Neural Network Viscosity Models for Multi-Component Liquid Mixtures

by

Adel A Elneihoum

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Chemical Engineering

Waterloo, Ontario, Canada, 2009

©Adel A Elneihoum 2009

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

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 the input. A comparison of the experimental values and the results predicted from the neural network revealed a satisfactory correlation, with the overall absolute average deviation (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.

Acknowledgements

I wish to express my sincere thanks to my supervisor, Professor Ali Elkamel, who mentored me and kindly allowed me this outstanding opportunity to achieve my Master of Applied Science. I would like to extend my appreciation to him, who also provided the guidance; useful advises, extensive knowledge, and the help that made my dream come true.

I would also like to thank Dr Mazda Biglari and Professor Xianshe Feng for their time to review my thesis, and presented their guidance and valuable suggestions.

Special thanks to Walid El Garwi for being generous with his assistance that he provided during this work. My thanks go also to Professor A Asfour for many helpful discussions.

Last, but certainly not least, my heartfelt appreciation to my father, Abubaker and mother, Fatma, for your constant prayers and support. Expressions of gratitude are not adequate to express how much I am indebted to you.

Finally, I would like to thank my dear wife, Wegdan, and my children the joy of my life, Fatma, Abubaker, Heba, Lena and Sarah for their patience, understanding, and encouragement that they provided throughout this journey.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background and Objectives

The viscosity of liquids is a measure of a liquid's internal resistance to flow. This property is one of the significant transport properties essential for many scientific studies and practical applications. In many chemical applications, such as separation, fluid flow, mass transfer, heat transfer, and oil recovery, prior knowledge of thermodynamics and the transport properties of multicomponent mixtures are extremely important, especially in the early stages of designing a chemical process. However, while extensive data have been published in the literature with respect to the thermodynamic and transport properties of binary liquid mixtures, far fewer measurements of the kinematic viscosity of ternary, quaternary, and quinary liquid mixtures have been reported. It is therefore essential and of significant practical value to develop a relatively simple model that can utilize the existing information about binary solutions to accurately predicts the kinematic viscosity of ternary, quaternary, and quinary solutions.

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.

Unfortunately, the behavior of the molecules can, in general, limit an accurate prediction of the thermodynamic and transport properties of the mixtures. These mixtures are typically difficult to model without a large number of data, which are often unavailable for a particular mixture. As well, the composition of a liquid mixture generally has a major effect on its viscosity.

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) 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.
- Compare the results of these neural network models with the results obtained from experimental tests.

- Evaluate the results produced by the neural networks by comparing them with those produced by the generalized McAllister model, which is one of the most reliable predictive models reported in the literature.

1.2 Organization of the Thesis

Chapter 2 of this thesis focuses on the principle of viscosity theory, Eyring's equation of absolute theory, the McAllister models, the technique developed by Asfour and co-workers (Asfour et al., 1991) and the generalized McAllister model, and also introduces the effective carbon number (ECN).

Chapter 3 presents artificial neural networks, including a historical overview; the definitions used by a variety of researchers; and their fundamentals, architecture, and type of training.

Chapter 4 is devoted to the development of an artificial neural network for viscosity prediction: the research procedure, the methodology, the software used, and the data selection, normalization, and post-processing. Also described are the proposed structures and design of the ANN, which involve the selection of the network interaction, the number of epochs, the transfer function, and the number of neurons utilized.

Chapter 5 is divided into two parts: the first part presents and discusses the results produced by the ANN, and the second provides a comparison of the results from the ANN with those produced by the generalized McAllister model.

Chapter 6 summarizes the accomplishments of the thesis, explains the contributions of the research, and provides suggestions for future work.

Chapter 2
Development of the Generalized McAllister Theory

2.1 Introduction

Viscosity in simplest terms can be defined as the resistance of a fluid to deformation, and could also be thought of as the measure of internal fluid frictional forces which tend to resist any dynamic transformation in the fluid movement. However, a Newtonian fluid has a linear relationship between the shear stress per unit area at any point and the velocity gradient that can be illustrated in Figure 2.1, which is called the constant of proportionality μ and also known as the dynamic viscosity of the liquid,

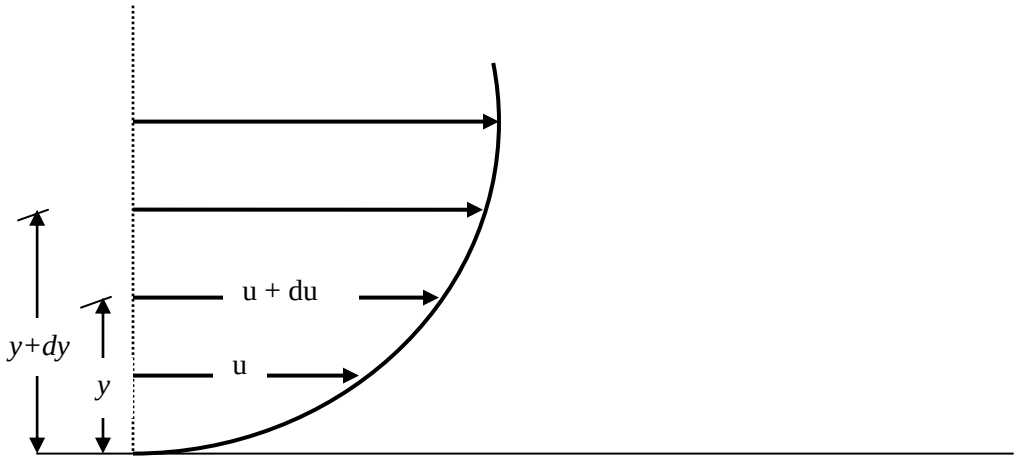

Figure 2.1: The relationship of shear stress and velocity gradient for liquids

$$\mu = \tau / \frac{du}{dy} \tag{2.1}$$

where

τ = shear stress per unit area and, du/dy is the Velocity Gradient.

μ has the dimension of (force)(time)/(length)^2 or, also (mass)/(length) (time).

The ratio of the dynamic viscosity to the density is called the kinematic viscosity,

$$\nu = \frac{\mu}{\rho} \tag{2.2}$$

with dimension (length)/(time)^2.

2.2 Viscosity of Pure Liquids

The literature addressing liquid viscosity is extremely broad. However, because of the limited knowledge of the nature of the liquid state there is no theory, yet, that would allow the calculation of liquid viscosities without empiricism despite the many attempts to explain theoretically liquid viscosities in terms of temperature or chemical constitution. Therefore, one had to rely on empirical estimation techniques (Pedersen, et al., 1984 and Andrade & Da, 1934). Most relations correlate the viscosity of liquids with temperature since the viscosity of liquids is not often affected by moderate pressure. The viscosity of liquids decreases as temperature increases. Andrade correlation (Andrade & Da, 1934) expresses this and it is the most frequently used temperature dependence of the viscosity relationship:

$$\mu = Ae^{B/T} \tag{2.3}$$

where A and B are constants and T is the absolute temperature. The Andrade equation was initially considered to be applicable over the whole temperature range between the freezing and boiling points. Unfortunately, the equation fails near the freezing temperature, because the logarithm of viscosity plotted versus 1/T becomes nonlinear near the freezing temperature and the values of A and B are typically found through a linear plot. Various approaches were considered to overcome this situation that led to many attempts to correlate between A and B in equation (2.3) with latent heat of vaporization, vapour pressure or other physical properties (Ewell & Eyring, 1937; Nissan et al., 1940).

2.3 Eyring Equation-The Absolute Reaction Rate Theory

Eyring and coworkers (Ewell & Eyring, 1937), (Glasston, et al., 1941) applied the theory of absolute reaction rates and proposed a description for the liquid viscosity phenomena. They considered the viscous flow as a chemical reaction wherein the fundamental process is the movement of an individual molecule from one equilibrium position to another crossing a potential energy barrier. Eyring suggested two layers of molecules in a liquid, spaced by λ_1, and proposed that a shear among two layers of liquid, forces the individual molecule to shift from one equilibrium position to another. The movement process needs a hole or a site with area λ_2 λ_3 which is considered as the average area typically occupied by a molecule as shown in Figure 2.2

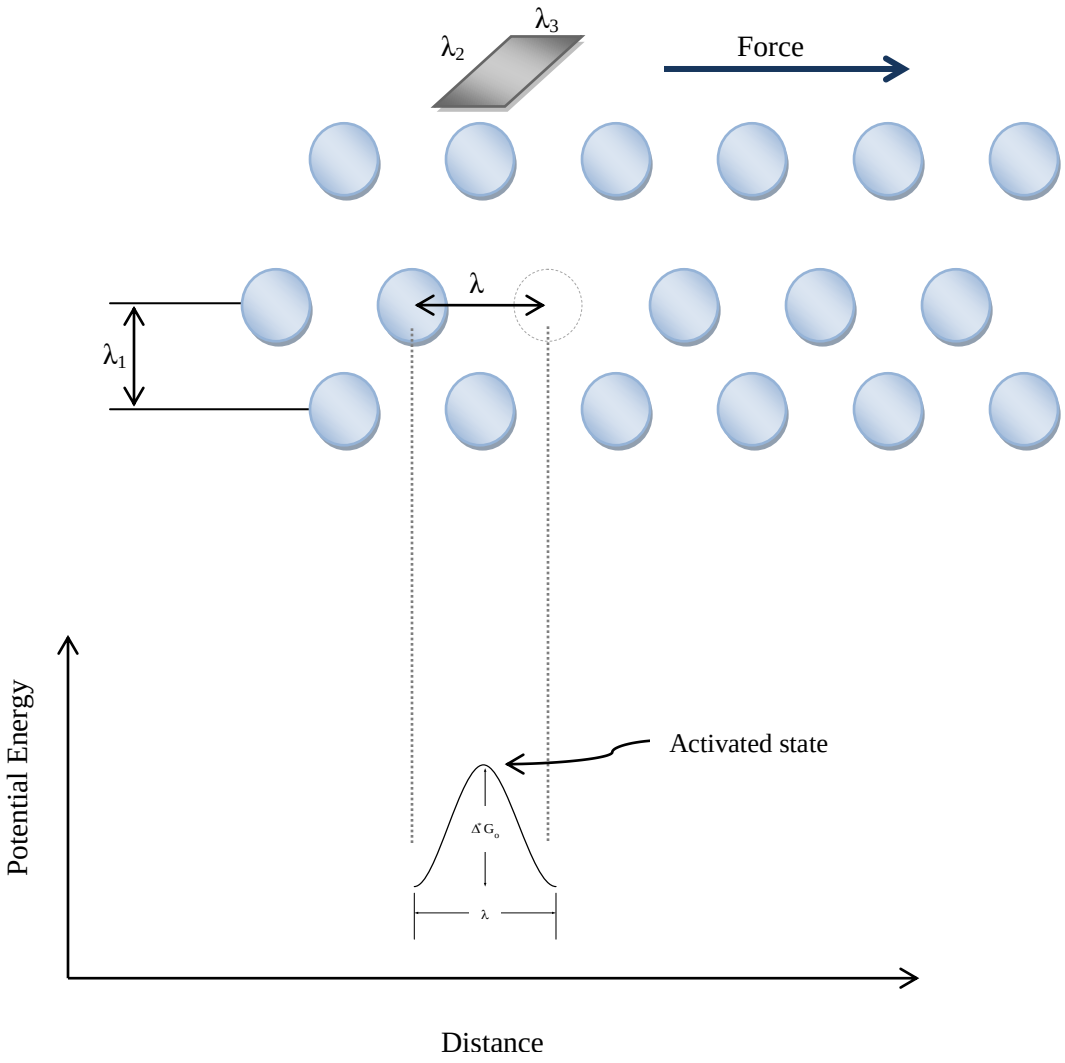

Figure 2.2 : Cross section of an idealized liquid illustrating the fundamental rate process involved in viscous flow. (Hirschfelder, Curtiss, & Bird, 1954)

The creation of a hole requires the expenditure of energy given that work must be done in pushing around other molecules. Eyring considered these transfer of molecules as chemical reactions in which a potential free energy ΔG is needed for such process (Figure 2.2).

In the stress-free state, which represents the position of minimum potential energy, Eyring suggested that molecules are continuously vibrating with frequency for the forward and backward molecules jumps as

$$r_0 = \frac{kT}{h}\exp\left(-\frac{\Delta G_0}{KT}\right) \tag{2.4}$$

where k is Boltzman's constant, h is Plank's constant, and T is absolute temperature. When a shear stress is applied, the total force acting on the molecule is f, and has the magnitude of f λ_2 λ_3. It is suggested that the only mechanical work generated is to transfer the molecule to the top of the energy barrier traveling a distance of $\lambda/2$ where λ is the space separating two molecules. On the other side of the energy barrier it is suggested that the molecules lose energy as heat. Therefore the work is given by:

$$\text{Work} = f\lambda_1\lambda_2\left(\frac{\lambda}{2}\right) \tag{2.5}$$

The forward and backward rates are equal when the system is in a steady state. However, when there is an outside force, one direction is favoured. The hopping rate in the forward direction is r_f

$$r_f = \left(\frac{kT}{h}\right)exp\left(-\frac{\Delta G_0^* - f\lambda_1\lambda_2\frac{\lambda}{2}}{KT}\right) \tag{2.6}$$

and the rate of backward direction r_b is:

$$r_b = \left(\frac{kT}{h}\right)exp\left(-\frac{\Delta G_0^* + f\lambda_1\lambda_2\frac{\lambda}{2}}{KT}\right) \tag{2.7}$$

for a single hopping the net rate r_{net} is r_f-r_b

$$r_{net} = \left(\frac{kT}{h}\right) exp\left(-\frac{\Delta G_0^*}{KT}\right)\left[exp\left(\frac{f\lambda_2\lambda_3\lambda}{2KT}\right) - exp\left(-\frac{f\lambda_2\lambda_3\lambda}{2KT}\right)\right] \tag{2.8}$$

rearranging equation (2.8) yields the following:

$$r_{net} = \left(\frac{kT}{h}\right) exp\left(-\frac{\Delta G_0^*}{KT}\right)\left[exp\left(2\frac{f\lambda_2\lambda_3\lambda}{2KT}\right)\right] = \left(\frac{f\lambda_2\lambda_3\lambda}{h}\right) exp\left(-\frac{\Delta G_0^*}{KT}\right) \tag{2.9}$$

The following equation represents the velocity gradient between two molecular layers with distance λ_1 apart.

$$\text{Velocity Gradient} = \frac{\text{Velocity Difference}}{\lambda_1} \tag{2.10}$$

$$\text{Velocity Gradient} = \frac{\text{distance per jump} \times \text{number of jumps per second}}{\lambda_1} \tag{2.11}$$

$$\text{Velocity Gradient} = \frac{\lambda r}{\lambda_1} \tag{2.12}$$

By definition the absolute viscosity η is

$$\eta = \frac{\text{Shear Stress}}{\text{Velocity Gradient}}$$

(2.13)

By substituting Equation (2.12) into Equation (2.13)

$$\eta = \frac{f\lambda_1}{\lambda r} \tag{2.14}$$

And utilising equation (2.9) into equation (2.14) yields

$$\eta = \frac{h\lambda_1}{\lambda_2\lambda_3\lambda^2} exp\left(\frac{\Delta G_0^*}{KT}\right) \tag{2.15}$$

Finally assuming that λ_1 and λ are of the same order and λ_1 λ_2 λ_3 is the volume occupied by a single molecule V_0, and rewriting equation (2.15) as follows:

$$\eta = \frac{h}{V_0} exp\left(\frac{\Delta G_0^*}{KT}\right) \tag{2.16}$$

Molar volume V_m is introduced to the equation as follows:

$$\eta = \frac{hN}{V_m} exp\left(\frac{\Delta G^*}{KT}\right) \tag{2.17}$$

Where N is the Avogadro's number and ΔG^* is the molar activation energy of the viscous flow.

2.4 McAllister Model

2.4.1 The McAllister Three-body Interaction Model

McAllister (McAllister, 1960) applied Eyring's equation (2.4) of shear viscosity to develop a model for kinematic viscosity of binary liquid mixtures,

$$\nu = \frac{hN}{M} exp\left(\frac{\Delta G^*}{KT}\right) \tag{2.18}$$

McAllister assumed for a binary liquid mixture consisting of two types of molecules, type 1 and type 2, as molecule 1 shifts over the energy barrier; it may interact mostly with type 1, with type 2 or with some combinations of 1 and 2, depending on the neighbourhood concentration. Furthermore, McAllister (McAllister, 1960) proposed that the interaction could be three-body or four-body interaction.

The four-body type is more complicated than the three-body type. The three body type interaction is suggested to have six interactions 1-1-1, 1-1-2, 1-2-1, 1-2-2, 2-1-2, and 2-2-2 as illustrated in Figure 2.3.

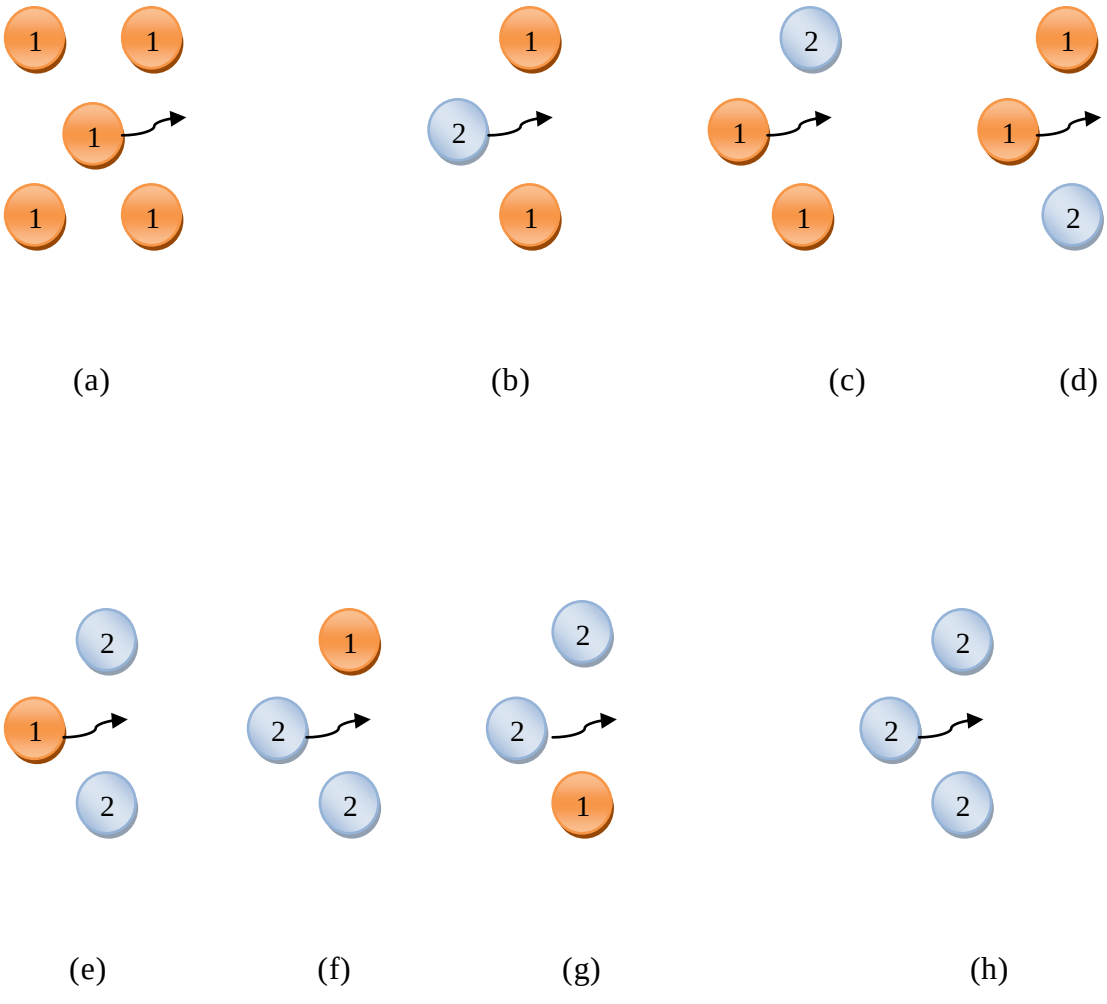

Figure 2.3 : Types of a three-body model viscosity interactions in a binary mixture.

