range temperature.*

4. MATLAB AND ANN

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Developed by The MathWorks, This collection includes the following topics as typical uses include Math and computation, Algorithm development, Data acquisition, Modeling, simulation and prototyping, Data analysis, exploration and visualization, Scientific and engineering graphics and Application development. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The Neural Network Toolbox is designed to allow for many kinds of networks. This makes it possible for many functions to use the same network object data type. In this paper the Neural Network Toolbox use feed-forward neural network training structure and Back-propagation algorithm. The optimized number of hidden layer and neurons in layer were determined during the learning phase by trial and error procedure. Neural network parameters were obtained through a learning phase by Levenberg-Marquardt algorithm (http://www.mathworks.com/products/).

Table 4- *The percent of error reduction by ANN modeling respect to the Antoine equation.*

Substance	LOW RANGE TEMPERATURE	HIGH RANGE TEMPERATURE
R10	0.29	18.99
R11	65.38	66.67
R12	72.50	20.93
R12B1	92.86	60.00
R13	33.33	59.18
R13B1	32.04	35.00
R14	90.00	55.00
R20	48.10	52.80
R21	-80.95	34.65
R22	75.61	68.89
R23	57.89	16.16
R30	42.42	19.12
R32	57.14	18.02
R40	89.90	87.20
R123	62.50	40.96
R124	53.66	21.08
R134A	50.00	-2.38
R142B	65.10	35.07
R143A	80.00	10.20
R150	30.56	17.65
R150A	13.64	15.88
R160	38.89	3.13
R160B1	65.71	8.04
R161	12.50	10.10
Average	47.88	32.18

5. CONCLUSION

ANN structure was used to optimize the vapor pressure prediction via Antoine equation for some halogenated refrigerants. Neural network training structure was feedforward and back propagation algorithm. The results show that the modeling predicts vapor pressure better than traditional Antoine equation. The modeling offers more accurate predictions for a wider range of temperature. The modeling reduced the average error for the refrigerants from 0.69% to 0.31% for low range temperature and from 1.39% to 0.99% for high range temperature. Finally, ANN modeling reduced the average error for the above refrigerants in comparison to the Antoine equation, by 47.88% and 32.18% for low and high range temperature, respectively.

NOMENCLATURE

A	Antoine constant
$\mathrm{A}_{\mathrm{ANN}}$	Calculated Antoine constants using ANN
a^i	Output of ith layer
a_k	The learning rate
B	Antoine constant
b	Bias
$\mathrm{B}_{\mathrm{ANN}}$	Calculated Antoine constants using ANN
$b^i{}_j$	Bias for jth node in ith layer of ANN structure
C	Antoine constant
e	Vector of network errors
f^i	Function of nodes in ith layer
g_{k}	The current gradient
H	Hessian matrix
J	Jacobian matrix
MSE	Squared error between the network outputs and the target outputs
k	Number of ANN layer
n	Net input value (the sum of the weighted inputs and the bios)
n^i_j	The argument of the transfer function in jth neuron of ith layer
P	Pressure , (bar)
p	Input vector to ANN
$\mathrm{P}_{\mathrm{ANN}}$	The calculated vapor pressure using ANN (bar)
$\mathrm{P}_{\mathrm{exp}}$	The reported experimental vapor pressure (bar) from literatures
p_i	Input parameter of ANN structure
P^{sat}	Saturation pressure ,(bar)
R	Number of elements in input vector
R^2	Square of the Pearson product moment correlation coefficient through data points
S^i	Number of neurons in the ith layer
T	Temperature , (Kelvin(
t	Target output
W	Single row matrix that includes the weight element
weight (i, j)	Element of weight matrix in ith row and ith column
W^i	Weight matrix of ith layer
$W_{i,j}$	Weight for ith input to jth node in ANN structure
Wp	Dot product of the (single row) matrix W and the vector p
x_{k}	Vector of current weights and biases
m	Performance function

6. BIBLIOGRAPHY

1. Bishop, C.M., 1994. Neural networks and their applications. Rev. Sci. Instrum. 65, 1803–1832.
2. Bishop, C.M., 1996. Neural Networks for Pattern Recognition. Oxford University Press.
3. Bulsari, A.B., 1995. Neural Networks for Chemical Engineers. Elsevier, Amsterdam.
4. Chouai, A., Cabassud, M., Le Lann, M.V., Gourdon, C., Casamatta, G., 2000. Multivariable control of a pulsed liquid-liquid extraction column by neural networks. Neural Comput. Application. 9, 181–189.
5. Chouai, A., Laugier, S., Richon, D., 2002. Modeling of thermodynamic properties using neural networks application to refrigerants. Fluid Phase Equilibria. 199. 53–62.
6. Funahashi, K.I., 1989. On the approximate realization of continuous mappings by neural networks. Neural Networks. 2, 183–192.
7. Golden, R.M., 1996. Mathematical methods for neural networks analysis and design. The MIT Press.
8. Haykin, S., 1994. Neural Networks: A Comprehensive Foundation. Macmillan College Publishing.
9. Kabeel, A.E., 2005. Augmentation of the performance of solar regenerator of open absorption cooling system. Renewable Energy. 30, 327– 338.
10. Laugier, S., Defaye, G., Richon, D., 1996. Proceedings of the 7th Congress of the Asian Pacific Confederation of Chemical Engineers, Vol. 3, Taipei, Taiwan. 1039–1046.
11. Laugier, S., Richon, D., 2003. Use of artificial neural networks for calculating derived thermodynamic quantities from volumetric property data. Fluid Phase Equilibria. 210, 247–255.
12. Lippmann, R.P., 1987. An introduction to computing with neural nets. IEEE ASSP Magazine. 4–22.
13. McLinden, M.O., Didion, D.A., 1989. Thermophysical-property needs for the environmentally acceptable halocarbon refrigerants. Int. J. Thermophys, 10, 563–576.
14. Perry, R.H., Green, D., 1999. Perry's Chemical Engineering Handbook. 7th Ed., McGraw Hill, New York, NY.
15. Pham, D.T., Liu, X., 1995. Neural Networks for Identification, Predication and Control. Springer-Verlag, London.
16. Polinig, B.E., Prausnitz, J.M., O'connell, J.P., 2001. The properties of gases and liquids. 5th Edition, McGraw Hill, New York, NY.
17. Potukuchi, S. Wexler, A.S., 1997. Predicting vapor pressures using neural networks. Atmosph. Envir. 31, 741–753.Watanabe, K., Sato, H., 1992. Progress of the thermophysical properties research on environmentally acceptable refrigerants in Japan. Fluid Phase Equilibria. 80, 1–17.
18. Raznjevic, K., Handbook of thermodynamic; tables and charts. Hemisphere Publishing Company, New York, NY.
19. Scalabrin, G., Piazza, L., Cristofoli, G., 2002. Application of neural networks to a predictive extended corresponding states model for pure halocarbon thermodynamics. Int. J. Thermophysics. 23, 57–75.
20. Scalabrin, G., Piazza, L., Richon, D., 2002. An equation of state for R227ea from density data through a new extended corresponding states-neural network technique. Fluid Phase Equilibria. 199, 33–51.
21. Smith, B.D., Srivastava, R., 1986. Thermodynamic data for pure compounds, part B: halogenated hydrocarbons and alcohols. Elsevier, New York, NY.
22. Treybal, R.E., 1980. Mass Transfer Operations, 3rd ed., McGraw-Hill, New York.
23. Wasserman, P., 1993. Advanced Methods in Neural Computing. Van Nostrand Reinhold, New York, USA.
24. http://www.mathworks.com/products/