Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management
Detroit, Michigan, USA, September 23-25, 2016

Neural Network Viscosity Models for Multi-Component Liquid Mixtures

Adel Elneihoum, Hesham Alhumade, Ibrahim Alhajri, Walid El Garwi, Ali Elkamel
Department of Chemical Engineering, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
halhumade@uwaterloo.ca, Aelkamel@uwaterloo.ca

Abstract

An artificial neural network has been developed for the prediction of the kinematic viscosity of ternary, quaternary, and quinary systems. The systems investigated consisted of the following components: Heptane, Octane, Toluene, Cyclohexane, and Ethylbenzene at atmospheric pressure and temperatures of 293.15, 298.15, 308.15, and 313.15 K. The developed model was based on a three-layer neural network with six neurons in the hidden layer and a back propagation learning algorithm. The neural network was trained with binary systems consisting of 440 data sets and using mole fractions combined with temperature as input. A comparison of the experimental values and the results predicted from the neural network revealed a satisfactory correlation, with the overall AAD for the ternary, quaternary, and quinary systems of 0.8646 %, 1.1298 %, and 4.3611 %, respectively. The results were further compared to the generalized McAllister model as an alternative empirical model. The neural network produced better results than the generalized McAllister model. The new approach established in this work helps reduce the amount of experimental work required in order to determine most of the parameters needed for other models and illustrates the potential of using a neural network method to estimate the kinematic viscosity of many other mixtures.

Keywords
Optimization, Viscosity, Fluids, Neural Network.

1. Introduction

An Artificial Neural Network (AAN) is a computation process that attempts to mimic some of the basic information processing methods of the human brain. The study of neural networks attracted many researchers from a wide variety of disciplines such as biology, engineering and mathematics. AAN consists of many simple processing elements called neurons. The neurons are interconnected in layers and simultaneously execute computations in parallel. Models of artificial neural networks have been under development for many years in a variety of scientific fields, the objective is to obtain meanings from complicated data sets and to build forecasting. In this chapter, artificial neural networks are introduced as a process modeling tool. In particular their applications in chemical engineering modeling and more specifically in physical properties modeling are briefly reviewed. Artificial neural networks represent an alternative approach to physical modeling; furthermore they are frequently utilized for statistical analysis and data modeling, in which their role is perceived as an alternative to standard nonlinear regression or cluster analysis techniques [1]. Artificial neural networks can deal successfully with non-linearity, handle noisy or irregular data, correlate hundreds of variables or parameters, and provide generalized solutions [2].

The estimation of the viscosities at different temperatures of compounds that have yet to be manufactured requires a generalized predictive model with a high level of accuracy. The kinetic theory of gases provides a comprehensive explanation of the viscosity of gases. However, for liquids, a theoretical explanation is complex because of the wide range of intermolecular forces involved, such as hydrogen bonding, attraction, repulsion, electrostatic effects, and the high degree of disorder between molecules.

Solving chemical engineering problems with traditional techniques has limitations, for example, those encountered in the modeling of extremely complex and nonlinear systems. Artificial neural networks (ANNs)

© IEOM Society International

are general tools that have been proven to be capable of estimating, classifying, predicting, modeling and optimizing complex systems and are therefore extremely practical in technical applications. ANNs are a division of artificial intelligence (AI) that has the goal of replicating the abilities of the human brain. One of the main characteristics of neural networks is that they learn from observation and improve their performance accordingly; furthermore, they are suitable for complicated phenomena which involve experimental data whose relationships are not well understood.

The objective of this research can be stated as follows: Investigate the validity of artificial neural networks for predicting the kinematic viscosity of a multi-component mixture at a variety of temperatures. Develop, train, and implement a set of artificial neural networks in order to predict the kinematic viscosity of a multi-component mixture. Finally, compare the results of these neural network models with the results obtained from experimental tests.

2. Experimental

Obtaining experimental data for viscosity composition at different temperatures is a very laborious work that requires intensive effort and time. Reducing the laboratory work using reliable predictive models is of great deal of importance since it saves such efforts. The aim of this work is to use ANNs to predict viscosity of multi-component liquid mixtures using only binary data that is widely available in the literature and compare its predictions with the generalized McAllister Model. The modeling methodology used in the present work is illustrated by the flowchart depicted in Figure 1. Details of the methodology are explained in order.

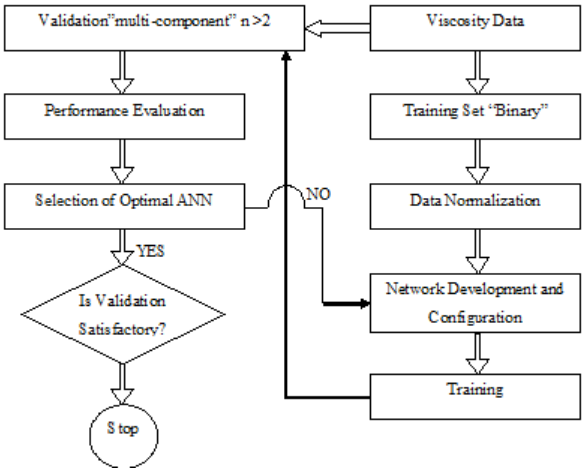


Figure 1. Methodology for developing an ANN Architecture.