The free energies of activation for viscosity are the summation of the mole fractions based on the additional assumption that interactions are dependent on concentration.

The free energy of activation of the mixture can be expressed in the general form of

$$\Delta G^{*} = \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} x_i x_j x_k \Delta G^{*}_{ijk} \tag{2.19}$$

Expanding equation (2.19) would yield the following:

$$\Delta G^{*} = x_1^3\,\Delta G_1^{*} + x_2^3\,\Delta G_2^{*} + x_1^2 x_2\,\Delta G_{121}^{*} + 2x_1^2 x_2\,\Delta G_{112}^{*} + x_1 x_2^2\,\Delta G_{212}^{*} + 2x_1 x_2^2\,\Delta G_{112}^{*} \tag{2.20}$$

where x represents the mole fraction.

More assumptions were made by McAllister, namely:

$$\Delta G_{121}^{*} = \Delta G_{112}^{*} = \Delta G_{12}^{*} \tag{2.21}$$

$$\Delta G_{212}^{*} = \Delta G_{122}^{*} = \Delta G_{21}^{*} \tag{2.22}$$

Substituting Equations (2.21) and (2.22) into equation (2.20) give the following:

$$\Delta G^{*} = x_1^3\,\Delta G_1^{*} + 3x_1^2 x_2\,\Delta G_{12}^{*} + 3x_1 x_2^2\,\Delta G_{21}^{*} + x_2^3\,\Delta G_2^{*} \tag{2.23}$$

For each type of activation considered in equation (2.23), a corresponding kinematic viscosity can be assigned based on equation (2.19).

The kinematic viscosity of the mixture

$$\nu = \left(\frac{hN}{M_m}\right) e^{\Delta G^{*}/RT} \tag{2.24}$$

where $M_m = \sum_i x_i M_i$ (2.25)

The kinematic viscosity for pure component 1is given by the following equation:

$$\nu_1 = \left(\frac{hN}{M_1}\right) e^{\Delta G_1^* / RT} \tag{2.26}$$

For interactions of type b, c and d in Figure (2.4), ν_{12} is defined as:

$$\nu_{12} = \left(\frac{hN}{M_1}\right) e^{\Delta G_{12}^* / RT} \tag{2.27}$$

Given that these interactions involve two molecules of type 1 and one of type 2, therefore M_{12} is:

$$M_{12} = (2M_1 + M_2)/3 \tag{2.28}$$

For interactions of type e, f, and g in Figure (2.4), ν_{21} is:

$$\nu_{21} = \left(\frac{hN}{M_1}\right) e^{\Delta G_{21}^* / RT} \tag{2.29}$$

Given that these interactions involve two molecules of type 2 and one of type 1, therefore $M_{21}$is

$$M_{21} = (M_1 + 2M_2)/3 \tag{2.30}$$

And for pure component 2

$$\nu_2 = \left(\frac{hN}{M_1}\right) e^{\Delta G_2^* / RT} \tag{2.31}$$

Substituting equation (2.23) into equation (2.24) yields

$$\nu = \left(\frac{hN}{M_m}\right) e^{\left(x_1^3 \Delta G_1^* + 3x_1^2 x_2 \Delta G_{12}^* + 3x_1 x_2^2 \Delta G_{21}^* + x_2^3 \Delta G_2^*\right)/RT} \tag{2.32}$$

Taking the natural logarithm of equations (2.26), (2.27), (2.29), (2.31), and (2.32), and combining them together to eliminate the free energy of activation and rearranging yields the well-known McAllister three-body model:

$$\ell n\nu = x_1^3\ell n\nu_1 + 3x_1^2x_2\ell n\nu_{12} + 3x_1x_2^2\ell n\nu_{21} + x_2^3\ell n\nu_2 - \ell n\left[x_1 + x_2M_2/M_1\right] + 3x_1^2x_2\ell n[(2+M_2/M_1)/3] + 3x_1x_2^2\ell n[(1+2M_2/M_1)/3] + x_2^3\ell n\left[M_2/M_1\right] \quad (2.33)$$

The McAllister model contains two interaction parameters ν_{12} and ν_{21} that should be determined via experimental work. The equation is cubic and includes the possibility of having maximum, a minimum, neither or both.

2.4.2 The McAllister Four-body Interaction Model

McAllister followed the same procedure for developing another equation that he called a four-body interaction model. This model is applicable when one molecule is much larger than the other molecules in the mixture with a ratio greater than 1.5 of the size.

$$\begin{aligned}\ell n\nu = {} & x_1^4\ell n\nu_1 + ax_1^3x_2\ell n\nu_{1112} + 6x_1^2x_2^2\ell n\nu_{1122} \\ & + 4x_1x_2^3\ell n\nu_{2221} + x_2^4\ell n\nu_2 - \ell n\left(x_1 + x_2\frac{M_2}{M_1}\right) \\ & + 4x_1^3x_2\ell n\left(\frac{3+M_2/M_1}{4}\right) + 6x_1^2x_2^2\ell n\left(\frac{1+M_2/M_1}{2}\right) \\ & + 4x_1x_2^3\ell n\left(\frac{1+3M_2/M_1}{4}\right) + x_2^4\ell n\left(M_2/M_1\right)\end{aligned} \quad (2.34)$$

The four body model has three interaction parameters ν_{1112}, ν_{1122}, and ν_{2221} that also have to be determined by experimental work. Determination of these interaction

parameters experimentally is a major drawback for the McAllister model. Based on these interaction parameters, the McAllister models are classified as correlative models and not predictive models. Another setback is that the experiments are time consuming and expensive to conduct.

2.5 Asfour et al. (1991) Technique.

Many researchers regarded the McAllister model as the best correlating technique available for liquid binary and ternary liquid systems (Reid et al., 1987). Although many realized the importance of the McAllister model, Asfour et al., (1991) acknowledged that the model requires experimental data for the interaction parameters and considered that this is a major drawback in the application of McAllister model. Consequently, they developed a method for predicting the values of the McAllister model interaction parameters utilizing pure component viscosity and molecular parameters for binary n-alkane liquid mixtures. Nhaesi and Asfour (1998, 2000a and 2000b) expanded the method for the binary regular solutions and multi-component liquid mixtures.

Asfour et al. (1991) proposed that to predict the McAllister binary interaction parameters in the case of a three-body interaction model it is assumed that:

$$v_{12} \propto (v_1 v_1 v_2)^{1/3} \tag{2.35}$$

$$v_{21} \propto (v_1 v_2 v_2)^{1/3} \tag{2.36}$$

And by plotting the lumped parameters $\nu_{12} / (\nu_1\nu_1\nu_2)^{1/3}$ versus $[(N_1-N_2)^2/ (N_1N_2)^{1/3}]$, where N_1 and N_2 are the number of carbon atoms per molecule of components 1 and 2, respectively, a straight line is obtained as shown in Figure 2.4.

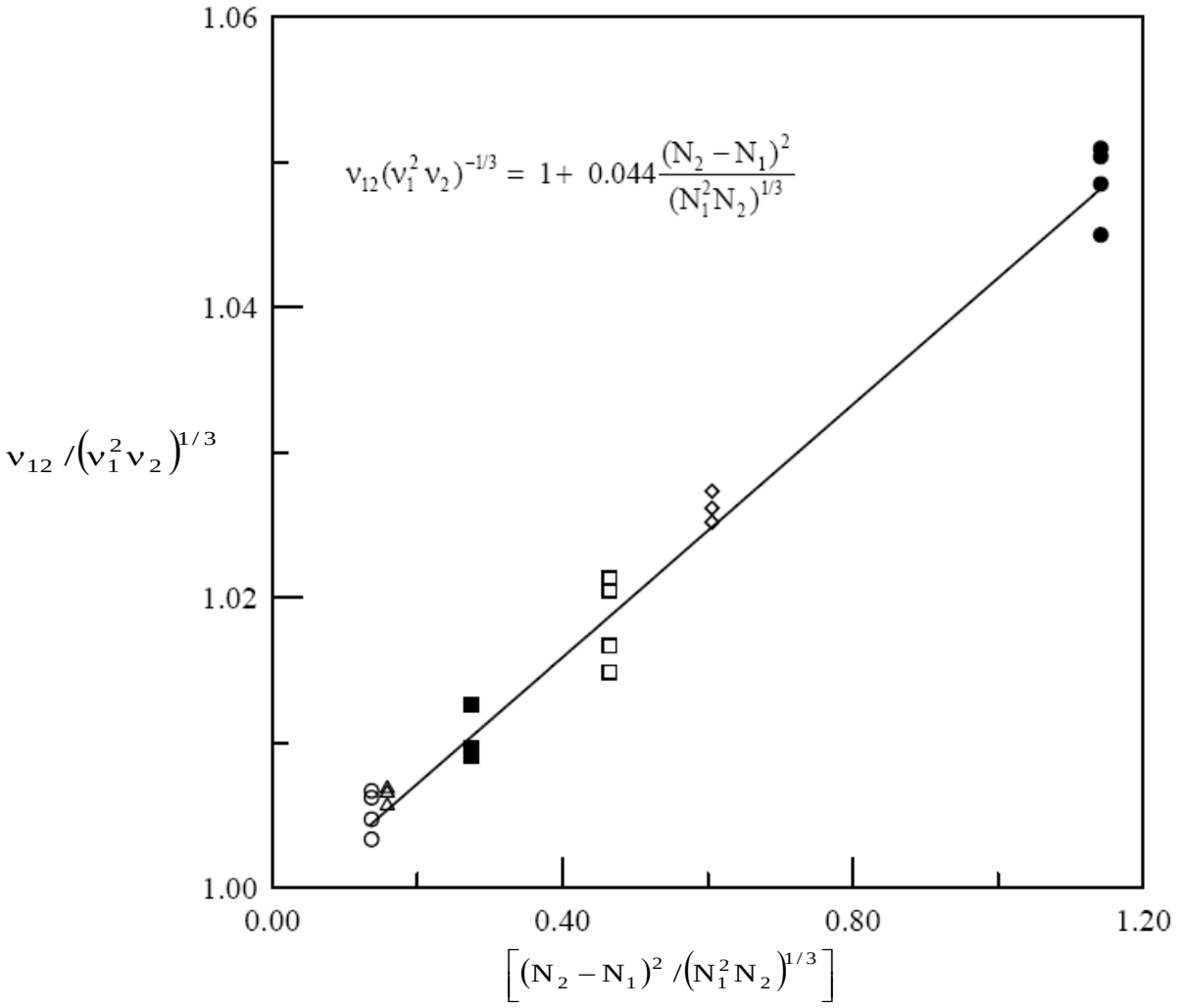

Figure 2.4: Variation of the Lumped Parameter $\nu_{12}/(\nu_1^2\nu_2)^{1/3}$ with $[(N_2-N_1)^2/(N_1^2N_2)^{1/3}]$ (Asfour *et al.* 1991).

where

△ Hexane- Heptane
◇ Hexane- Octane
○ Heptane- Octane
● Heptane- Decane
■ Tetradecane-Hexadecane
□ Octane-Decane

For the straight-line shown in Figure 2.4 the following equation is obtained by linear fitting

$$\frac{\nu_{12}}{\left(\nu_1^2\nu_2\right)^{1/3}} = 1 + 0.044\frac{\left(N_2 - N_1\right)^2}{\left(N_1^2 N_2\right)^{1/3}} \tag{2.37}$$

The McAllister model interaction parameter ν_{12} can be determined from Equation (2.37), where only the number of carbon atoms and the viscosities of pure components are required. Furthermore, Asfour *et al*. (1991) also demonstrated that the interaction parameter, ν_{21}, can be determined with the help of the following equation:

$$\nu_{21} = \nu_{12}\left(\frac{\nu_2}{\nu_1}\right)^{1/3} \tag{2.38}$$

where ν_1 and ν_2 are the kinematic viscosities of components 1 and 2 respectively and ν_{12} could be estimated from equation (2.37).

Asfour *et al*. (1991) stated that the McAllister four-body model should be employed for better outcome when the difference between the carbon numbers of the two components in a binary mixture is equal to or larger than 4 $\left|N_2 - N_1\right| \geq 4$. Therefore,

the same mechanism is followed to create an equation for the prediction of the McAllister four-body model.

$$\frac{\nu_{1122}}{\left(\nu_1^2\nu_2\right)^{1/3}} = 1 + 0.03\frac{\left(N_2 - N_1\right)^2}{\left(N_1^2 N_2\right)^{1/3}} \tag{2.39}$$

Where based on Asfour *et al*. (1991):

$$\nu_{1112} \propto \left(\nu_1^3\nu_2\right)^{1/4} \tag{2.40}$$

$$\nu_{1122} \propto \left(\nu_1^2\nu_2^2\right)^{1/4} \tag{2.41}$$

$$\nu_{2221} \propto \left(\nu_1\nu_2^3\right)^{1/4} \tag{2.42}$$

in the same way, the following equations were suggested:

$$\nu_{1112} = \nu_{1122}\left(\frac{\nu_2}{\nu_1}\right)^{1/3} \tag{2.43}$$

$$\nu_{2221} = \nu_{1122}\left(\frac{\nu_2}{\nu_1}\right)^{1/3} \tag{2.44}$$

Asfour *et al*. (1991) fitted new data into the models collected from literature that have not been used to develop this model, and concluded that the new model outperformed the existing techniques.

2.6 The Generalized McAllister Three-Body Model

Nhaesi and Asfour (2000b) proposed a generalized McAllister three-body interaction model for multi-component liquid mixtures. They also reported a method for predicting the ternary interaction parameters of the McAllister's model. Utilizing only pure compounds information, the generalized model allows the user to predict the viscosity of liquid mixtures of any number of components. This extends the applications of the model beyond the typical binary and ternary systems.

Assuming three-body interactions and that the free energies of activation for viscous flow are additive on the basis of a mole fraction, Nhaesi and Asfour (2000b) reported the following equation for the activation energy of a multi-component system

$$\Delta G_m = \sum_{i=1}^{n} x_i^3 \Delta G_i + 3\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^2 x_j \Delta G_{ij} + 6\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} x_i x_j x_k \Delta G_{ijk} \qquad (2.45)$$

where n is the number of components in the mixture.

Furthermore, the following additional assumptions were made:

$$\Delta G_{iji} = \Delta G_{iij} = \Delta G_{ij} \qquad (2.46)$$

$$\Delta G_{jij} = \Delta G_{ijj} = \Delta G_{ji} \qquad (2.47)$$

For each type of activation energy, substituting the above expressions into equation (2.45) a corresponding Arrhenius type kinematic viscosity equation is obtained. The kinematic viscosity of the mixture is as follows:

$$\nu_m = \frac{hN}{M_{avg}} e^{(\Delta G_m / RT)} \qquad (2.48)$$

where

$$M_{avg} = \sum_{i=1}^{n} x_i M_i \tag{2.49}$$

The kinematic viscosity of pure component I is:

$$\nu_i = \frac{hN}{M_i} e^{(\Delta G_i / RT)} \tag{2.50}$$

The kinematic viscosity of the binary interactions parameters is:

$$\nu_{ij} = \frac{hN}{M_{ij}} e^{(\Delta G_{ij} / RT)} \tag{2.51}$$

Where M_{ij} is represented by the following equation:

$$M_{ij} = (2M_i + M_j)/3 \tag{2.52}$$

and, the kinematic viscosity of the ternary interactions parameters are:

$$\nu_{ijk} = \frac{hN}{M_{ijk}} e^{(\Delta G_{ijk} / RT)} \tag{2.53}$$

where

$$M_{ijk} = (M_i + M_j + M_k)/3 \tag{2.54}$$

In order to eliminate the free energies of activation in equation (2.43), we take the logarithms of equations (2.48), (2.50), (2.51), and (2.53) and substitute them into equation (2.45) to get the following form of the McAllister's three-body interaction model for n-component (multi-component) liquid mixtures (Asfour 2000b):

$$\ell n\nu_m = \sum_{i=1}^{n} x_i^3 \ell n(\nu_i M_i) + 3\sum_{\substack{i=1 \\ i\neq j}}^{n}\sum_{j=1}^{n} x_i^2 x_j \ell n(\nu_{ij} M_{ij}) + 6\sum_{\substack{i=1 \\ i\neq j\neq k}}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} x_i^2 x_j x_k \ell n(\nu_{ijk} M_{ijk}) - \ell n(M_{avg}) \quad (2.55)$$

Equation (2.55), can predict the kinematic viscosity of n-component mixtures by employing the values of binary and ternary interaction parameters only. As mentioned earlier, the kinematic viscosity of the pure components and their molecular parameters determine the values of binary and ternary interaction parameters. For the estimation of the number of binary and ternary parameters needed for a particular n-component mixture system, Nhaesi and Asfour (2000b) proposed the following equation for the number of the binary interaction parameters:

$$N_2 = \frac{n!}{(n-2)!} \quad (2.56)$$

where n is the number of components in the mixture. The following equation is for the number of ternary interaction parameters:

$$N_3 = \frac{n!}{3!(n-3)!} \quad (2.57)$$

For the ternary interaction parameters, Nhaesi and Asfour (2000b) reported the following equation for the prediction of the ternary interaction parameters:

$$\frac{\nu_{ijk}}{(\nu_i \nu_j \nu_k)^{1/3}} = 0.9637 + 0.0313\frac{(N_k - N_i)^2}{N_j} \quad (2.58)$$

Consequently, Nhaesi and Asfour (2000b), in addition to extending the McAllister model to multi-component mixtures, also converted the model into a predictive technique that requires only pure components' kinematic viscosities and their molecular parameters.

2.6.1 Effective Carbon Number (ECN)

Nhaesi and Asfour (1998) extended the Asfour *et al*. (1991) technique to regular binary liquid solutions. In order to achieve this, they developed a technique to calculate the "effective carbon numbers" (ECN) of compounds other than *n*-alkanes. They prepared a semi-log plot of the kinematic viscosities of pure liquid *n*-alkane hydrocarbons at 308.15 K against their carbon numbers (Nhaesi and Asfour, 1998).

A straight-line is obtained as illustrated in Figure 2.5. The line is represented by the following equation:

$$\ell n(\nu) = -1.943 + 0.193(\mathrm{ECN}) \qquad (2.59)$$

where ν in Centistokes (cSt).

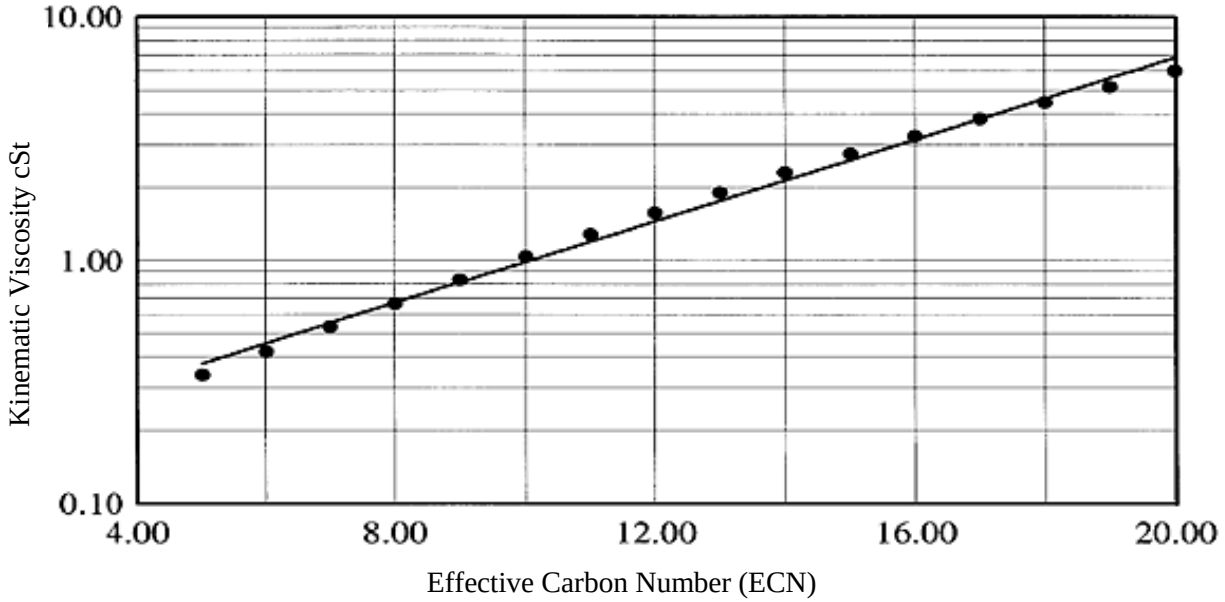

Figure 2.5: Experimental kinematic viscosity for n-alkanes versus the effective carbon number (Nhaesi & Asfour, 1998).

Nhaesi and Asfour (1998) pointed out that utilizing equation (2.59) and employing the kinematic viscosity of any regular compound at 308.15 K could determine the effective carbon number (ECN) of that compound. This allowed them to develop an equation similar to the equation they had developed earlier for *n*-alkane systems. The equation is as follows:

$$\frac{\nu_{12}}{\left(\nu_1^2\nu_2\right)^{1/3}} = 0.8735 + 0.0715\frac{\left(\mathrm{ECN}_2 - \mathrm{ECN}_1\right)^2}{\left(\mathrm{ECN}_1^2\mathrm{ECN}_2\right)^{1/3}} \tag{2.60}$$

Chapter 3
Artificial Neural Networks

3.1 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 (Cheng and Titterington, 1994). Artificial neural networks can deal successfully with non-linearity, handle noisy or irregular data, correlate hundreds of variables or parameters, and provide generalized solutions (Jasper, 1972).

3.2 Historical overview of ANNs

In 1943, McCullough and Pitts launched the first mathematical model of a single idealized neuron. Drawing inspiration from the human brain, they have proposed a general theory of information processing based on networks of neurons. They showed that a network like neural networks of liner threshold elements could compute any logical function. Fundamentally, this paper had a more pronounced effect on computer science than on neural networks (Rabaey, 1995). Canadian neuroscientist Donald Hebb in 1949 produced a major study on learning and memory that suggested neurons in the brain actually change strength through repeated use, and therefore a network configuration could learn. In 1957, Frank Rosenblatt built a hardware neural net called Perceptron that was capable of visual pattern recognition. In 1960, Widrow-Hoff described a more robust learning procedure called ADALINE (ADaptive LINear Element). This is an acronym for adaptive linear element that also commonly known as the Least Mean Square (LMS) learning method (Widrow, 1960). In 1969, Minsky and Papert published a book entitled Perceptron (Minsky and Papert 1969), wherein the authors showed a number of major limitations of the networks. These limitations were widely believed to apply to neural networks in general and not just to the specific type of "*Perceptron*" currently known as one-layer perceptrons. Because of this publication, neural network research thought to be a "dead end", and funding stopped almost completely.

In spite of the setback caused by Minsky and Papert book, few researchers continued their efforts in the field of artificial neural networks. Fukushima (1975) developed the cognitron; Grossberg (1987) pioneered the basics for his Adaptive Resonance Theory (ART), and the most notable Teuvo Kohonen (1988), who investigated nets that used topological features. It was nearly a decade later before it was widely recognized that the valid criticism of perceptrons by Minsky and Papert did not apply to more complicated neural network models.

Significant breakthroughs came in the 1980s, when Hopfield (1982), assembled many of the ideas from previous researches and established neural network model the so-called "Hopfield network", based on fully interconnected binary units. The unit has two states 0 or 1, and all units connected to each other with the exception that no unit has any connection with itself. During the 1980s, Rumelhart, *et. al*.(1986), popularized the back propagation algorithm for training feed-forward networks which was also independently developed by Paker (1985). Later on, it was discovered that Werbos (1974) had developed this algorithm.

3.3 Definition of Artificial Neural Network

The current literature shows that there is no generally approved definition among the researchers of what neural network is. According to the DARPA Neural Network Study (Network, 1988),

> "*A neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes*."

Aleksander and Morton (1990) state that:

> "*Neural computing as, the study of adaptable nodes which, through a process of learning from task examples, store experiential knowledge and make it available for use*"

Zurada (Zurada, 1992) defined artificial neural networks as following:

> "*Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge*."

The following statement is from Haykin (Haykin, 1994) defining neural network as follows:

> "*A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects, knowledge is acquired by the network through a learning process, and interneuron*

connection strengths known as synaptic weights are used to store the knowledge."

Fausett (Fausett, 1994) defines an artificial neural network as:

"*An information processing system that has certain performance characteristics, such as adaptive learning, and parallel processing of information, in common with biological neural networks*."

From the above definitions, we can conclude that most of the researchers agreed on the following: the artificial neural network is a network that consists of many simple processing elements called neurons, each neuron possibly having small amount of local memory. The neurons have high degree of interconnections and arranged for parallel computations. The links between units have weights (scalar values) associated with it, which could be modified during training.

3.4 Artificial Neural Network Fundamentals

Artificial neural networks are a large number of parallel and distributed computational structures composed of simple processing units called "*neurons*" interconnected via unidirectional signal channels called weights (Figure 3.1). The network consists of three main elements. First, the connection links that represent weights (w_{ij}), the second element is an adder that sums up the weighted input ($w_{ij}*x_j$), and the third is a transfer function that generates the output y_j. The output y_j may be used as an input to the neighboring neuron in the next layer.

The neuron is a simple mechanism that represents the function of a biological brain cell. In an artificial neural network, the neuron processes many functions such as input signal assessment, addition of signals and evaluation of threshold values to compute the output value. In addition, because of the parallel structure of the system a number of neurons can perform their computation concurrently.

Every neuron could accept many input signals simultaneously, however thee neuron would produce only one output value, which relies mainly on the input signals, weights and threshold for that particular neuron. Several network models have an additional input called bias. Elkamel (1998) defined the bias, as a neuron that connected to all neurons in the hidden layers and the output layers in addition its function is to supply an invariant output.

Neurons are typically categorized into three types based on their inputs and outputs: input neurons, output neurons, and hidden neurons. The input neurons are those that receive input signals from outside the system, whereas the output neurons are those that fire the signal out of the system. Hidden neurons have their inputs and outputs within the system; as illustrated in Figure 3.1 in reference to a three-layer perceptron.

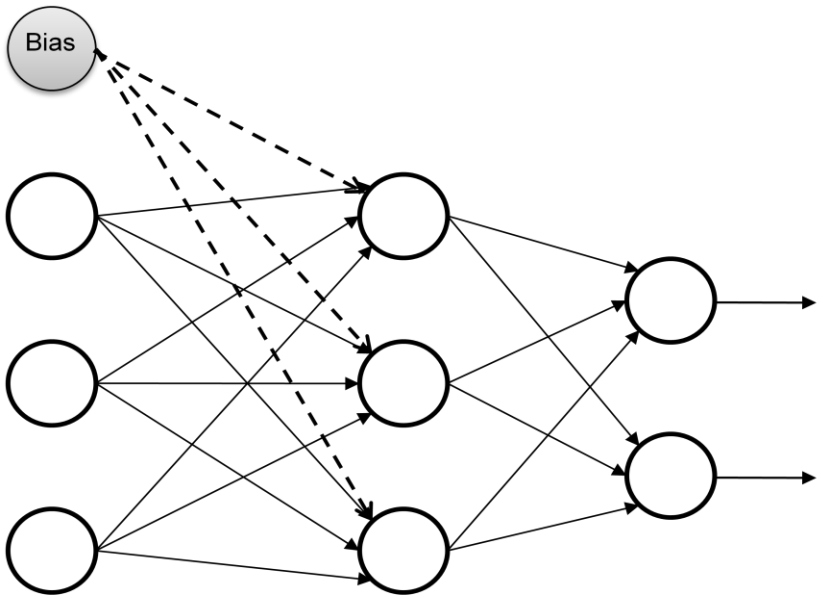

Figure 3.1: Artificial Neural Network

The transfer function is the process of converting the input signals to output signals for each neuron. The first step is to create the net input value for a neuron. Frequently, some inputs may be more significant than others, so there is a corresponding weight associated with each input introduced to a neuron. These weights represent the strength of the interconnections between neurons, and in general, are symbolized in terms of vectors such as $w_j = (w_{j1}, w_{j2}, \ldots, w_{jn})$. Once a neuron receives all the input signals, it determines the total input received from its input paths according to their weights. The most frequently applied technique is utilizing the summation function:

$$net_i = \sum_{i=1}^{n} w_{ij} x_i + w_{bj} \qquad (3.1)$$

where net_i is the net weighed input accepted by the $neuron_i$, w_{ij} is the weight value in the interconnection from $neuron_j$ to $neuron_i$ and x_i is the input signal from $neuron_j$, and finally w_{bj} represents the weight value from the bias.

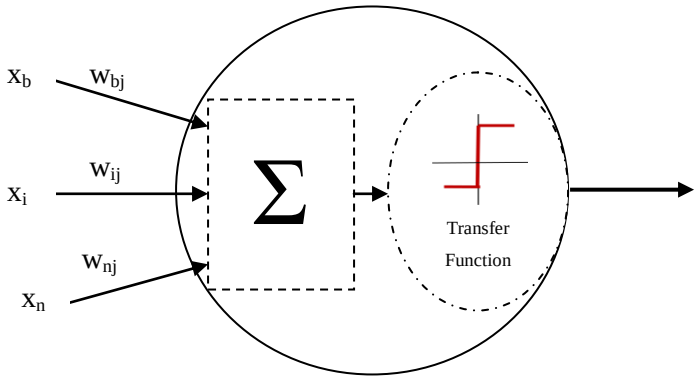

Figure 3.2 : The basic model of a neuron

Generating the activation value for each neuron is the second step of the process and it employs transfer functions. The transfer function can be a simple linear function or a more complex function such as a sigmoid function. Other functions used are threshold logic, step function, or the hyperbolic tangent function. The transfer function is continuous and non-linear. The most frequently used transfer function is the sigmoid function shown in Figure 3.2, which is bordered within a specific range, such as (0,1) and has a continuous first derivative. This function is expressed mathematically as:

$$f_{(x)} = \frac{1}{1+e^{-x}}$$

The transfer function creates the activation value for each neuron into its output value. The output of the neuron is determined by comparing it with the threshold value. The neuron will produce the output signal if the net weighed input is greater than its threshold value, or else the neuron will not produce any signal.

$$\text{Output}_i = \begin{cases} f(\text{sum}_i) & \text{if} \quad \text{sum}_i > \theta_i \\ 0 & \text{otherwise} \end{cases}$$

where θ_i is the threshold value corresponding to each neuron .

3.5 Artificial Neural Network Architectures

Artificial neural network architectures can be classified into three categories: feedback, feed forward, and self-organizing neural networks.

Feed forward is the second category of neural networks. In this architecture the signals go in only one direction; hence there are no loops in the network as illustrated in Figure 3.1. Linear feed forward is the earliest neural network model. Anderson (1972) and Kohonen (1972) independently and simultaneously each published an article in 1972 introducing the same model.

Feedback network is the network where the output signals of neurons directly feed back into neurons in the previous or the same layers, as illustrated in Figures: 3.3 and 3.4. This is called feedback network and also denoted as recurrent network, because the way neurons interconnect with each other, they could send signals to the previous layer as an

input signal. Williams and Zipser (1989) developed a fully recurrent neural network. The fully recurrent network is illustrated in Figure 3.3 (Williams and Zipser Network) and consists of many fully connected neurons with a sigmoidal activation function.

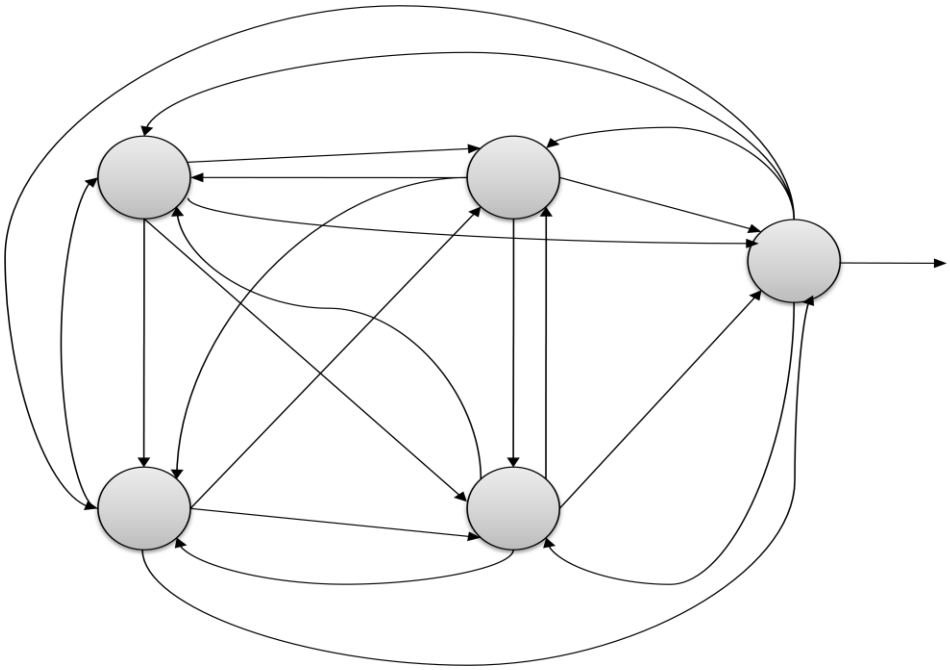

Figure 3.3: Fully Recurrent Network (Williams and Zipser Network)

Elman (1990) and Jordan (1986) proposed a partially feedback networks based on feed forward multilayer perceptrons and containing an additional so-called context layer as shown in Figure 3.4.

The neurons in these layers serve as internal states of the model. The main characteristic of the partial feedback network is that the hidden or the output layer are respectively fed to the context units.

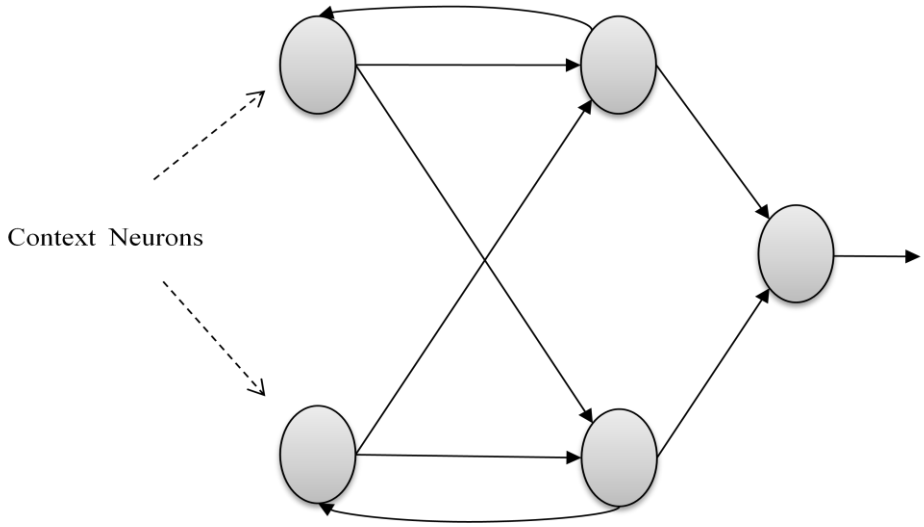

Figure 3.4: Partly Recurrent network (Elman Network)

Currently the most common utilized networks are nonlinear feed forward models. The current feed forward network architecture performs better than the current feedback architecture for a number of reasons. First, the capacity of feedback network is unimpressive. Secondly, feed forward networks are faster since they need to make only a single pass through the system to find the solution; on the other hand, feedback networks, need to loop repeatedly until a solution is found.

The third category in artificial neural network architecture is self-organized neural networks. Kohonen in 1982 introduced self-organizing neural networks, which are also known as Kohonen networks.

A Kohonen network has two layers: an input layer and a Kohonen layer. The input layer consists of neurons that distribute the input pattern values to each neuron in the Kohonen layer via links as shown in Figure 3.5. The Kohonen layer is a collection of neurons arranged in a tabular format. Once the network is exposed to samples, it begins to generate self-organizing clusters. Eventually, the Kohonen network will organize clusters and could be used for further operations where no output or dependent variable is known. Each link between the neurons maintains a specific value weight that is modified during the training process.

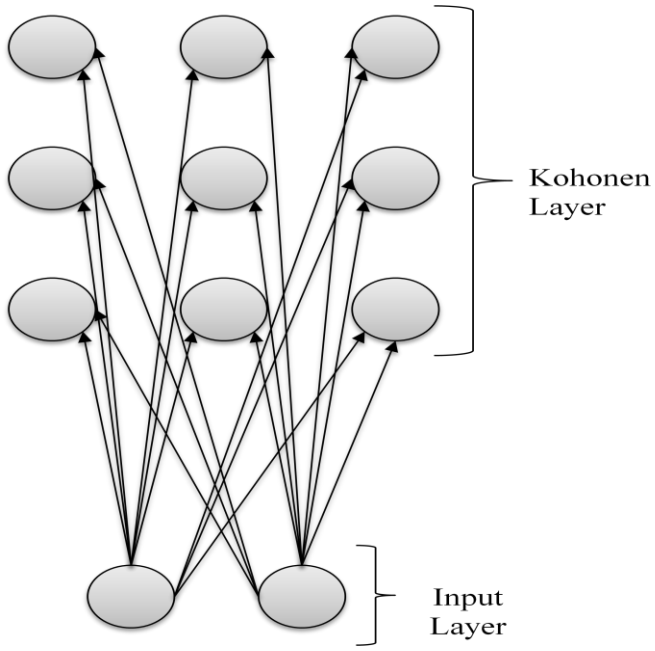

Figure 3.5: Kohonen Network

3.6 Training of Artificial Neural Network

The artificial neural network has the ability to learn from data during the training process and demonstrates an intelligent capability. A neural network consists of individual neurons that are interconnected with each other; as computation is progressing neurons signal to each other these signals carried through the connections. The connections have uneven magnitude; each connection is assigned a unique connection weight. If there is no connection between two neurons, then their connection weight is zero. These weights are what establish the output of the neural network; accordingly, it can be said that the weights form the memory of the neural network. Once network architecture structured for a particular application, the network is ready to be trained. Most of the literature agrees on two main approaches for artificial neural network training: supervised and unsupervised. There are two terminologies used in the literature for describing the process "*training*" and "*learning*"; therefore, both terms are used in this section. A trained network is expected to remember and generalize well. However, the ability to generalize can be determined by evaluating the performance of the network classification on new data beyond the training data sets. Most of the networks currently utilize supervised training. Unsupervised training is employed to determine some initial characterization of inputs.