2.1 Selection of Data

The data sets which consist of Heptane, Octane, Toluene, Cyclohexane, and Ethylbenzene included the viscosity values for binary, ternary, quaternary, and quinary liquid mixture are categorized as systems and subsystems as listed in Table 1. Artificial Neural Networks, like other empirical models, could be completed with databases of any size; however generalization of these models to data from outside the model development domain will be negatively affected. Since Artificial Neural Networks are essential to generalize for unseen cases, they must be utilized as an interpolator. Training data should be satisfactorily large to cover the possible known variation in the problem domain. Models developed from data generally depend on database size; however more data also helps when noise is present in the datasets.

Table 1. The data of the system and subsystems of kinematic viscosities.

System	Temperature (K)	McAllister Model %AAD	Data Source
Heptane-Octane	293.15-313.15	0.23	[3]
Heptane-Cyclohexane	293.15-313.15	1.83	[3]
Heptane-Toluene	293.15-313.15	1.46	[3]

© IEOM Society International

Heptane-Ethylbenzene	293.15-313.15	1.88	[3]
Octane-Cyclohexane	293.15-313.15	1.71	[3]
Octane-Toluene	293.15-313.15	2.31	[3]
Octane-Ethylbenzene	293.15-313.15	2.33	[3]
Cyclohexane-Toluene	293.15-313.15	3.09	[3]
Cyclohexane-Ethylbenzene	293.15-313.15	2.96	[3]
Toluene-Ethylbenzene	293.15-313.15	0.33	[3]
Heptane-Cyclohexane-Ethylbenzene	293.15-313.15	4.89	[4],[5]
Heptane-Octane-Cyclohexane	293.15-313.15	2.49	[4],[5]
Heptane-Octane-Ethylbenzene	293.15-313.15	3.93	[4],[5]
Heptane-Octane-Toluene	293.15-313.15	3.37	[4],[5]
Heptane-Toluene-Cyclohexane	293.15-313.15	5.12	[4],[5]
Heptane-Toluene-Ethylbenzene	293.15-313.15	2.59	[4],[5]
Octane-Cyclohexane-Ethylbenzene	293.15-313.15	2.88	[4],[5]
Octane-Toluene-Cyclohexane	293.15-313.15	2.98	[4],[5]
Octane-Toluene-Ethylbenzene	293.15-313.15	3.4	[4],[5]
Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	3.66	[4],[5]
Heptane-Octane-Cyclohexane-Ethylbenzene	293.15-313.15	2.83	[4],[5]
Heptane-Octane-Toluene-Cyclohexane	293.15-313.15	3.35	[4],[5]
Heptane-Octane-Toluene-Ethylbenzene	293.15-313.15	2.15	[4],[5]
Heptane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	3.83	[4],[5]
Octane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	2.72	[4],[5]

The data set of the system consisted of Heptane, Octane, Toluene, Cyclohexane, and Ethylbenzene was separated into a training set and testing sets to validate the network performance. The training set contained the binary data for the system. All ten binary combinations were used in the training. The testing datasets contained the ten ternary sub-systems: the five quaternary sub-system and one quinary system as listed in Table 1. The percent absolute average derivation %AAD is used for comparison purposes with the ANN predictions. The %AAD is defined as:

$$AAD = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\nu_i^{\exp} - \nu_i^{pred}}{\nu_i^{\exp}}\right| \qquad (1)$$

Where n is the number of data points.

© IEOM Society International

2.2 Weight Idealization

The weights and bias are generated with random values. The most common weight and bias initialization function used in the Matlab is "rands". This is a symmetric random weight/bias initialization where the weights are generated with small random values between -1 and 1. The weight initialization improvement is very crucial for a large number of hidden neurons used with complicated desired outputs and when training time is required to be reduced significantly from days to hours. The above mentioned parameters and conditions do not apply to our network therefore random weight initialization was chosen.

2.3 Normalization of Data

The input and output data of neural networks should be normalized to have the same order of magnitude. Normalization is a very significant step in the process. If the input and the output variables are not of the same order of magnitude, some variables may override other variables and appear to have more significance than they actually do. The training algorithm has to balance for order of magnitude differences by adjusting the network weights, which is not very successful with many of the training algorithms (i.e., back propagation algorithms). There is no universal standard procedure for normalizing inputs and outputs. The method utilized throughout this research is the Min-Max Normalization Method.

Expanding the normalization range so that the minimum value of the normalized variable, xi, norm, is set at zero (0) and the maximum value, xi, norm is set at one (1). We define the normalized variable xi, norm by using the minimum and maximum values of the original variable, xi, min and xi, max, respectively, i.e.

$$X_{i,norm} = \frac{\left(X_i - X_{i,min}\right) \times \left(L_{max} - L_{min}\right)}{X_{i,max} - X_{i,min}} + L_{min} \qquad (2)$$

Where L_{max} and L_{min} are the upper and lower limits of the new output range (0,1). More complex techniques for normalization are given by Masters, 1994, Swingler, 1996, and Dowla & Rogers, 1995. The Min-Max Normalization Method is ideal for this model because it could correspond to the entire range of the transfer function utilized in this research (0, 1), and every input value in the data set has a similar distribution range. A MATLAB code was developed and integrated into the main program to perform the data normalization.

2.4 Post Processing

Typically the output of the neural network is a set of unit less values on a scale between 0 to 1 or -1 to 1. The data must be renormalized to the desired data range, because most of the applications have data ranges outside of the neuron outputs. The output values represent a continuous scale and need to be interpreted as real-world amount with units. Therefore renormalizing the output linearly using equation (3) will achieve this goal,

$$y_{i,renorm} = \frac{\left(y_i - y_{i,min}\right) \times \left(M_{max} - M_{min}\right)}{y_{i,max} - y_{i,min}} + M_{min} \qquad (3)$$

Where y_i represents the output and $y_{i,\,renorm}$ represent the rescaled output.

2.5 Development of the Neural Network Architecture

The ultimate objective of the neural network investigation is to construct networks that present optimal generalization performance. The researchers desire the network to perform well on data that are not integrated in the training set. There are many approaches described in the literature that attempt to accomplish this. The following subsections explain the steps carried out to achieve the optimal neural network architecture for this particular research. Throughout training, the weights and biases of the network are repeatedly updated and altered to optimize the network performance. The performance function utilized for the feed forward networks is the mean square error (MSE) that is the average squared error between the predicted outputs and the experimental outputs, i.e.

© IEOM Society International

$$MSE = \frac{1}{n-1}\sum_{1}^{n}\left(\nu_{predeicted} - \nu_{experimental}\right)^{2} \qquad (4)$$

- Network Interconnection

There are many different types of interconnection between neurons and the layers for instance feed-forward, bi-directional, fully-connected and partially-connected. The feed forward neural network is utilized in this research; where the neurons in a single layer send their signals forward to the next layer. During this process the neurons never receive signals from the layers to the front of them. Multilayer Perceptron (MLP) is the frequently used network and it has been broadly analyzed for which numerous learning algorithms have been reported. The MLP is used in this research because they are flexible, general-purpose, nonlinear models made up with many neurons that are structured into parallel layers. The number of neurons and layers establish the complexity of the MLP network therefore it is very essential to optimize the network structure.

- Number of Epochs

Another parameter that needs to be optimized is the number of epochs. The epoch is defined as a sequence of training data sets presented to the network between weight updates. For enhanced training, the optimum epoch size should be determined because the epoch size is a function of the data in the back propagation training and furthermore an additional motivation for obtaining the optimal number of epochs is that the neural networks can easily over fit causing the error rate of testing to be much larger than the error rate of the training. Therefore, determining the number of epochs is a very significant step. One of the performance measures used is the Mean Square Error (MSE), which calculates the average squared error between the network outputs and the desired output [6]. During training the MSE decreases in the early epochs of the back propagation but after a while it begins to increase. The point of minimum MSE is a good indicator of the best number of epochs. For training the network, the binary datasets are used. The neural network run twelve (12) sessions at 100 epochs interval between each session (100, 200, and 1200) and the MSE of each single run was recorded as presented in Table 2. The performance values (MSE) of the entire training dataset are plotted against the number of epochs as shown in Figure 2. From Table 2 and Figure 2 the lowest MSE value occurs at 400 epochs; therefore, it can be concluded that the network should be "early stopped" at 400 epochs. This criterion will be implemented throughout this research.

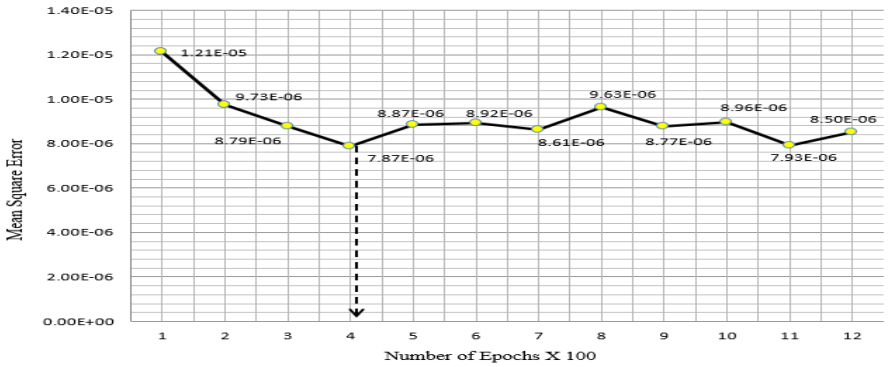


Figure 2. Mean Square Error verses Number of Epochs.