3.6.1 Supervised Training

In supervised training, the network is provided with an input data and the expected output. The resulted output of the network is compared with the expected output and the

difference between these two values is considered as an error. The network then computes and propagates the error back through the system and adjusts weights accordingly. The weights are adjusted constantly by updating them at each step of the learning process. This process continues over several iterations until a desired output is computed with minimal error. The training is terminated when the performance goal is achieved.

Reinforcement learning is a less commonly used form of neural network learning and is occasionally considered as a special case of supervised learning (Pal and Srimani, 1996). It allows a network to learn the input-output mapping through trial and error, with the aim of maximizing a performance. The network will be told if it produced the desired result or not (true or false). If an action has been successful then the weights are altered to reinforce that behavior otherwise that action is discouraged in the modification of weights. The weights are updated based on these results and only the input vector is utilized for weight correction. Reinforcement learning lies between the supervised learning and the unsupervised learning.

3.6.2 Unsupervised Training

Unsupervised learning, also known as self-organization, takes place when the desired output data is not available and the network relies only upon local information during the learning process. A typical unsupervised network consists of an input layer and a competitive layer. Since the desired output is unknown, error information is not employed to improve the network behavior. Neurons on the competitive layer compare

with each other via a simple competitive learning rule to represent a given input pattern. During the learning process, the network updates its parameters and this process is commonly known as self-organization. A self-organizing network also known as self-organizing map (SOM), or Kohonen network, is the most common algorithm used in unsupervised neural networks (Kohonen, 1982).

3.7 Strengths and Limitations of Artificial Neural Networks

As many other artificial intelligent methods, artificial neural networks have strengths and limitations. Some of the major strengths of the artificial neural network technique are as follows: the ability of learning to recognize patterns or to approximate functions by identifying weight values. An artificial neural network can generate an organization based on the input and output data received through training. Artificial neural networks function in parallel rather than in serial, therefore faulty units or connections result in graceful degradation rather than a sudden collapse (Kohonen, 1988b). The most important strength of an artificial neural network is the ability of a suitable configuration to extend its forecasting capability beyond the set of calibrated data.

Artificial neural networks have some limitations that should be taken into account when considering their applications. One such limitation is that artificial neural networks still require a sound understanding of scientific principles to interpret the data created in an effective and efficient manner; otherwise, the neural network becomes a black box. A second limitation is that there are no guidelines or procedure for selecting the number of hidden layer nodes, number of training iterations, and preprocessing of

data. A third limitation of artificial neural networks is that the quality and scatter of the data set used to train the network can affect the ability of the neural network model to generalize and accurately predict the response factors (Kesavan, 1996).

Justifications for using artificial neural networks are based on the following properties: the ability of the neural network to learn through a repetitive training process that enables the network to improve its performance (McClelland & Rumelhart, 1986). The network could generalize what it learned consequently enabling it to respond to unexpected situations (Denker, et al., 1987), and finally the artificial neural network architecture allows massive parallel processing simultaneously.

Chapter 4

Development of Artificial Neural Network Model for Viscosity Prediction

4.1 Introduction

The objective of this chapter is to develop an ANN model for viscosity prediction of liquid mixtures. The issues investigated in this chapter are viscosity data, data normalization, development of the network, training, validation, ANN predicted viscosity, performance evaluation, and the selection of an optimal ANN. The criteria for choosing the parameters and conditions are also addressed. The best ANN configuration is sought as explained in the sequel.

4.2 Research Procedure and Methodology

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 4.1. Details of the methodology are explained in order.

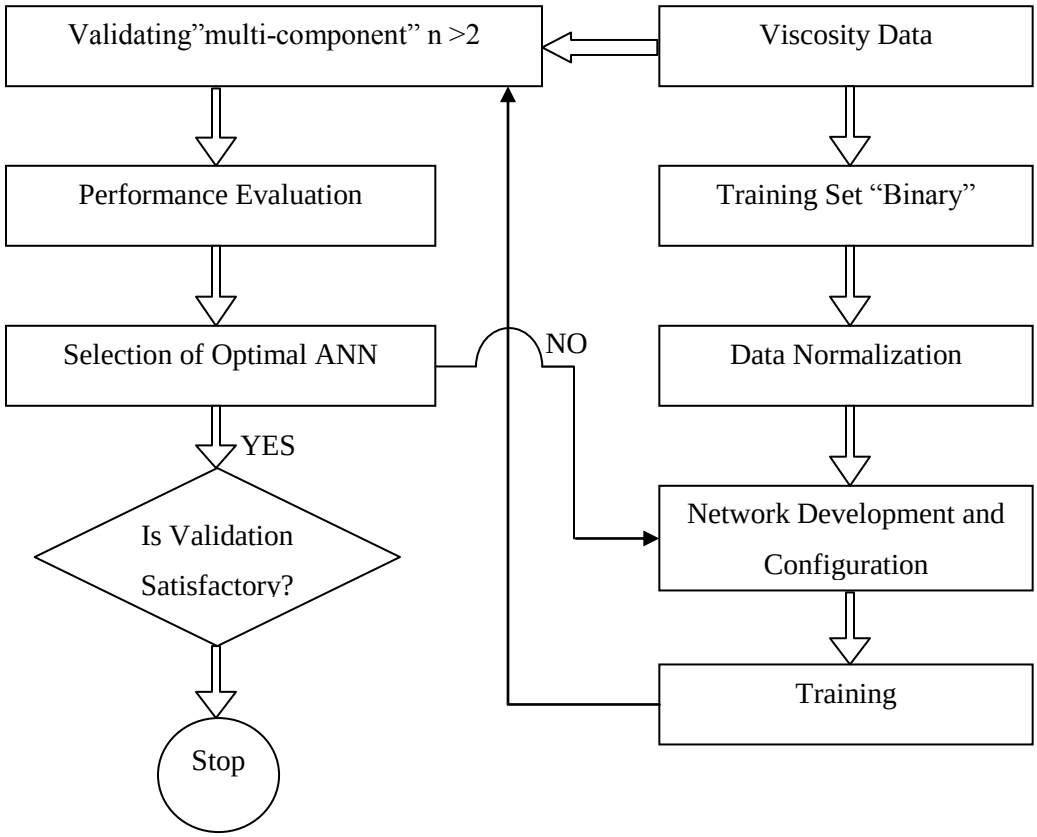

Figure 4.1: Methodology for developing an ANN Architecture

4.3 Why Artificial Neural Network

Neural networks take a different approach to problem solving than that of conventional techniques such as computers. Conventional techniques use an algorithmic approach, i.e. the computer follows a set of instructions to solve a problem. An algorithm can solve a problem only if the specific steps needed to be followed are specified in detail. The problem solving in such a case is therefore restricted to problems that we already understand and know how to solve. Neural networks, on the other hand, with their remarkable ability to derive meaning

from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. The ability of neural networks to learn by example, make them suitable for tasks that cannot be solved algorithmically or for values with unknown correlation. One of the distinct strengths of neural networks is their ability to generalize. The network is said to generalize well when it sensibly interpolates input patterns that are new to the network. Neural networks provide, in many cases, input-output mappings with good generalization capability. It can be said that neural networks behave as trainable, adaptive, and even self-organizing information systems (Schalkoff, 1997).

4.4 Type of Software Utilized

The programming needed for this research was done with the Matlab programming language. Matlab is a matrix based computer language developed by the Mathworks Company. Matlab stands for Matrix Laboratory and was originally written to provide easier user access to the powerful LINPACK and EISPACK mathematical libraries. It is interpreted, meaning there is no separate compile step before run. The user can type commands in a terminal window and have them executed immediately. The syntax is simple to learn yet the language is very powerful, chiefly due to the large amount of toolboxes, or user contributed code extensions. Matlab is used in many contexts, within algorithm development, data analysis, graphical visualization, simulation, engineering and scientific computation, and application development. Matlab is used extensively in education institutes and industry applications allowing for an iterative improvement cycle.

4.5 Selection of Data

Numerous researchers have measured viscosity for many different liquids. Asfour Group is one of the few groups who published viscosity data at different temperatures and they are the only group, to our knowledge, who reported experimental data for quinary systems. Viscosity data were collected at different temperatures and concentrations. The data sets which consist of Heptane, Octane, Toluene, Cyclohexane, and Ethylbenzene included the viscosity values for binary, ternary, quaternary, and quinary liquid mixture.The viscosity data are categorised as systems and subsystems as listed in Table 4.1

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.

The data set of the system consisting 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 4.1.

In addition, the McAllister predictive capability represented by the percent absolute average derivation (%AAD) is also given and will be used for comparison purposes with the ANN predictions. The %AAD is defined as:

$$\text{AAD} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\nu_i^{exp} - \nu_i^{pred}}{\nu_i^{exp}}\right| \tag{4.1}$$

where n is the number of data points.

Table 4.1 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	Al-Gherwi, 2005
Heptane-Cyclohexane	293.15-313.15	1.83	Al-Gherwi, 2005
Heptane-Toluene	293.15-313.15	1.46	Al-Gherwi, 2005
Heptane-Ethylbenzene	293.15-313.15	1.88	Al-Gherwi, 2005
Octane-Cyclohexane	293.15-313.15	1.71	Al-Gherwi, 2005
Octane-Toluene	293.15-313.15	2.31	Al-Gherwi, 2005
Octane-Ethylbenzene	293.15-313.15	2.33	Al-Gherwi, 2005
Cyclohexane-Toluene	293.15-313.15	3.09	Al-Gherwi, 2005
Cyclohexane-Ethylbenzene	293.15-313.15	2.96	Al-Gherwi, 2005
Toluene-Ethylbenzene	293.15-313.15	0.33	Al-Gherwi, 2005
Heptane-Cyclohexane-Ethylbenzene	293.15-313.15	4.89	Cai, 2004 and El Hadad, 2004
Heptane-Octane-Cyclohexane	293.15-313.15	2.49	Cai, 2004 and El Hadad, 2004
Heptane-Octane-Ethylbenzene	293.15-313.15	3.93	Cai, 2004 and El Hadad, 2004

Heptane-Octane-Toluene	293.15-313.15	3.37	Cai, 2004 and El Hadad, 2004
Heptane-Toluene-Cyclohexane	293.15-313.15	5.12	Cai, 2004 and El Hadad, 2004
Heptane-Toluene-Ethylbenzene	293.15-313.15	2.59	Cai, 2004 and El Hadad, 2004
Octane-Cyclohexane-Ethylbenzene	293.15-313.15	2.88	Cai, 2004 and El Hadad, 2004
Octane-Toluene-Cyclohexane	293.15-313.15	2.98	Cai, 2004 and El Hadad, 2004
Octane-Toluene-Ethylbenzene	293.15-313.15	3.4	Cai, 2004 and El Hadad, 2004
Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	3.66	Cai, 2004 and El Hadad, 2004
Heptane-Octane-Cyclohexane-Ethylbenzene	293.15-313.15	2.83	Cai, 2004 and El Hadad, 2004
Heptane-Octane-Toluene-Cyclohexane	293.15-313.15	3.35	Cai, 2004 and El Hadad, 2004
Heptane-Octane-Toluene-Ethylbenzene	293.15-313.15	2.15	Cai, 2004 and El Hadad, 2004
Heptane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	3.83	Cai, 2004 and El Hadad, 2004
Octane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	2.72	Cai, 2004 and El Hadad, 2004

4.6 Weight Initialization

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.

4.7 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, $x_{i,\ norm}$, is set at zero (0) and the maximum value, $x_{i,\ norm}$ is set at one (1). We define the normalized variable $x_{i,\ norm}$ by using the minimum and maximum values of the original variable, $x_{i,\ min}$ and $x_{i,\ max}$, respectively, i.e.

$$X_{i,norm} = \frac{(X_i - X_{i,min}) \times (L_{max} - L_{min})}{X_{i,max} - X_{i,min}} + L_{min} \quad (4.2)$$

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.

4.8 Post Processing

Typically the output of the neural network is a set of unitless 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 (4.3) will achieve this goal,

$$y_{i,renorm} = \frac{(y_i - y_{i,min}) \times (M_{max} - M_{min})}{y_{i,max} - y_{i,min}} + M_{min} \quad (4.3)$$

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

4.9 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 feedforward networks is the mean square error (MSE) that is the average squared error between the predicted outputs and the experimental outputs, i.e.

$$\mathrm{MSE} = \frac{1}{\mathrm{n}-1}\sum_{1}^{\mathrm{n}}\left(\nu_{\mathrm{predeicted}} - \nu_{experimental}\right)^{2} \tag{4.4}$$

4.9.1 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 feedforward 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. Multi Layer 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.

4.9.2 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 overfit 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 (Demuth, Beale, & Hagan, 2007). 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 4.2. The performance values (MSE) of the entire training dataset are plotted against the number of epochs as shown in Figure 4.2.

Table 4.2: Number of Epochs and Mean Square Error of the Neural Networks

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

From Table 4.2 and Figure 4.2 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.

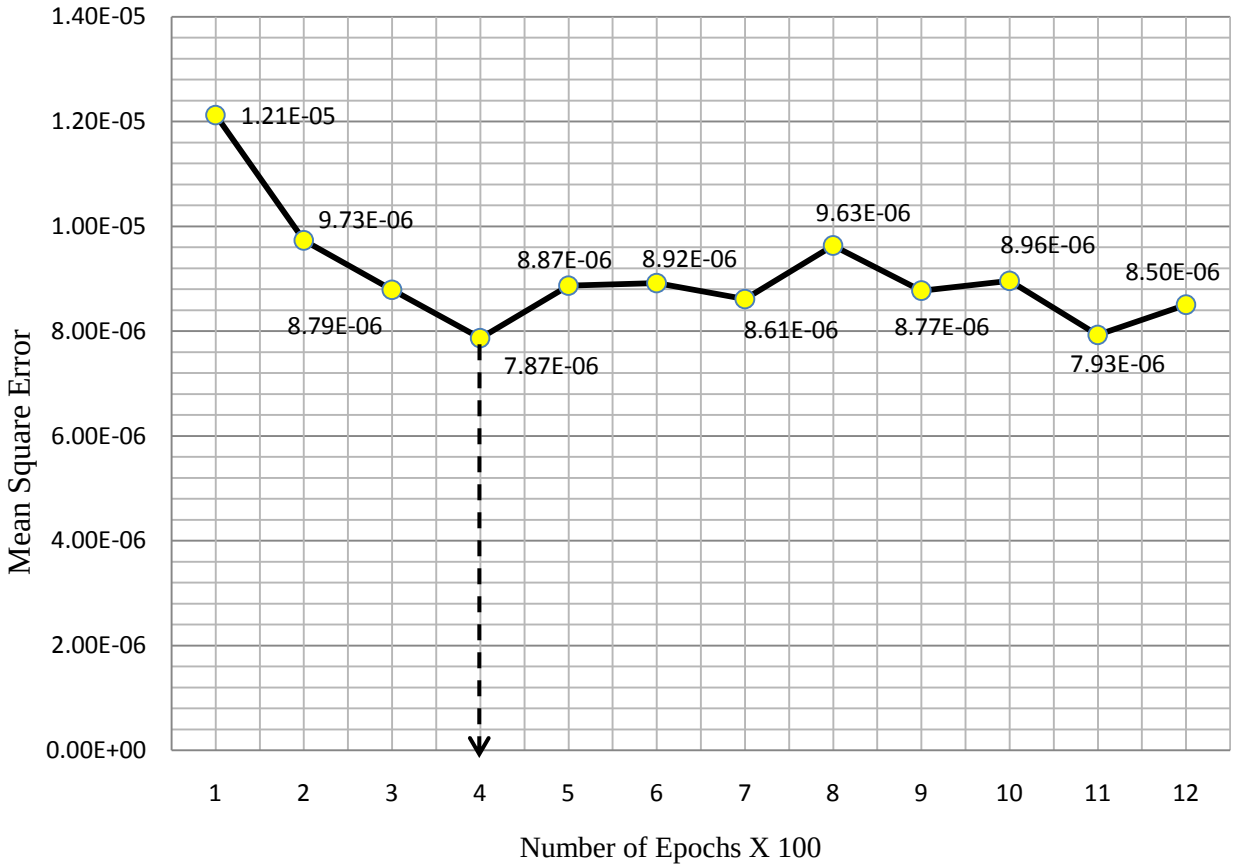

Figure 4.2: Mean Square Error versus Number of Epochs

4.9.3 Transfer Function

The transfer function fundamentals were addressed in Chapter 3. 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

(Anderson, 1995; Nelson & Illinworth, 1990). 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 (Hornik et al., 1989). The back-propagation algorithm used in this work requires continuous and differentiable transfer functions so it allows for weight update adjustments.

4.9.4 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 versus performance mean square error MSE was plotted as shown in Figure 4.3. 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 4.3 and plotted against the number of the hidden neurons as illustrated in Figure 4.4.

Table 4.3: Number of Neurons in the hidden layer versus 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

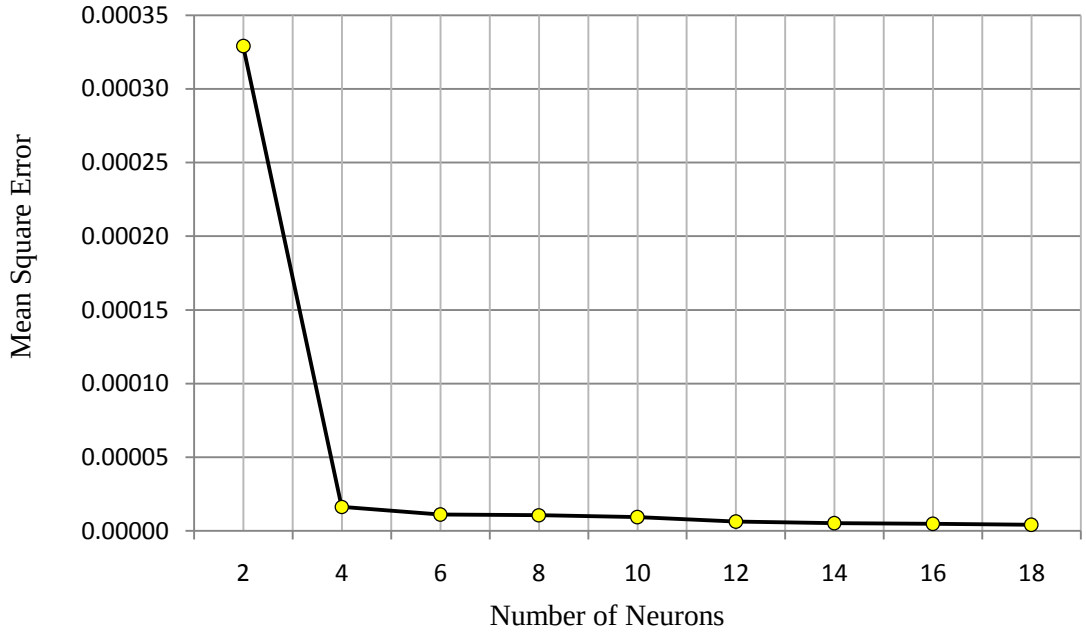

Figure 4.3: Number of Neurons in the hidden layer versus MSE

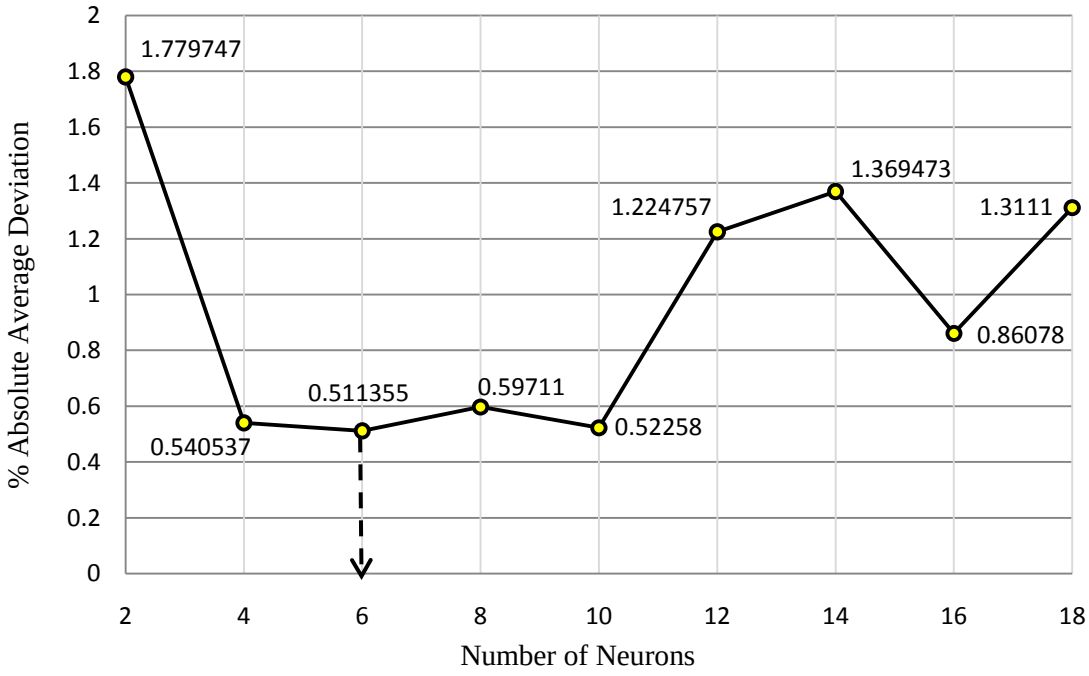

Figure 4.4: Number of Neurons in the hidden layer versus %AAD

Figure 4.4 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 overfitting.