Table 2. Number of Epochs and Mean Square Error of the Neural Networks.

Run	Number of Epochs	Mean Square Error
ANN1	100	1.21E-05
ANN2	200	9.73E-06
ANN3	300	8.79E-06
ANN4	400	7.87E-06
ANN5	500	8.87E-06
ANN6	600	8.92E-06
ANN7	700	8.61E-06
ANN8	800	9.63E-06

© IEOM Society International

ANN9	900	8.77E-06
ANN10	1000	8.96E-06
ANN11	1100	7.93E-06
ANN12	1200	8.50E-06

- Transfer Function

The most common transfer functions are sigmoid, hyperbolic tangent and radial-bias functions. The sigmoid function is used in this research because it is a non-linear transfer function that produces signals within the desired range (0,1). The non-linearity characteristic is an important factor, because in the case of a linear transfer function each input to the neuron is multiplied by the same proportion throughout the training. This might force the whole system to "drift" during training. Therefore a non-linearity in the system assists in isolating specific input pathways [7]. Furthermore the non-linear units have a higher representational power than ordinary linear units. Researchers have shown that a network with a single hidden layer consisting of a sufficient number of non-linear units can approximate any continuous function [8]. The back-propagation algorithm used in this work requires continuous and differentiable transfer functions so it allows for weight update adjustments.

- Number of Neurons

The number of neurons must be determined to achieve the optimal neural network. The input and output neurons correspond to the input parameters which are the mole fraction and the temperature and the desired output of the network in this research is kinematic viscosity. However, the determination of the number of neurons in the hidden layer(s) depends mainly on the application of each specific network. Currently, there is no rule of thumb to determine the optimal number of hidden layers or the number of neurons. Therefore the process is approached with an intensive trial and error technique. During this approach many models are created with different number of hidden neurons while the input and output neurons are fixed. Each network model run and stopped at 400 epochs. The MSE was recorded and a graph of the number of neurons verses performance mean square error MSE was plotted as shown in Figure 3. It is obvious that the more hidden neurons used in the network the superior the performance of the training. In this case the higher performance is not the only criteria required because the network might be overtrained and memorize the training data. For this reason choosing the highest number of neurons based on the MSE criteria might not be an ideal approach for the generalization characteristic of the network.

An additional step should be taken into account to determine the optimal number of the hidden neurons where new network models with 2, 4, 6, 8, 10, 12, 14, 16, and 18 hidden neurons were developed using only one (Heptane-Cyclohexane-Ethylbenzene) data set. The early stopping technique of 400 epochs was applied throughout these networks. At the end of each run, the percent absolute average deviation % AAD was recorded as shown in Table 3 and plotted against the number of the hidden neurons as illustrated in Figure 4. Figure 4 shows clearly that six neurons in one hidden layer produces the optimal values for the data set that has never been introduced to the network during training. Therefore this network has the highest generalization performance and any further training beyond six neurons might lead to over fitting.

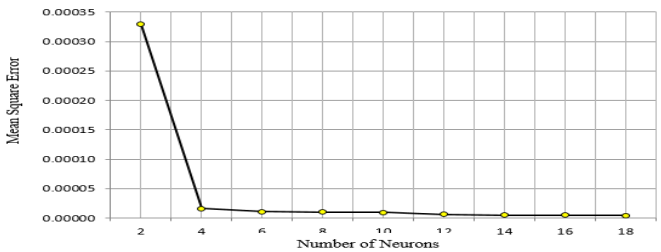


Figure 3. Number of Neurons in the hidden layer verses MSE.

© IEOM Society International

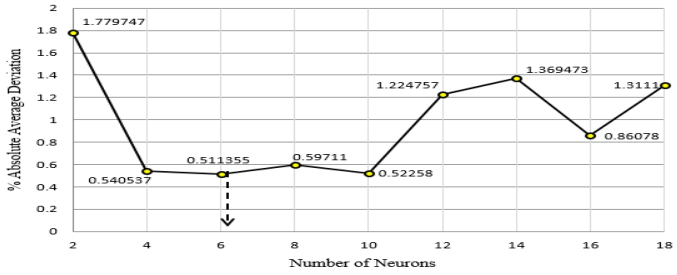


Figure 4. Number of Neurons in the hidden layer verses %AAD.

Table 3. Number of Neurons in the hidden layer verses MSE and %AAD.

	Number of Neurons	Mean Square Error	% AAD
ANN13	2	0.000329	1.779747
ANN14	4	1.62E-05	0.540537
ANN15	6	1.11E-05	0.511355
ANN16	8	1.06E-05	0.59711
ANN17	10	9.42E-06	0.52258
ANN18	12	6.26E-06	1.224757
ANN19	14	5.25E-06	1.369473
ANN20	16	4.82E-06	0.86078
ANN21	18	4.11E-06	1.3111

2.6 Training Methodology

One of most important features that make an ANN accomplish generalization through past experiences is the use of the connections between the neurons. The power of the connection and the bias between two neurons can increase or decrease the signal weight that passes between the neurons. Therefore, the significant inputs should be assigned more weight value and the less significant connections are assigned less weight. The training process achieves the optimal weight and bias for each of the connections between the neurons. Such algorithm is employed in the process which is defined as a procedure for adjusting the weights and biases of the network. The training algorithm is applied to train the network to perform some specific task. The back propagation training algorithm [9] is the most common algorithm used with MLP.

In this research, a variety of training algorithms were tested. The Levenberg-Marquardt training algorithm was used due to its ability to expedite the convergence of the network training. The back-propagation training technique involves the matching of the predicted output from the network with the experimental/desired output. The weights and biases are optimized by iteratively minimizing the output error. After training, the network is described as generalized and retains information of the relationship between the inputs and the outputs and a particular input can produce a suitable output. With the stable weights and biases established through training, the network has the ability to produce a predicted output from a particular input for data that has never been introduced to the network.

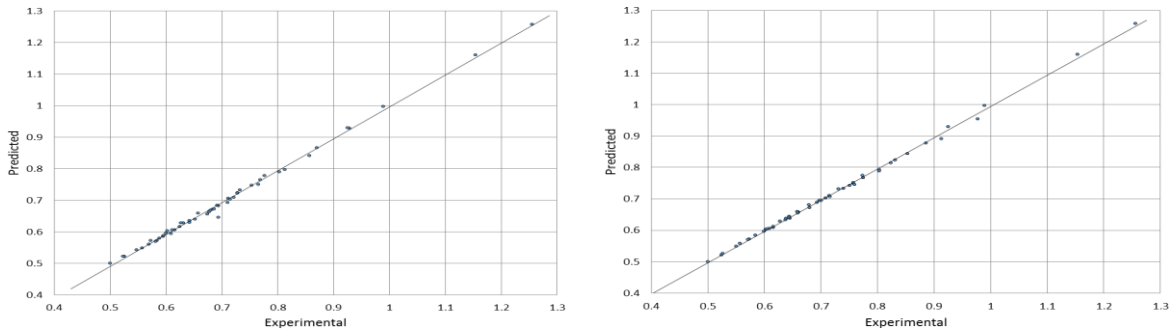
3. Results

The generalized parameters obtained from the neural network models are plotted against the experimental values as shown in Figure 5 to Figure 12. The systems utilized in this work were 10 ternary subsystems, five quaternary subsystems, and the main quinary system. The plots show the efficiency of the neural network model in predicting kinematic viscosity. For both the training and the testing data sets, the neural network model performed extremely well for most of the systems. When the results produced by the neural network were validated against testing data that was not integrated throughout the course of the training, the model showed excellent generalization ability and was able to predict with a satisfactory level of accuracy the kinematic viscosity at the entire temperature range.

© IEOM Society International

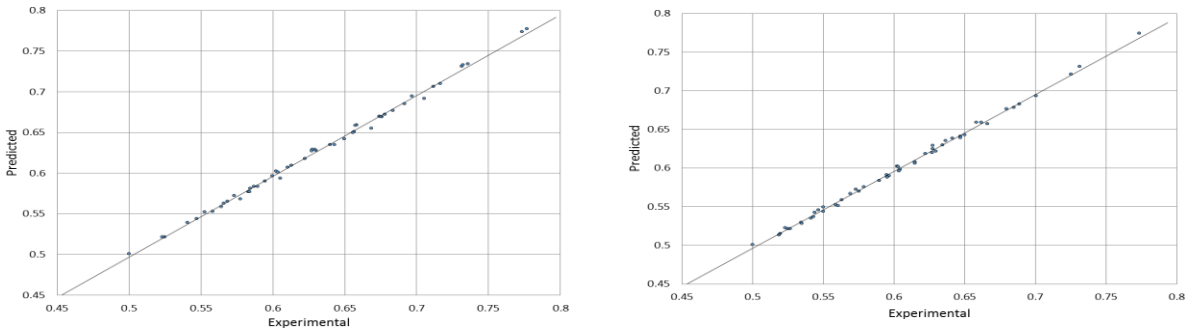
Table 4, shows the predictive performance of the neural network as a % AAD for the ternary subsystems; on average, the neural network was capable of representing the kinematic viscosity as summarized in Table 5. It is thus evident that the generalized neural network model predicts the kinematic viscosity of the majority of the test points of the ternary subsystems with an AAD of less than 1.5%; furthermore, an overall AAD of 0.8646% was also achieved, as illustrated in Figure 13.

Table 6, shows the predictive performance of the neural network as a %AAD of the kinematic viscosity for the quaternary subsystems. In general, five subsystems were utilized for this work, which had a min AAD of 0.49160 % and a max AAD of 1.78264%. The overall AAD for the quaternary subsystems is 1.1298%.