4.10 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 (Rumelhart et al., 1986) 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.

Chapter 5
Results and Discussion

5.1 Results

This chapter will present the results obtained from the neural network models employed to predict the kinematic viscosity of a multi-component mixture and will also compare the results produced by the neural network model with those from the generalized McAllister model.

The generalized parameters obtained from the neural network models are plotted against the experimental values as shown in Figures 5.1 to 5.16. 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.

Tables 5.1 to 5.10 show 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 5.17. 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 5.20.

Tables 5.11 to 5.15 show 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%.

The quinary system used in this work consisted of only one system, and the minimum and the maximum values were therefore unavailable. The performance of the neural network for the quinary system had an AAD of 4.3611%. Because the initial values for the quinary system were relatively higher than those for the other systems, an additional approach was employed, as explained in detail in section 5.2.

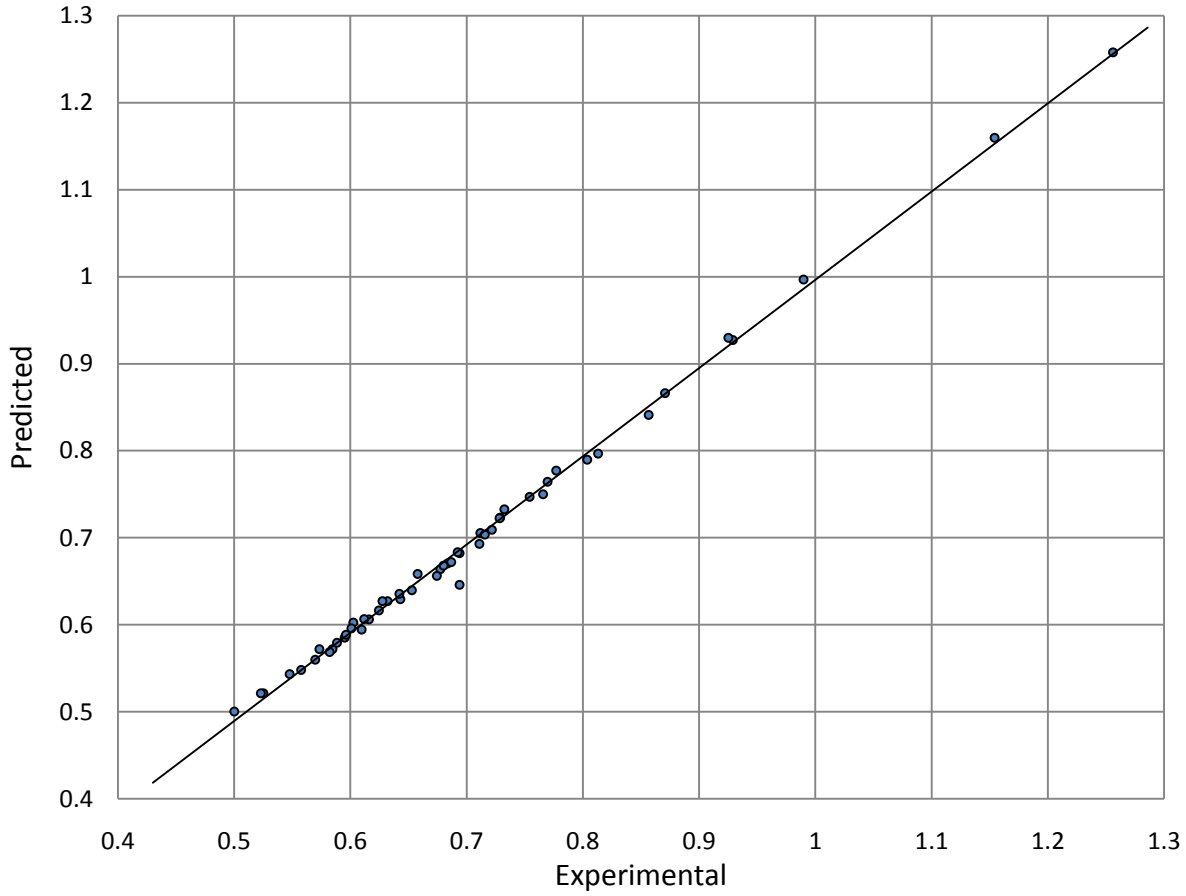

Figure 5.1: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.1: The predictive performance of neural network for "Heptane – Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.26010	6.83172	0.03552	0.99398

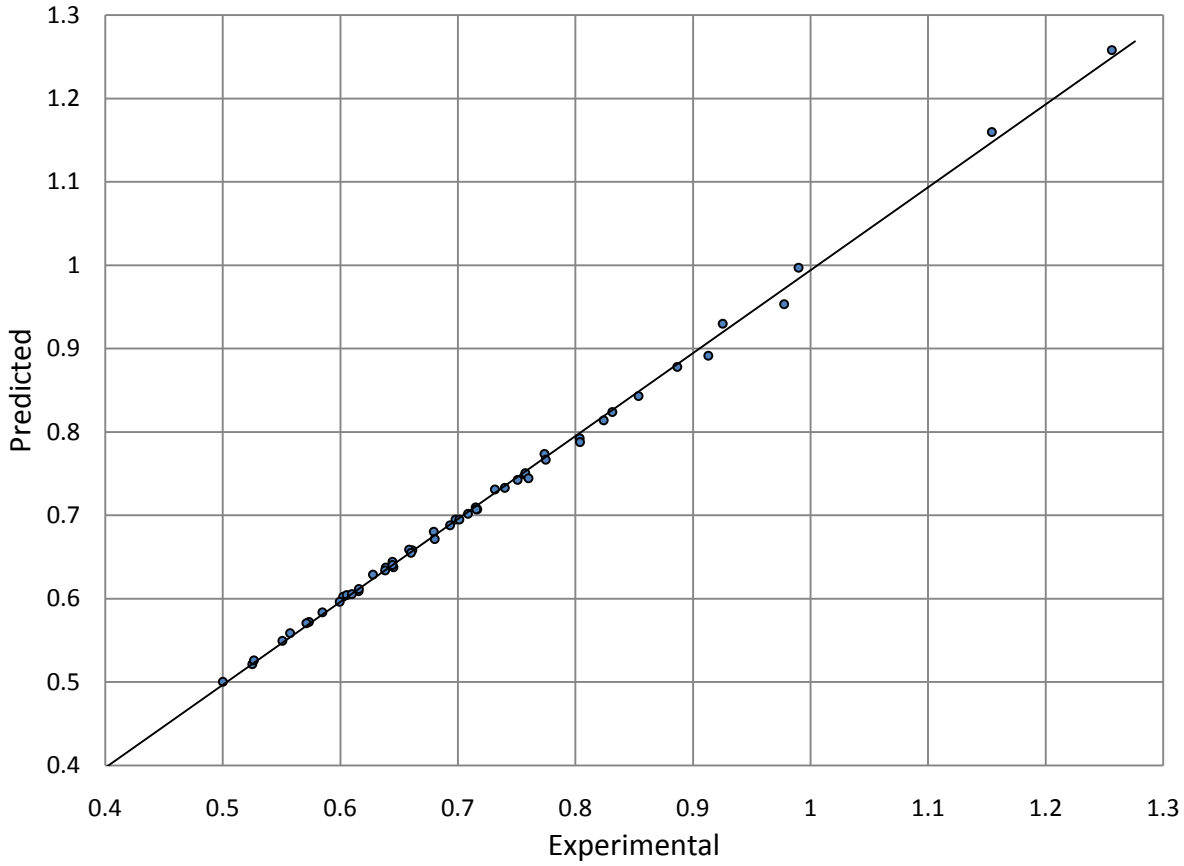

Figure 5.2: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Octane - Cyclohexane" for the entire temperature range 298 K - 313 K data set.

Table 5.2: The predictive performance of neural network for "Heptane - Octane - Cyclohexane" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.69556	2.4363	0.00122	0.99757

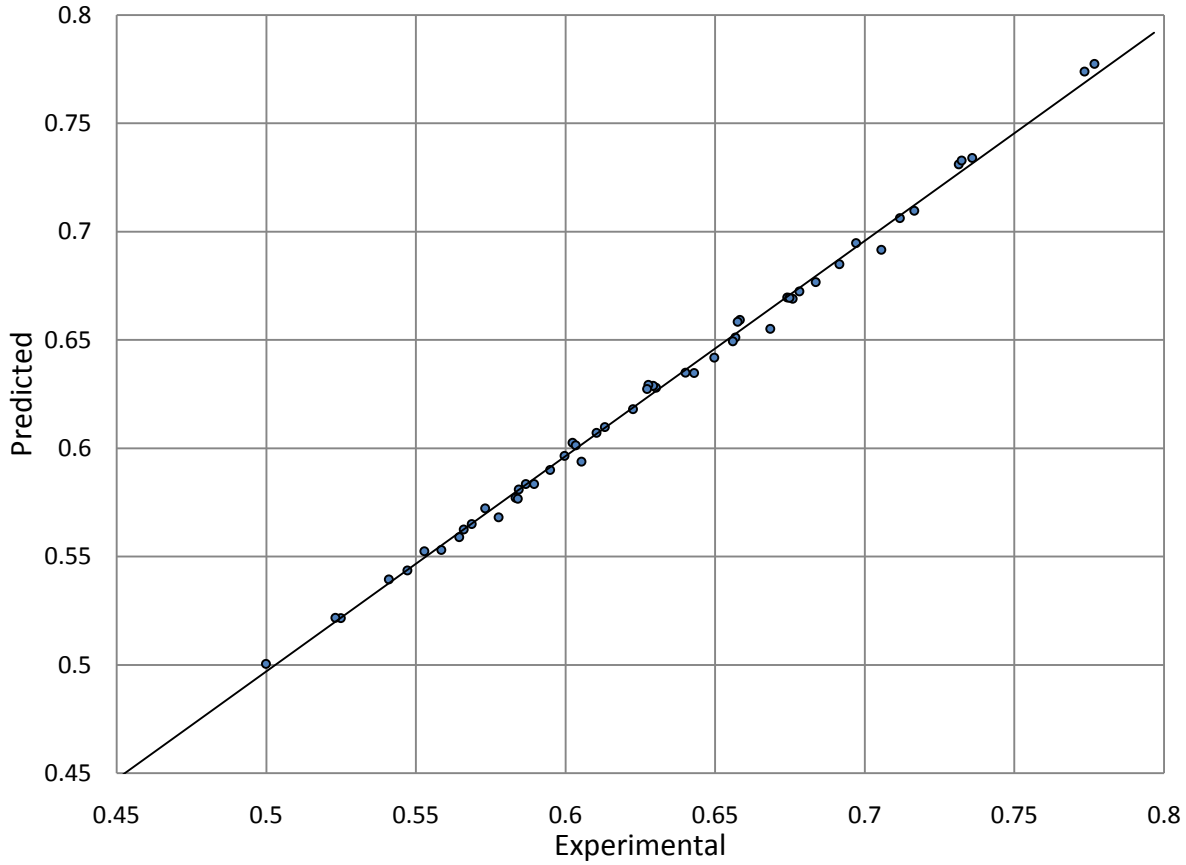

Figure 5.3 Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Octane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.3: The predictive performance of neural network for "Heptane - Octane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.65752	1.96910	0.02201	0.99338

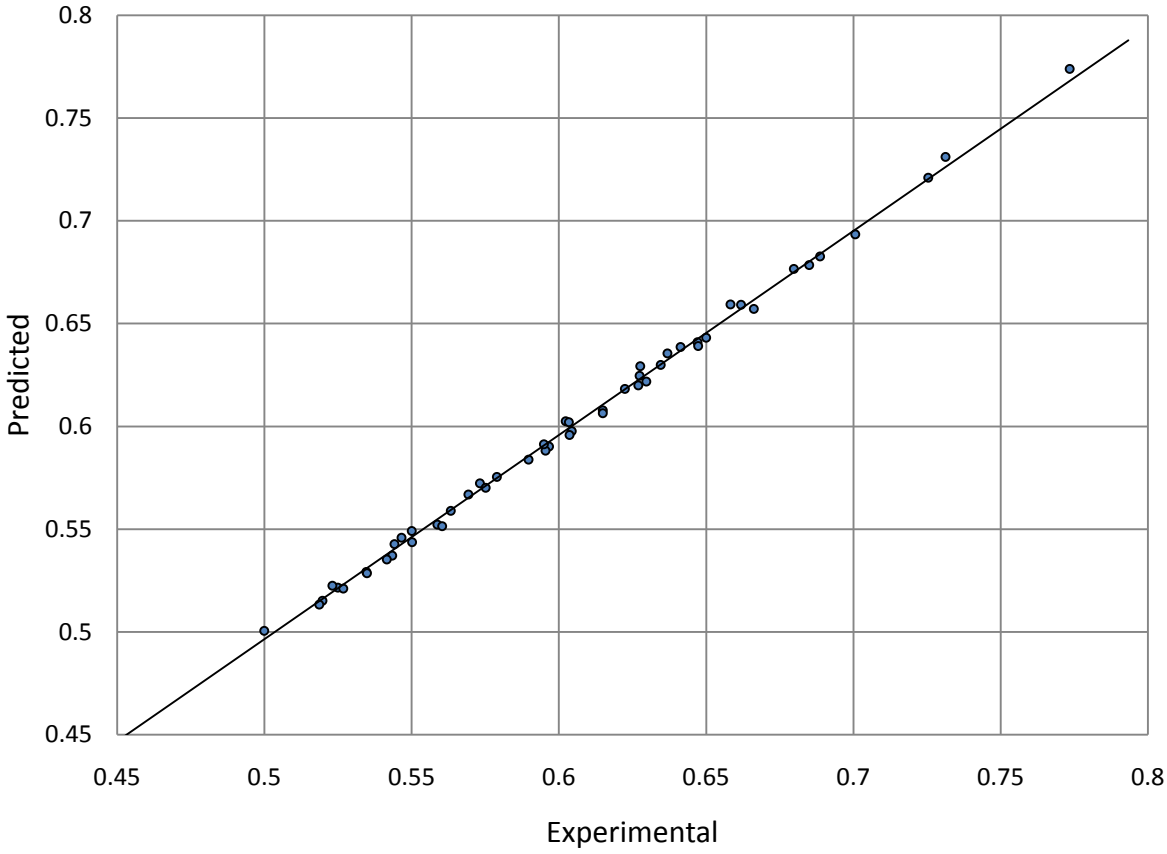

Figure 5.4: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Octane - Toluene" for the entire temperature range 298 K - 313 K data set.

Table 5.4: The predictive performance of neural network for "Heptane – Octane - Toluene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.74127	1.58725	0.02201	0.99289

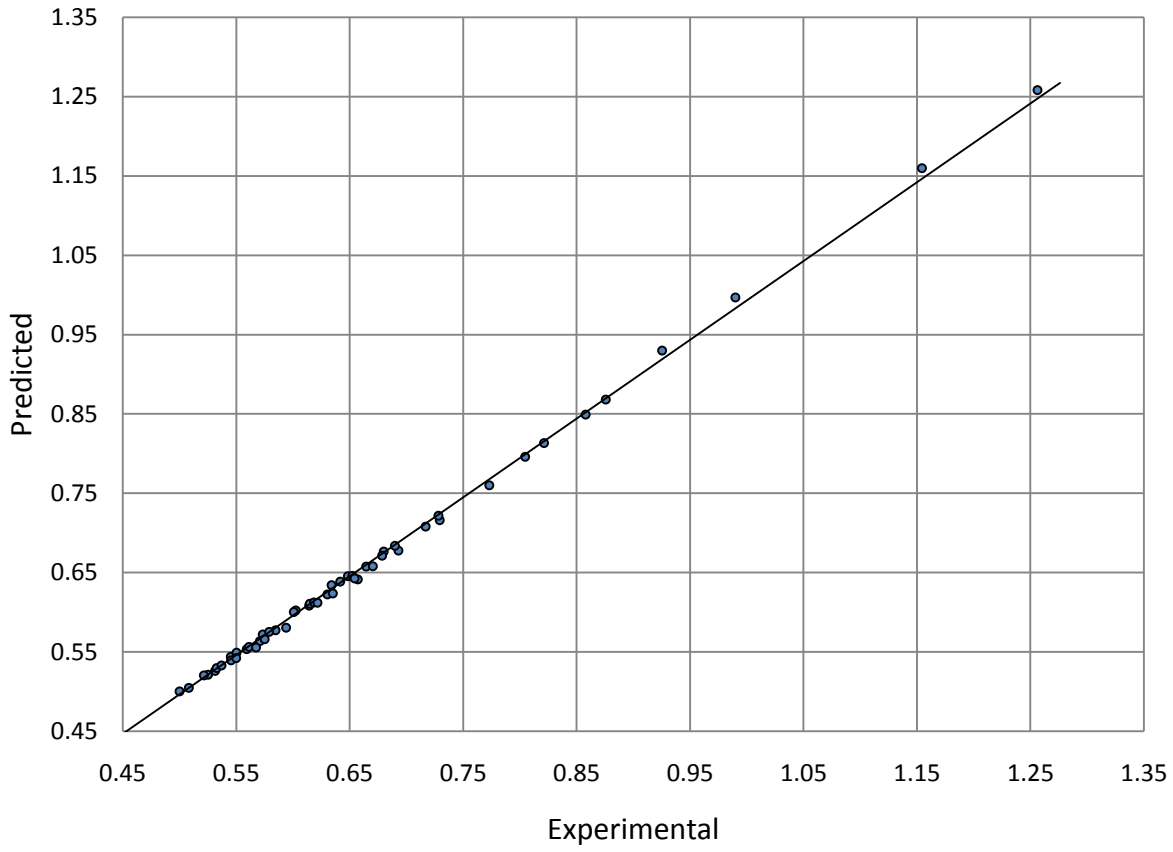

Figure 5.5: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Toluene - Cyclohexane" for the entire temperature range 298 K - 313 K data set.