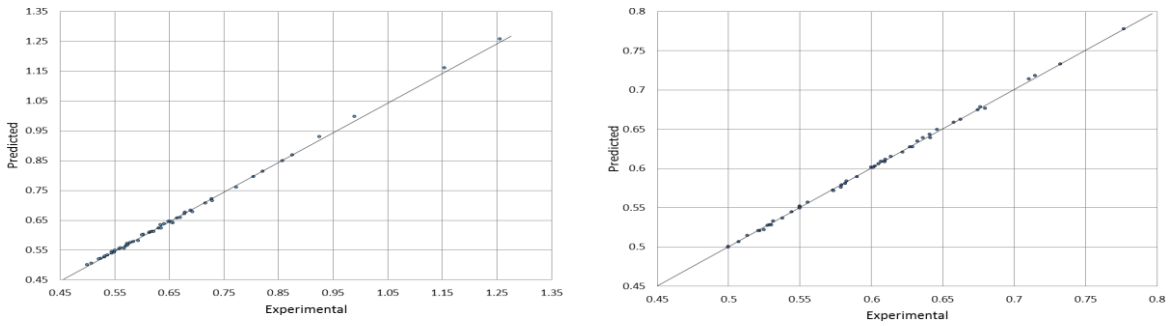
(a) (b)

Figure 5. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Heptane - Cyclohexane - Ethylbenzene" and (b) "Heptane - Octane - Cyclohexane" for the entire temperature range 298 K - 313 K data set.



(a) (b)

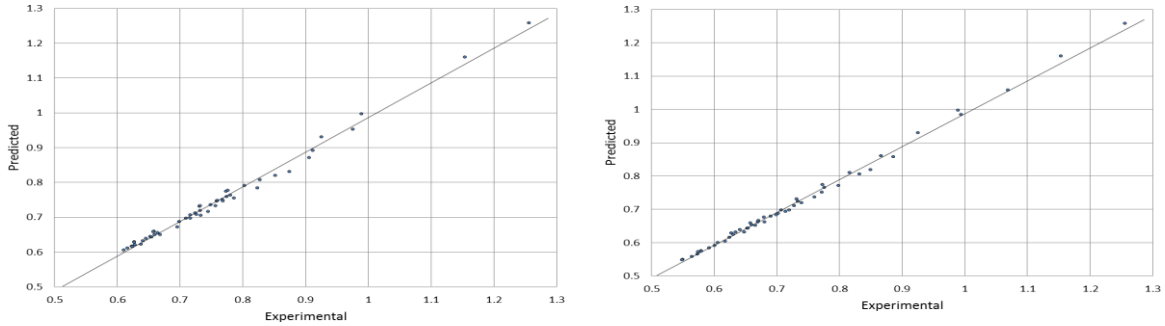
Figure 6. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Heptane - Octane - Ethylbenzene" and (b) "Heptane - Octane - Toluene" for the entire temperature range 298 K - 313 K data set.



(a) (b)

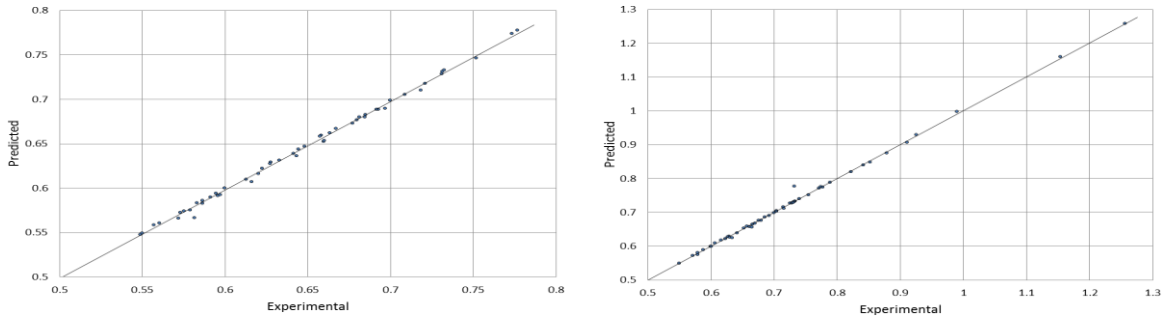
© IEOM Society International

Figure 7. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Heptane - Toluene - Cyclohexane" and (b) "Heptane - Toluene - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.



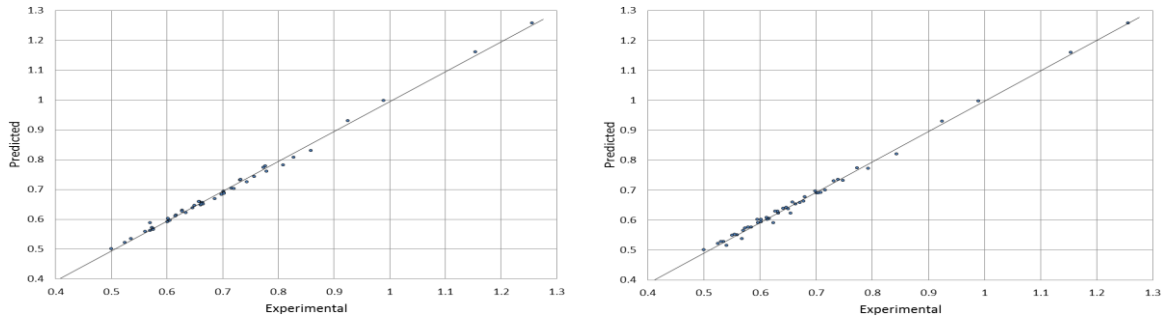
(a) (b)

Figure 8. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Octane - Cyclohexane - Ethylbenzene" and (b) "Octane - Toluene - Cyclohexane" for the entire temperature range 298 K - 313 K data set.



(a) (b)

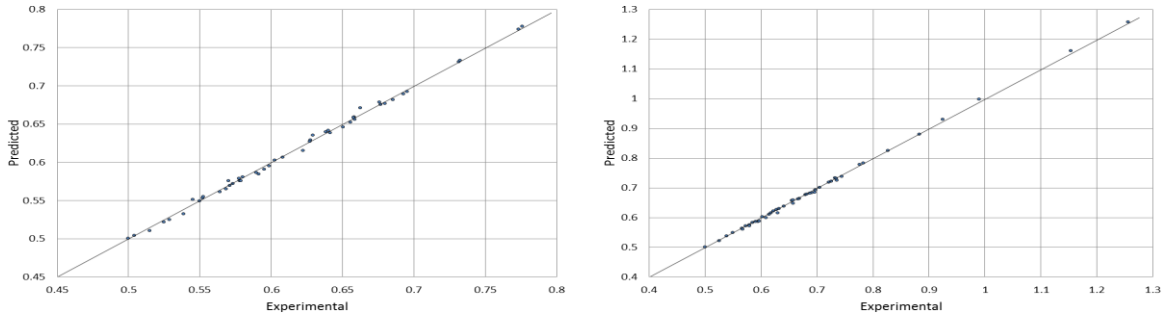
Figure 9. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Octane - Toluene - Ethylbenzene" and (b) "Toluene - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K -313 K data set.



(a) (b)

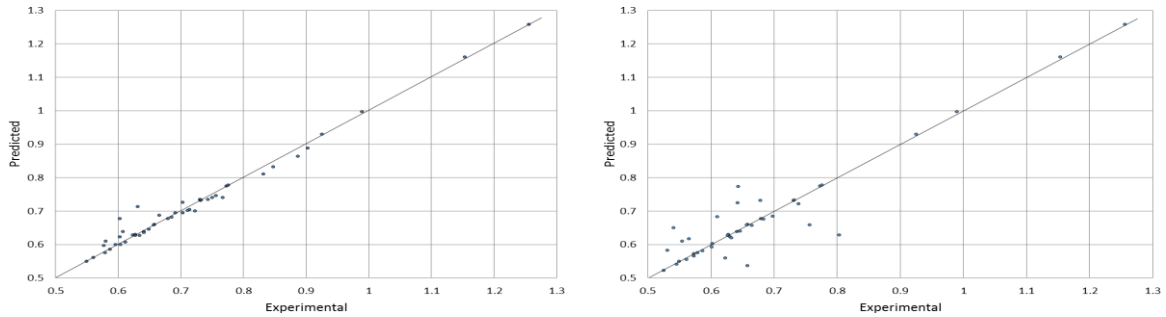
Figure 10. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Heptane-Octane - Cyclohexane - Ethylbenzene" and (b) "Heptane – Octane – Toluene - Cyclohexane" for the entire temperature range 298 K - 313 K data set.

© IEOM Society International



(a) (b)

Figure 11. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Heptane - Octane – Toluene - Ethylbenzene" and (b) "Heptane – Toluene – Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.



(a) (b)

Figure 12. Validation plot of experimental verses predicted values of kinematic viscosity with (a) "Octane - Toluene – Cyclohexane - Ethylbenzene" and (b) "Heptane - Octane –Toluene – Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

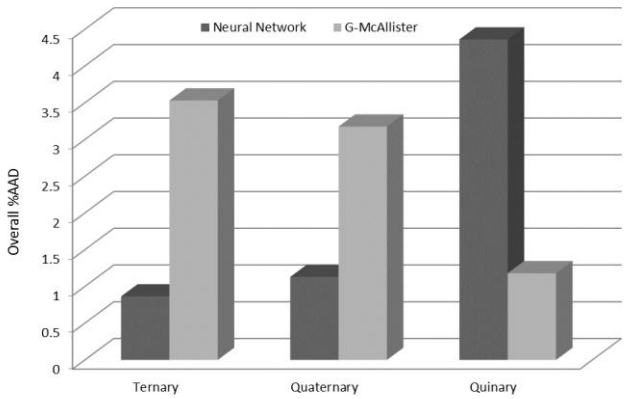


Figure 13. Overall % AAD of Neural Network and General McAllister Models for the Ternary, Quaternary and Quinary systems.