Table 5.5: The predictive performance of neural network for "Heptane - Toluene - Cyclohexane" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.92114	2.34120	0.01658	0.99773

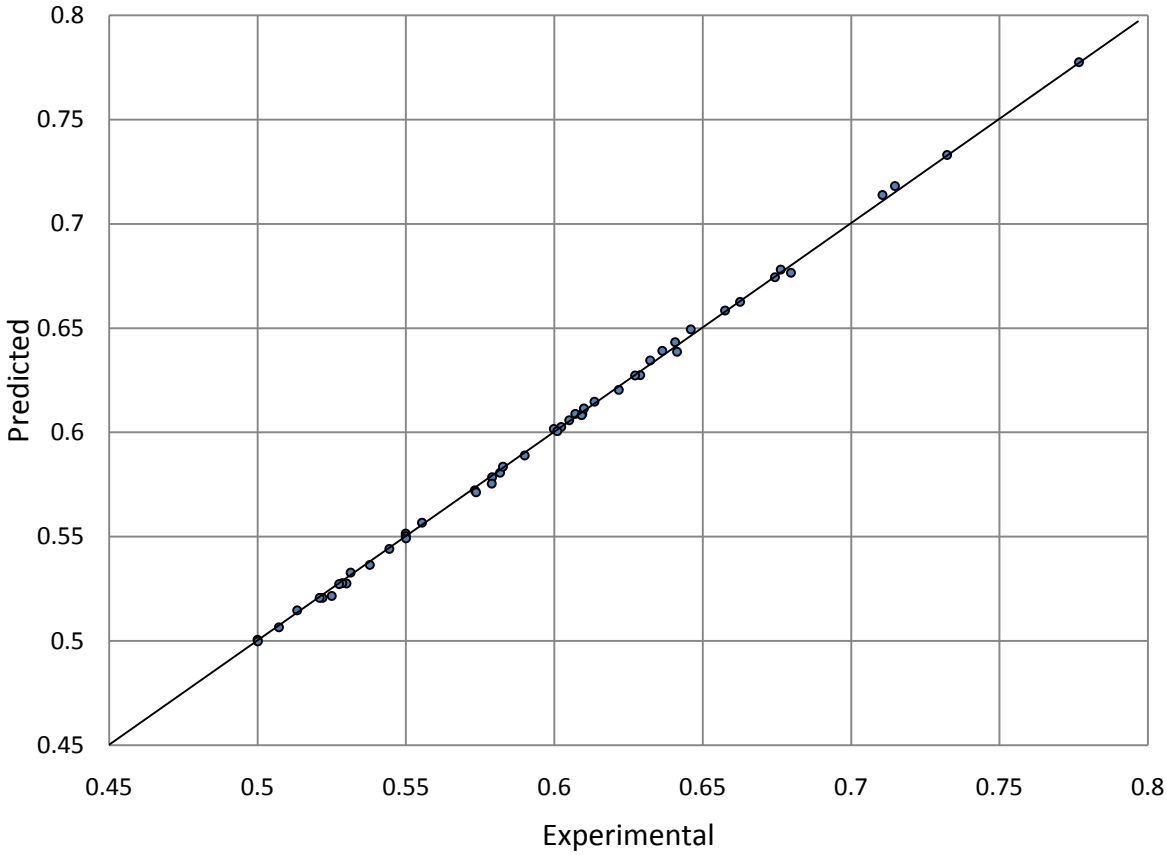

Figure 5.6: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Toluene - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.6: The predictive performance of neural network for "Heptane – Toluene - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.25856	1.47412	0.01483	0.99928

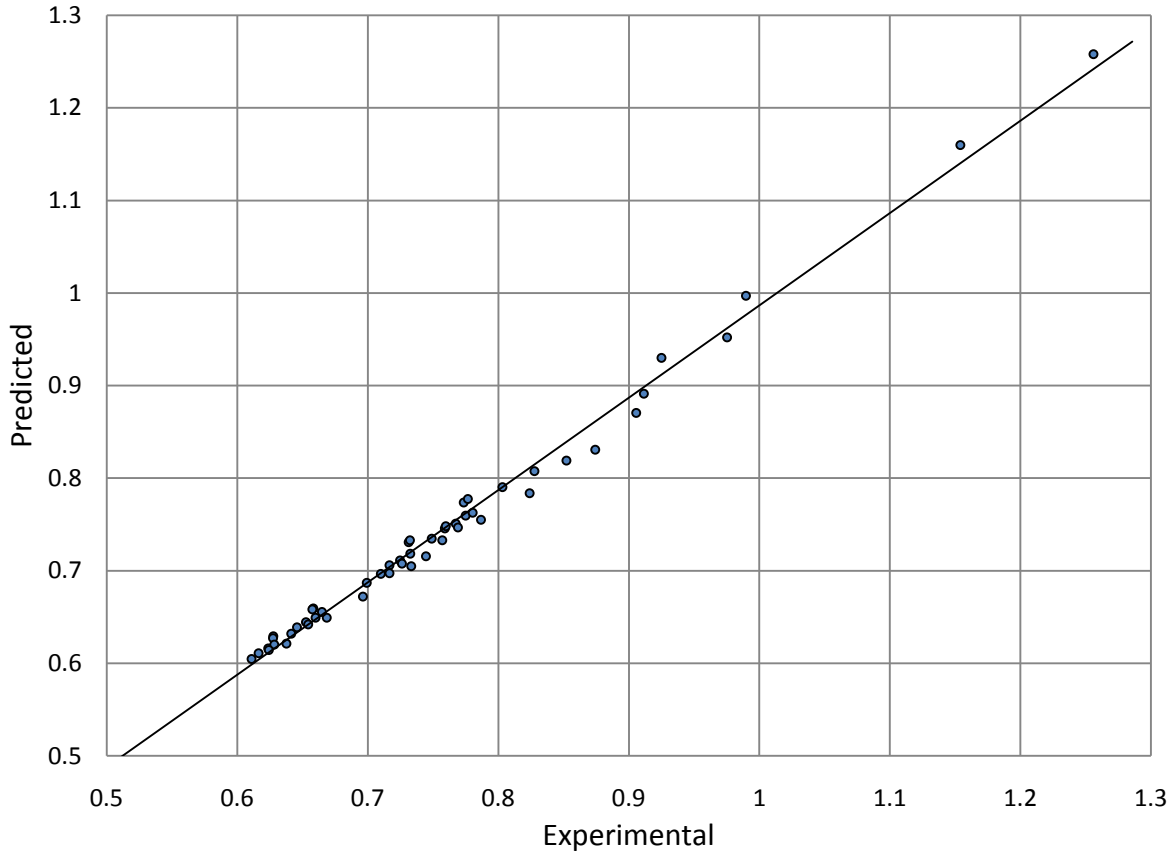

Figure 5.7: Validation plot of experimental versus predicted values of kinematic viscosity with "Octane - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.7: The predictive performance of neural network for "Octane - Cyclohexane-Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.82863	4.98058	0.02201	0.98175

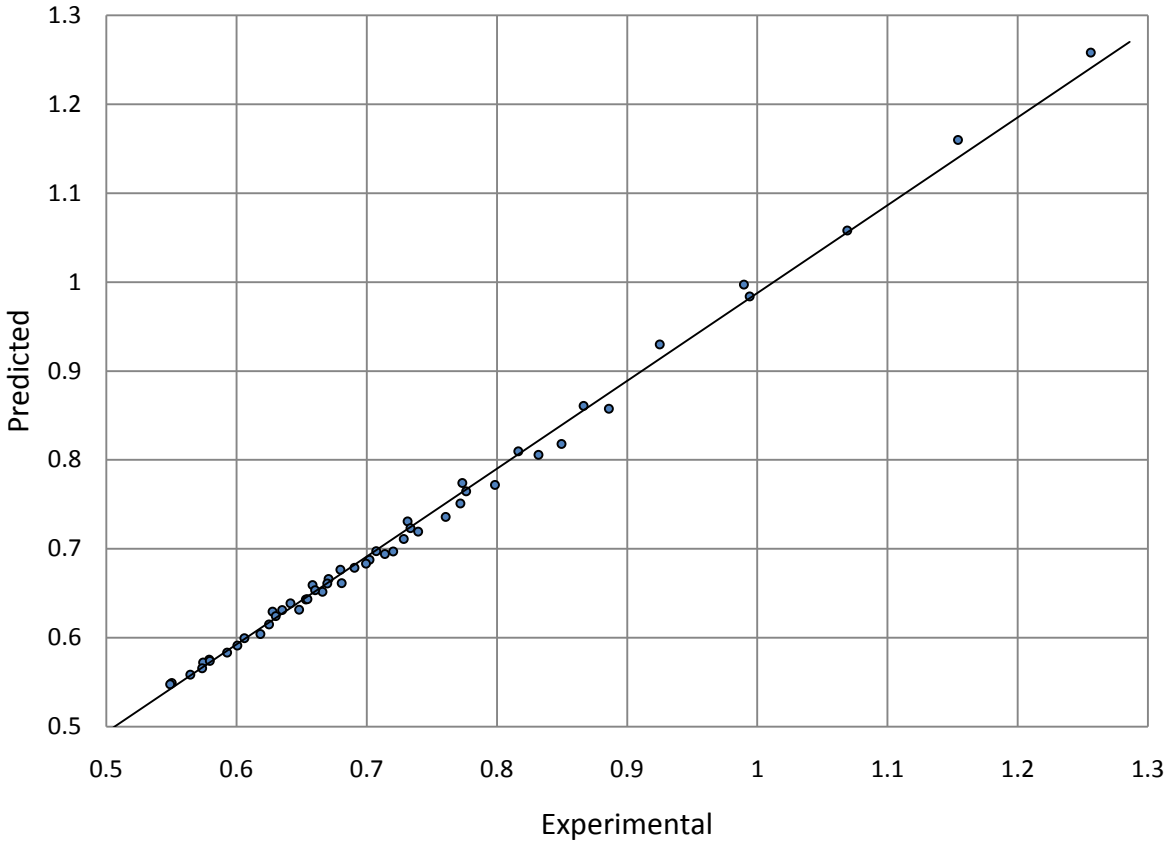

Figure 5.8: Validation plot of experimental versus predicted values of kinematic viscosity with "Octane - Toluene - Cyclohexane" for the entire temperature range 298 K -313 K data set.

Table 5.8: The predictive performance of neural network for "Octane – Toluene - Cyclohexane" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.43718	3.70471	0.02201	0.99237

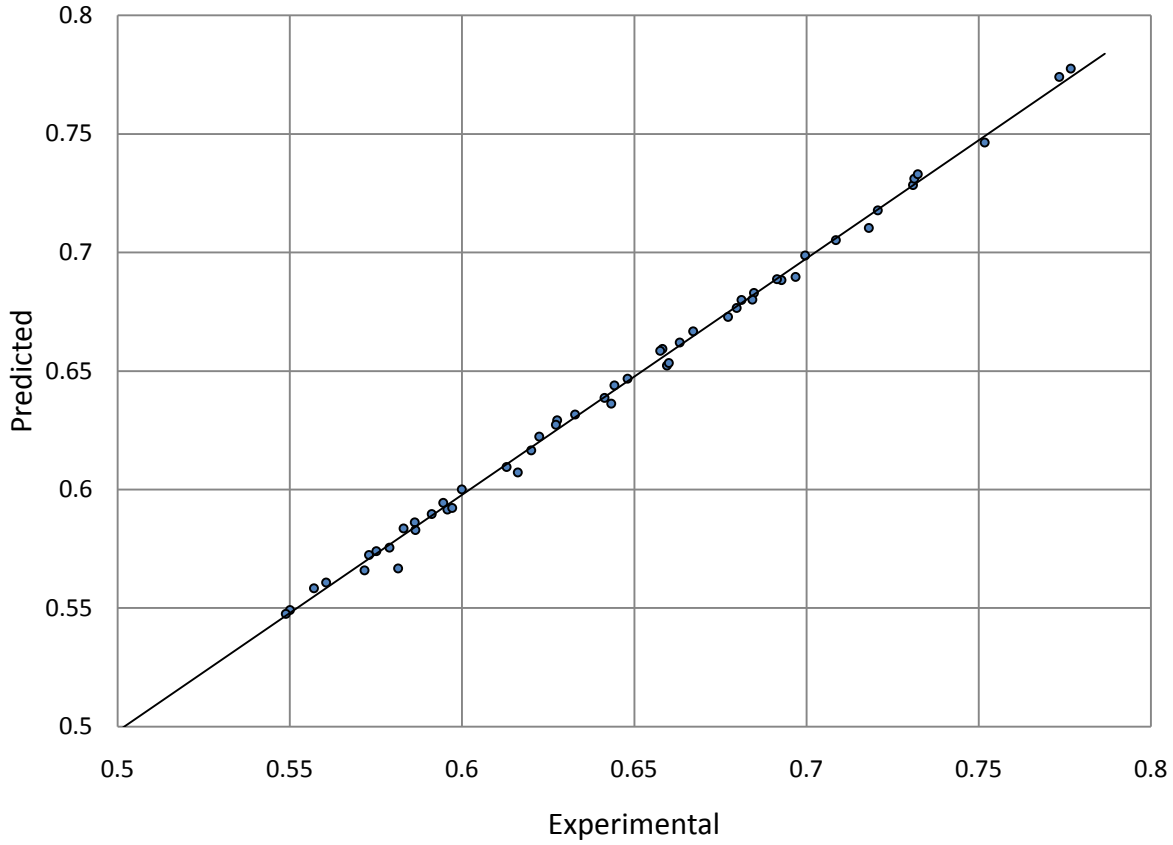

Figure 5.9: Validation plot of experimental versus predicted values of kinematic viscosity with "Octane - Toluene - Ethylbenzene" for the entire temperature range 298 K -313 K data set.

Table 5.9: The predictive performance of neural network for "Octane – Toluene - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.43444	2.52300	0.00770	0.99551

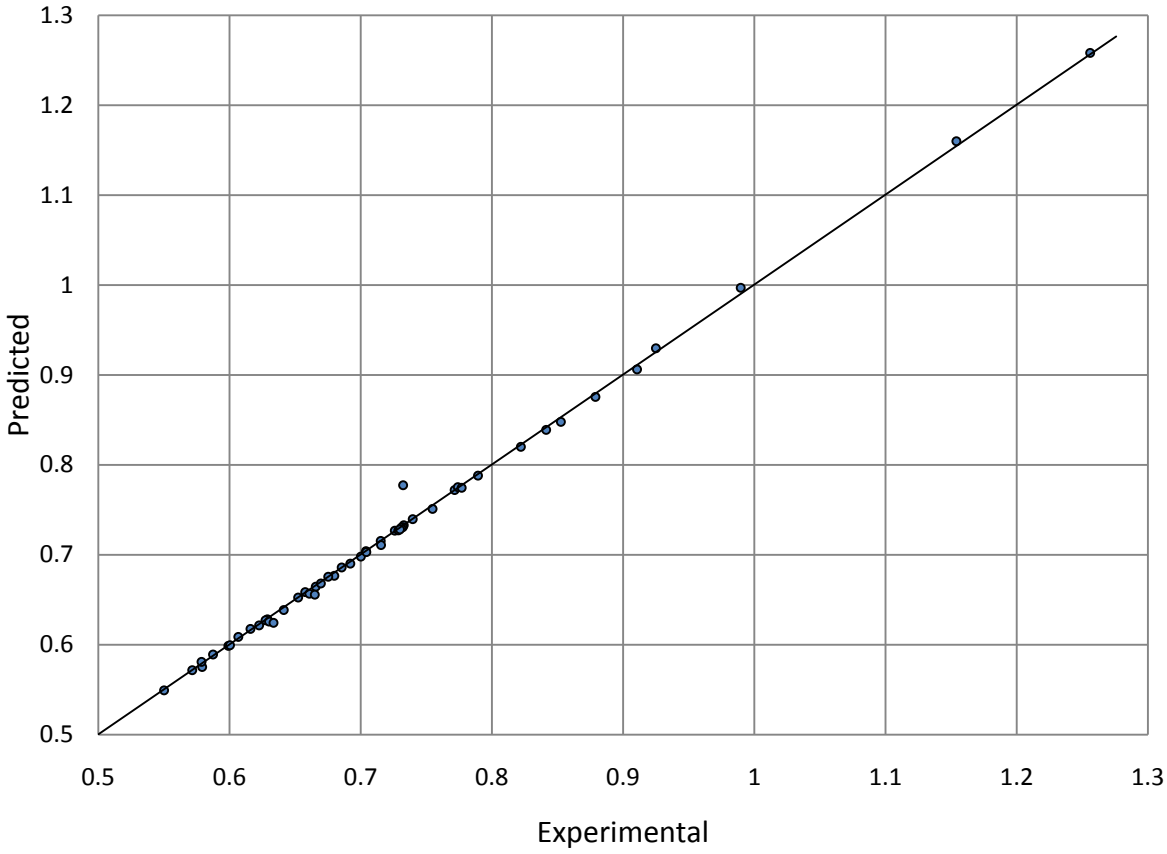

Figure 5.10: Validation plot of experimental versus predicted values of kinematic viscosity with "Toluene - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K -313 K data set.

Table 5.10: The predictive performance of neural network for "Toluene – Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.41207	6.16943	0.00088	0.99734

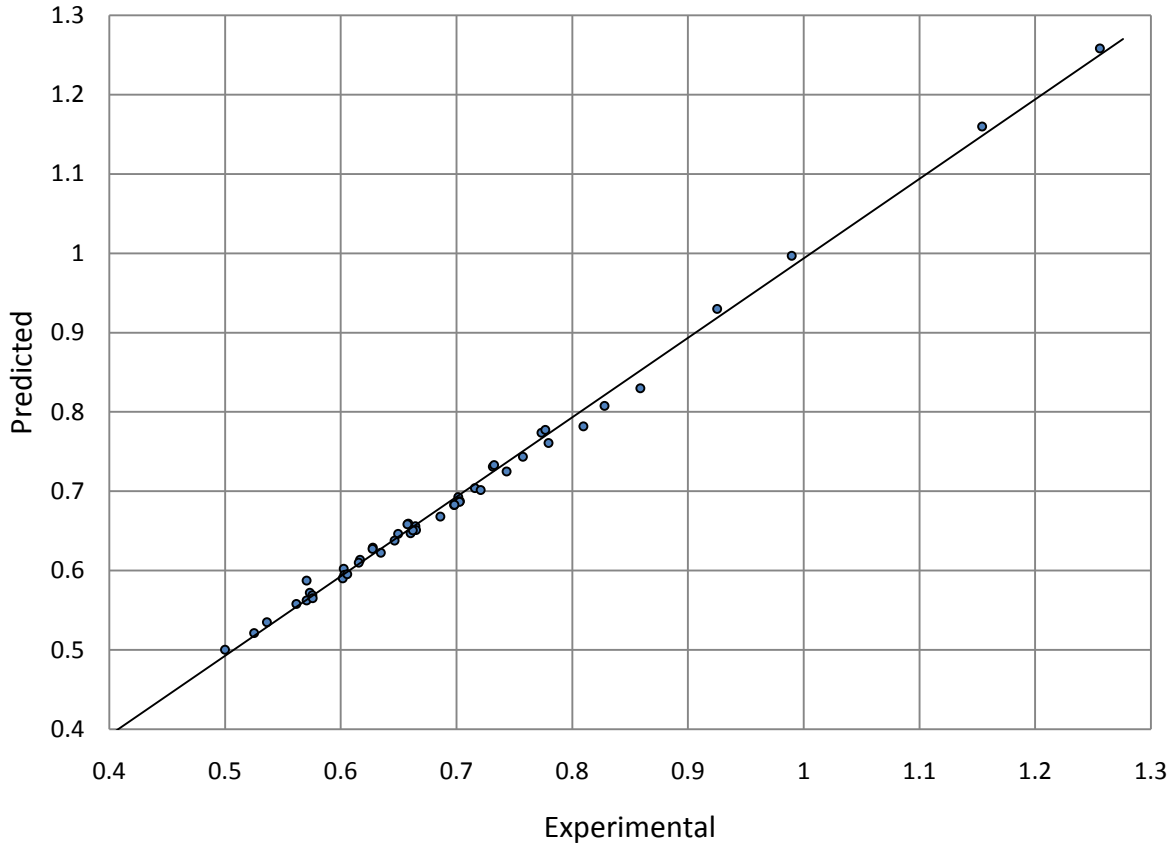

Figure 5.11: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane- Octane - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.11: The predictive performance of neural network for "Heptane – Octane – Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.31412	3.38938	0.02201	0.99366

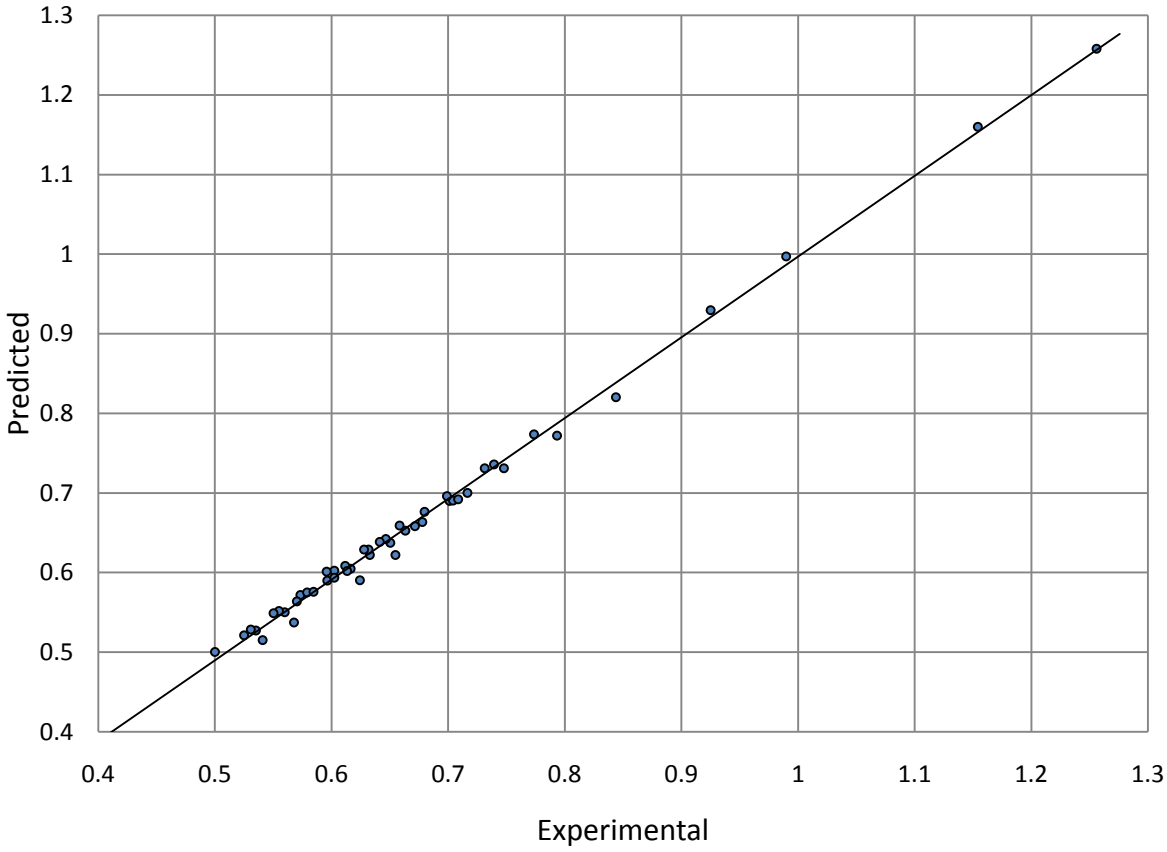

Figure 5.12: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane – Octane – Toluene - Cyclohexane" for the entire temperature range 298 K -313 K data set.

Table 5.12: The predictive performance of neural network for "Heptane – Octane – Toluene - Cyclohexane" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.40294	5.36505	0.02201	0.99301

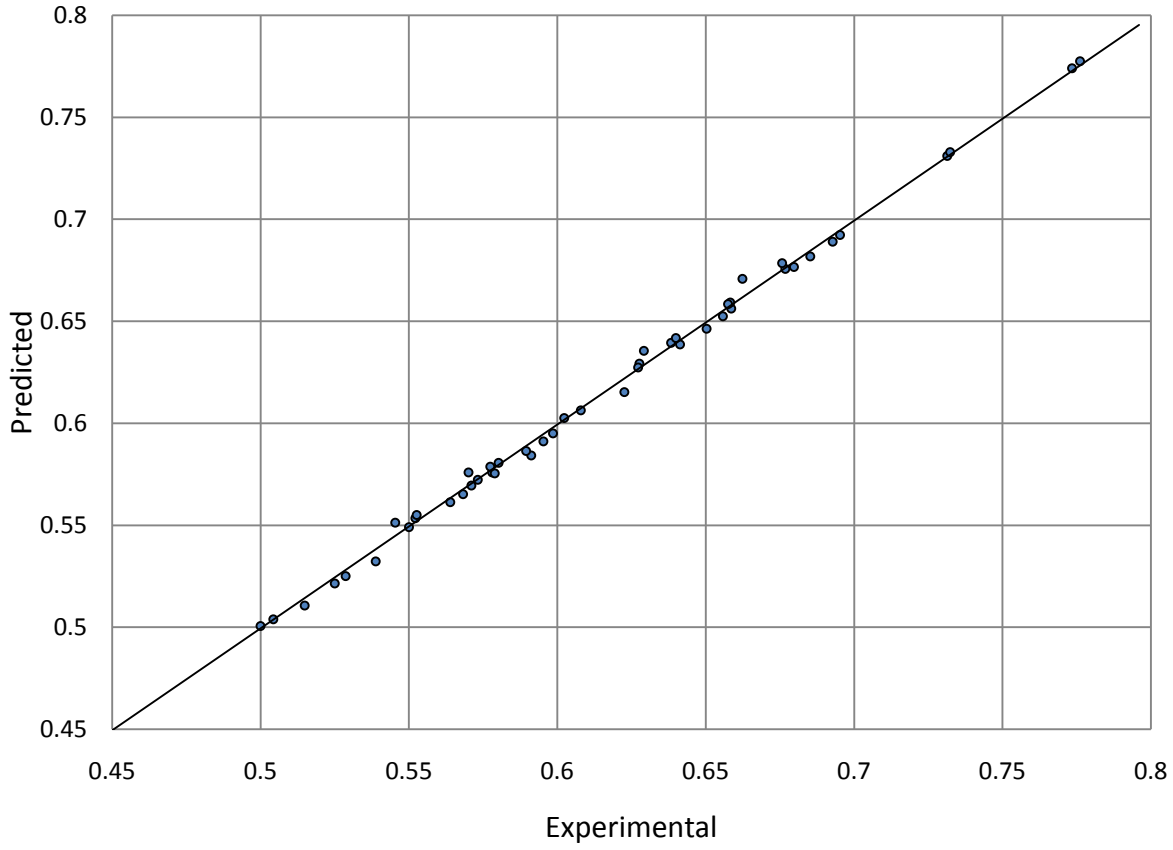

Figure 5.13: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Octane – Toluene - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.13: The predictive performance of neural network for "Heptane – Octane -Toluene - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.49160	1.38247	0.02201	0.99739

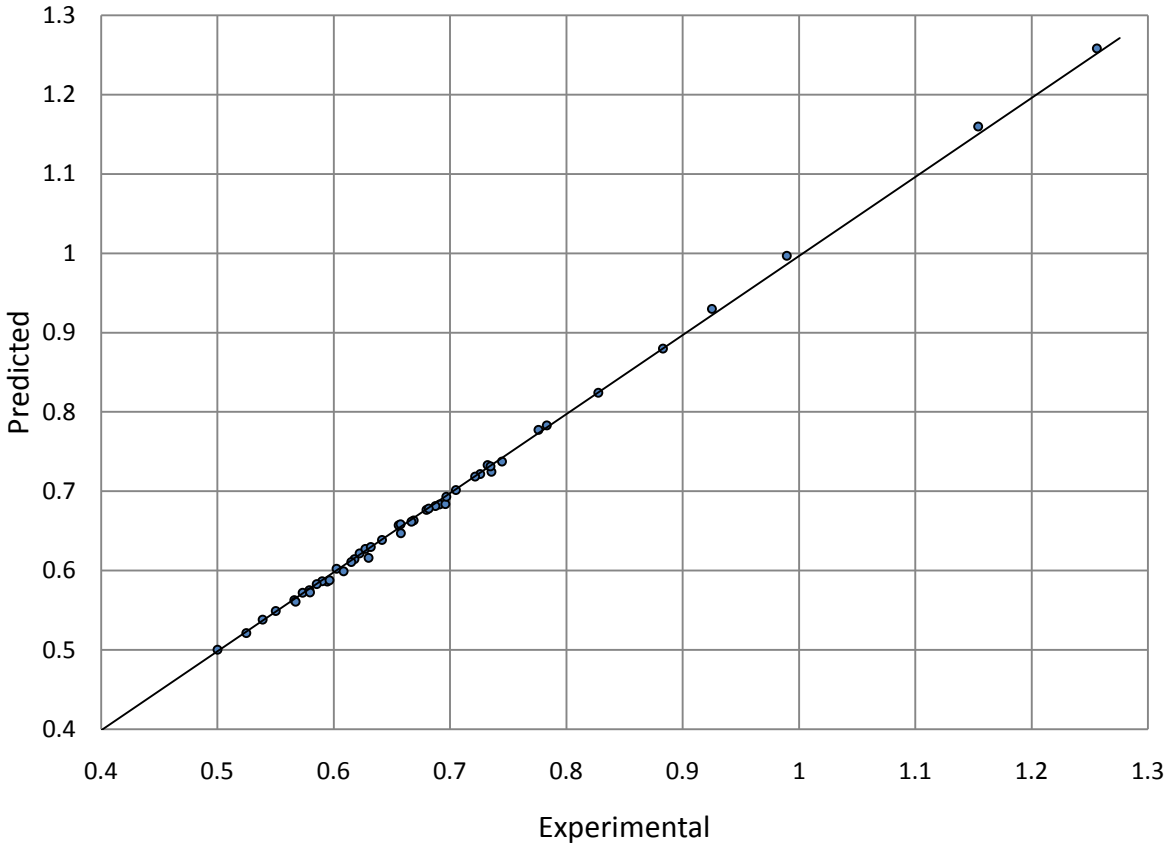

Figure 5.14: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane – Toluene – Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.14: The predictive performance of neural network for "Heptane - Toluene - Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.65777	2.19162	0.01532	0.99868

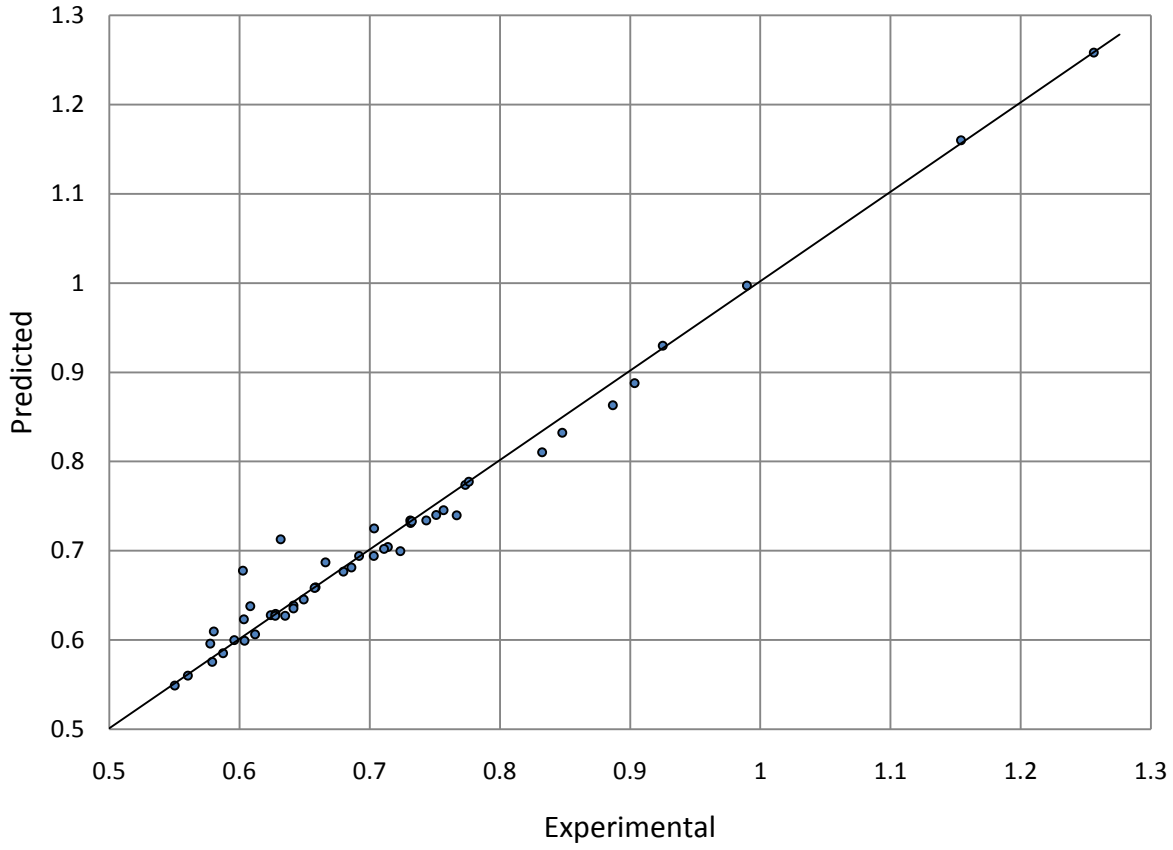

Figure 5.15: Validation plot of experimental versus predicted values of kinematic viscosity with "Octane - Toluene – Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.15: The predictive performance of neural network for "Octane – Toluene – Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
1.78264	12.91999	0.01296	0.98004

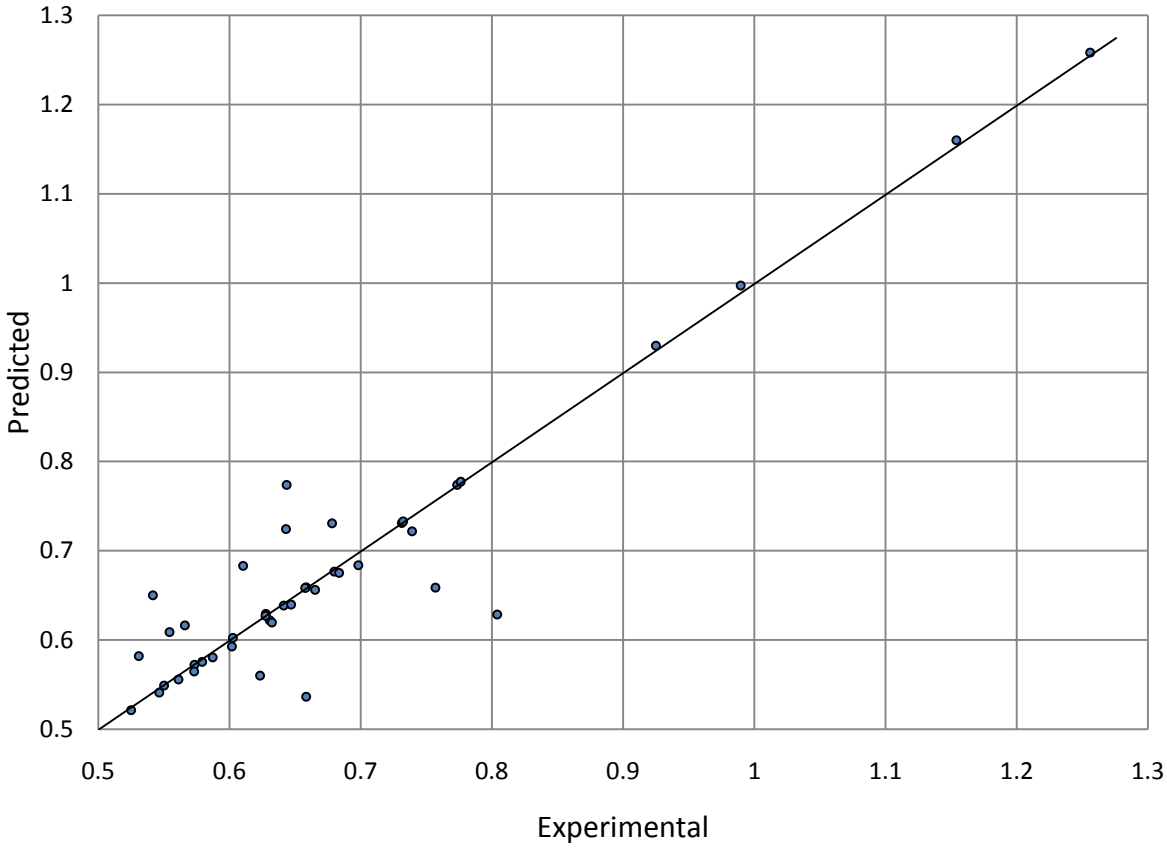

Figure 5.16: Validation plot of experimental versus predicted values of kinematic viscosity with "Heptane - Octane –Toluene – Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.16: The predictive performance of neural network for "Heptane – Octane –Toluene – Cyclohexane - Ethylbenzene" system.

%AAD	%AAD (Max)	%AAD (Min)	R^2
4.36113	21.77939	0.01296	0.88951

5.2 Comparison of results

With the goal of thoroughly evaluating the ANNs developed in this work, other models with identical multi-component systems were selected as a basis of comparison. The generalized McAllister model was preferred as an alternative model for comparison because of its many advantages: it has been widely investigated in the literature, it has been classified as a predictive model, and its predictive results have been superior to those from other models, as indicated by its percent absolute average deviation (%AAD). In order to compare the results from the developed neural network model and those from the generalized McAllister model, Equation 4.1 was used to calculate the average absolute deviation (AAD).

For the 10 ternary subsystems, Table 5.17 and Figure 5.17 show the results of the comparison of the %AAD of the neural network versus the generalized McAllister model. The minimum AAD of the neural network and of the Generalized McAllister model are 0.25856% and 2.49%, respectively. Furthermore, the maximum AAD is 1.82863% for the neural network and 5.12% for the generalized McAllister model. In general, the neural network outperformed the generalized McAllister model with respect to the ternary subsystems: the overall AADs are 0.864647% and 3.531% respectively. These results are shown in Figure 5.20 and Table 5.20.

The results of the comparison of the %AAD of the neural network versus those of the generalized McAllister model for the five quaternary subsystems are displayed in Table 5.18 and Figure 5.18.

The minimum AAD of the neural network and of the generalized McAllister model are 0.49160% and 2.15%, respectively. The maximum AAD is 1.78264% for the neural network and 3.83% for the generalized McAllister model. In general, the neural network outperformed the generalized McAllister model with respect to the quaternary subsystems also: the overall AADs are 1.1298% and 3.176%, respectively. These results are presented in Table 5.20 and Figure 5.20.