Table 4. The predictive performance of neural network.

System	% AAD	% AAD (Max)	% AAD (Min)	R^2
Heptane – Cyclohexane - Ethylbenzene	1.26010	6.83172	0.03552	0.99398

© IEOM Society International

Heptane - Octane - Cyclohexane	0.69556	2.4363	0.00122	0.99757
Heptane - Octane - Ethylbenzene	0.65752	1.96910	0.02201	0.99338
Heptane – Octane - Toluene	0.74127	1.58725	0.02201	0.99289
Heptane - Toluene - Cyclohexane	0.92114	2.34120	0.01658	0.99773
Heptane – Toluene - Ethylbenzene	0.25856	1.47412	0.01483	0.99928
Octane - Cyclohexane- Ethylbenzene	1.82863	4.98058	0.02201	0.98175
Octane – Toluene - Cyclohexane	1.43718	3.70471	0.02201	0.99237
Octane – Toluene - Ethylbenzene	0.43444	2.52300	0.00770	0.99551
Toluene – Cyclohexane - Ethylbenzene	0.41207	6.16943	0.00088	0.99734

Table 5. Comparison of ternary subsystems % AAD of Neural Network verses Generalized McAllister Model.

System	Temperature (K)	Neural Network % AAD	McAllister Model %AAD
Heptane-Cyclohexane-Ethylbenzene	293.15-313.15	1.26010	4.89
Heptane-Octane-Cyclohexane	293.15-313.15	0.69556	2.49
Heptane-Octane-Ethylbenzene	293.15-313.15	0.65752	3.93
Heptane-Octane-Toluene	293.15-313.15	0.74127	3.37
Heptane-Toluene-Cyclohexane	293.15-313.15	0.92114	5.12
Heptane-Toluene-Ethylbenzene	293.15-313.15	0.25856	2.59
Octane-Cyclohexane-Ethylbenzene	293.15-313.15	1.82863	2.88
Octane-Toluene-Cyclohexane	293.15-313.15	1.43718	2.98
Octane-Toluene-Ethylbenzene	293.15-313.15	0.43444	3.40
Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	0.41207	3.66

© IEOM Society International

Table 6. The predictive performance of neural network.

System	% AAD	% AAD (Max)	% AAD (Min)	R^2
Heptane – Octane – Cyclohexane - Ethylbenzene	1.31412	3.38938	0.02201	0.99366
Heptane – Octane – Toluene - Cyclohexane	1.40294	5.36505	0.02201	0.99301
Heptane – Octane -Toluene - Ethylbenzene	0.49160	1.38247	0.02201	0.99739
Heptane - Toluene - Cyclohexane - Ethylbenzene	0.65777	2.19162	0.01532	0.99868
Octane – Toluene – Cyclohexane - Ethylbenzene	1.78264	12.91999	0.01296	0.98004
Heptane – Octane –Toluene – Cyclohexane - Ethylbenzene	4.36113	21.77939	0.01296	0.88951

4. CONCLUSION

This work has demonstrated that artificial neural networks with moderately simple architecture can be used as a prediction technique in order to determine the kinematic viscosity of multi-component mixtures from the known viscosity of their binary mixture. The 16 systems examined in this work were composed of Heptane, Octane, Toluene, Cyclohexane, and Ethylbenzene and were categorized as binary, ternary, quaternary, and quinary systems. A neural network with six hidden neurons and one output neuron was utilized, and experimental data collected from the literature were used to train and test the network. The overall AADs for the ternary, quaternary, and quinary systems examined in this research were 0.8646 %, 1.1298%, and 4.3611 %, respectively. A comparison with the results produced by the generalized McAllister model showed that the neural network model produced more accurate results in 15 of the 16 systems investigated.

REFERENCES

Cheng, B., & Titterington, D. (1994). Neural Networks: a review from a staistical perspective. Statistical Science, 2-54.

Jasper, J. J. (1972). The Surface Tension of Pure Liquid Compounds. Journal of Physical and Chemical Reference Data, 841.

Al Gherwi, W. (2005). A study of the viscosities and densities of some binary and ternary regular liquid mixtures at different temperatures levels. Windsor: University of Windsor.

Cai, R. (2004). A study of the viscosities and densities of some multi-component regular non-electrolyte solutions at different temperatures. Windsor: University of Windsor.

El Hadad, O. (2004). A study of the viscosities and densities of some multi-component liquid regular solutions at different temperatures. Windsor: University of Windsor.

Demuth, H., Beale, M., & Hagan, M. (2007). Neural Network Toolbox 5. The Mathworks.

Anderson, J. (1995). An Introduction to Neural Networks. Cambridge, MA.: The MIT Press.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 359-366.

Rumelhart, D., Hinton, G., & Williams, R. (Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986).). Learning internal representations by error propagation. Parallel Distributed Processing, 1, 318-362.

© IEOM Society International