The quinary system used in this work was only one system consisting of 44 test points. The literature contained limited data for training, which constrained the development of the model and produced less than excellent prediction results for the quinary system. For the quinary system, the neural network had greater %AAD than that produced by the generalized McAllister model: the overall AADs were 4.3611% and 1.18%, respectively. These results are shown in Table 5.19 and Figure 5.19. Since the kinematic viscosity data of quinary mixtures are rarely available in the literature, an additional approach was considered in order to further validate the predictive performance of the neural network model. A new neural network model with identical parameters and conditions was utilized, and the data used for training and testing was produced with a new technique: all the datasets were grouped together and divided into two sets. The first set was the testing set, which was randomly selected and made up of 15% of the total dataset, and the second set was made up of the remaining 85% of the dataset, and were used as the training set. The results of this model are presented in Table 5.21 and Figure 5.21: the overall AAD is 0.77892%.

When this result is compared with the combined %AAD of generalized McAllister model for the Ternary, Quaternary and Quinary systems that is 2.629% , it is clear that the neural net work outperformed the generalized McAllister model

Table 5.17: Comparison of ternary subsystems %AAD of Neural Network versus 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

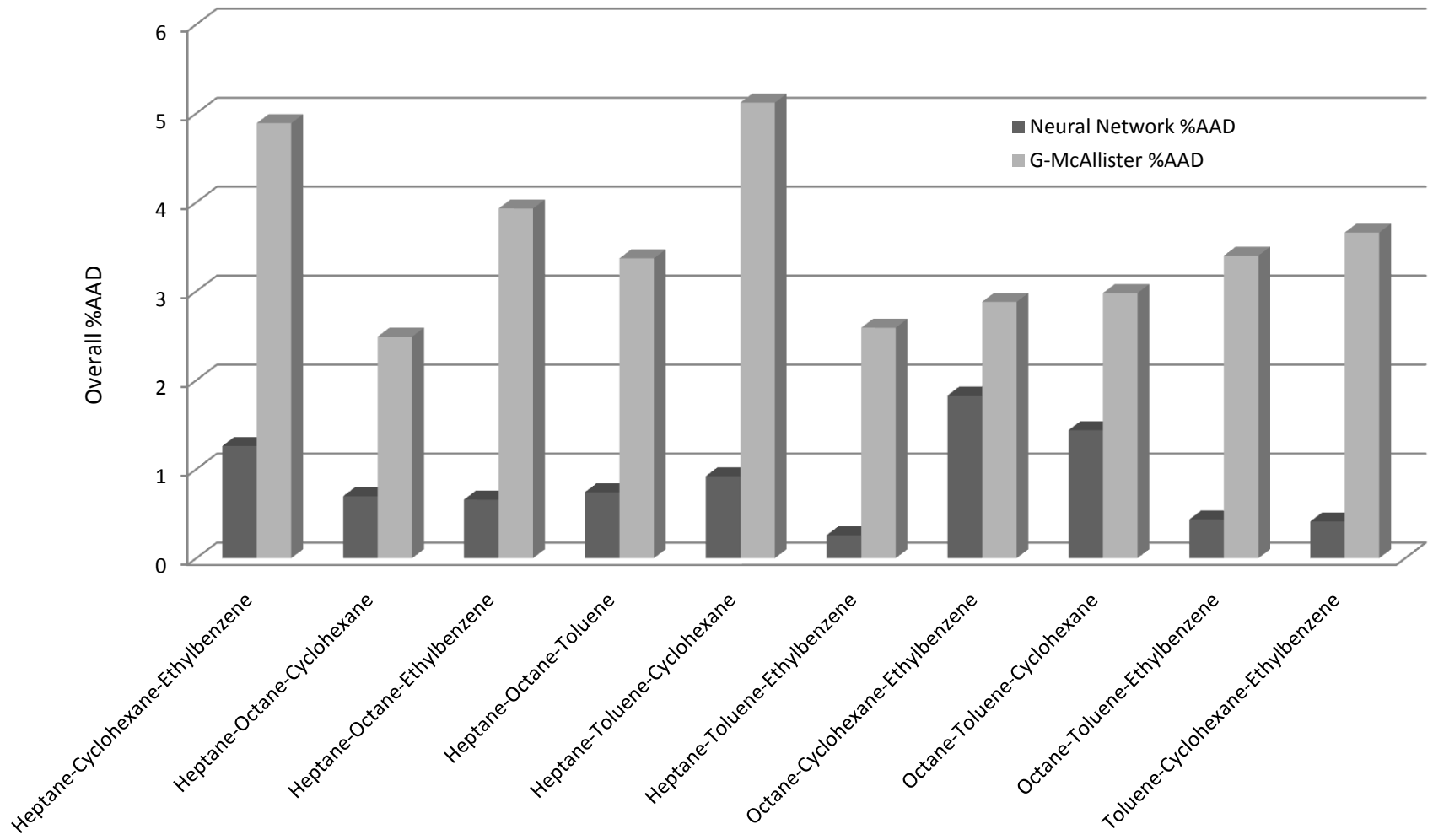

Figure 5.17: The %AAD of the Neural Network and General McAllister Models for Ternary Systems (293.15 - 313.15 K)

Table 5.18: Comparison of quaternary subsystems %AAD of Neural Network versus Generalized McAllister Model.

System	Temperature (K)	Neural Network %AAD	McAllister Model %AAD
Heptane-Octane-Cyclohexane-Ethylbenzene	293.15-313.15	1.31412	2.83
Heptane-Octane-Toluene-Cyclohexane	293.15-313.15	1.40294	3.35
Heptane-Octane-Toluene-Ethylbenzene	293.15-313.15	0.49160	2.15
Heptane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	0.65777	3.83
Octane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	1.78264	3.72

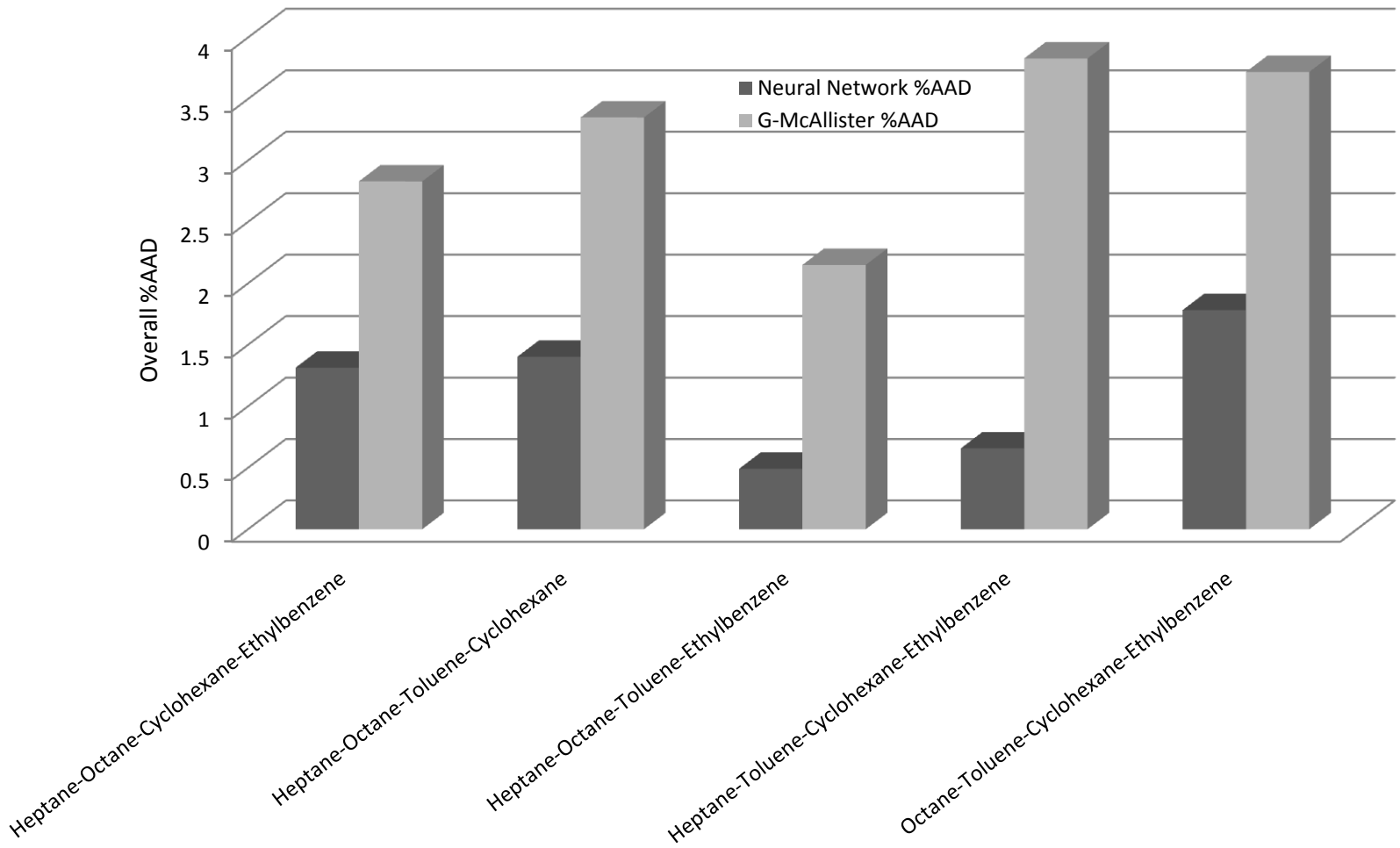

Figure 5.18: The %AAD of the Neural Network and General McAllister Models for Quaternary systems (293.15 - 313.15 K)

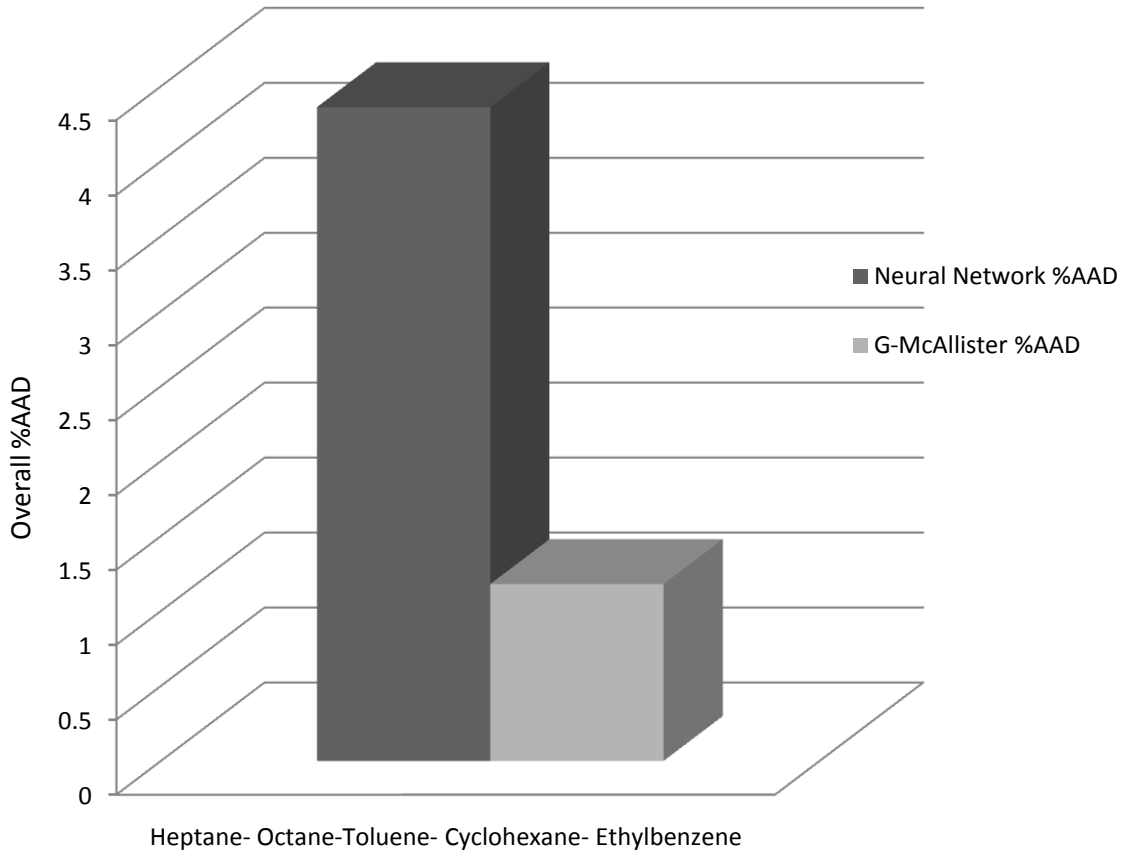

Figure 5.19: The %AAD of the Neural Network and General McAllister Models for Quinary System (293.15 K - 313.15 K).

Table 5.19: Comparison of quinary subsystems %AAD of Neural Network versus Generalized McAllister Model.

System	Temperature (K)	Neural Network %AAD	McAllister Model %AAD
Heptane-Octane-Toluene-Cyclohexane-Ethylbenzene	293.15-313.15	4.3611	1.18

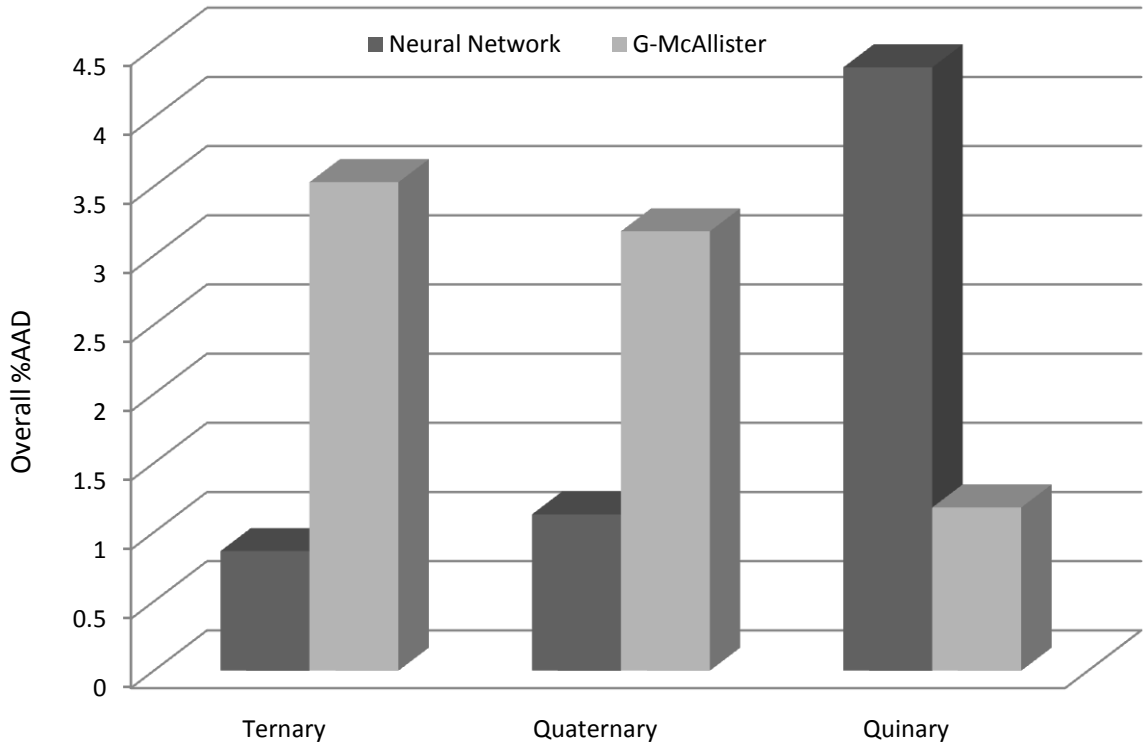

Figure 5.20: Overall %AAD of Neural Network and General McAllister Models for the Ternary, Quaternary and Quinary systems

Table 5.20: Comparison of %AAD of Neural Network versus Generalized McAllister Models for Ternary, Quaternary and Quinary systems.

System	Temperature (K)	Neural Network %AAD	McAllister Model %AAD
Ternary	293.15-313.15	0.864647	3.531
Quaternary	293.15-313.15	1.1298	3.176
Quinary	293.15-313.15	4.3611	1.18

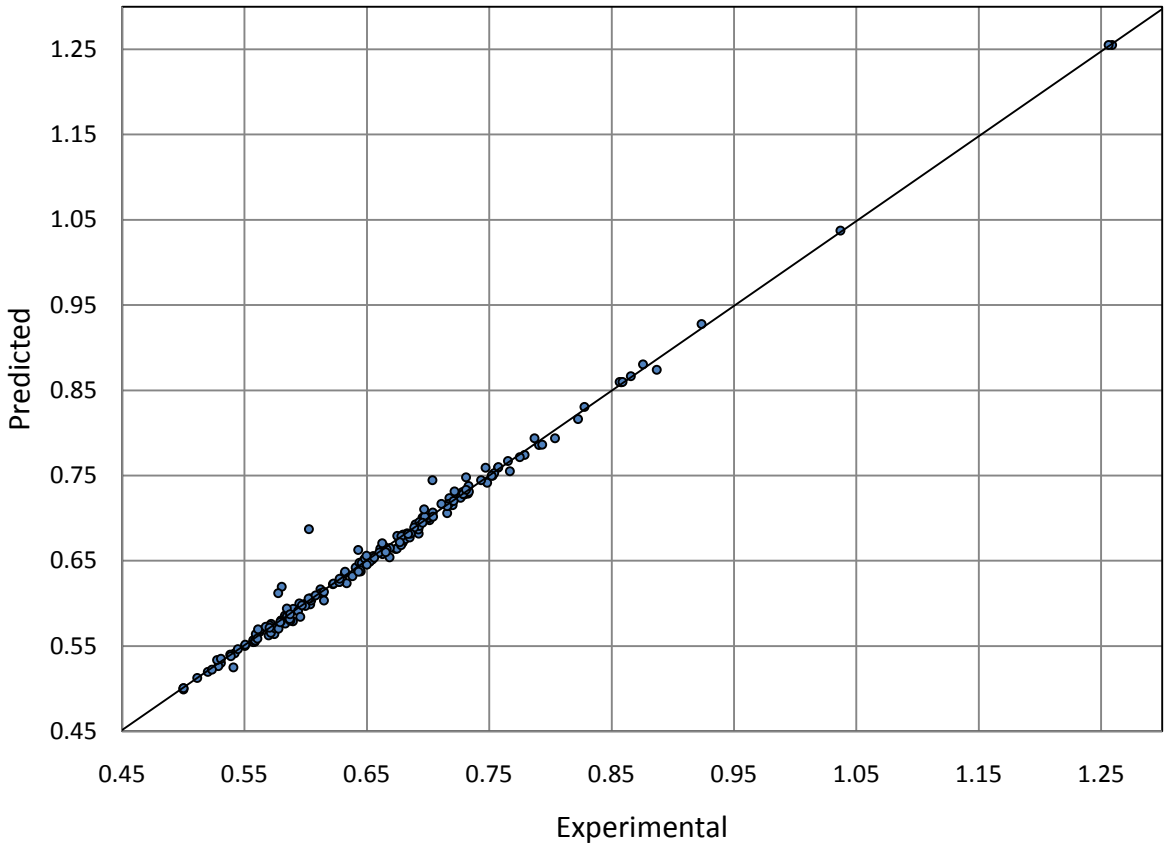

Figure 5.21: Validation plot of experimental versus predicted values of kinematic viscosity utilizing 15% of the dataset for testing "Heptane - Octane - Toluene - Cyclohexane - Ethylbenzene" for the entire temperature range 298 K - 313 K data set.

Table 5.21: The predictive performance of neural network for "Heptane - Octane - Toluene - Cyclohexane - Ethylbenzene" system, utilizing 15% of the dataset for testing.

%AAD	%AAD (Max)	%AAD (Min)	R^2
0.77892	14.10920	0.00452	0.99216

Chapter 6

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. The quinary system was the only system that had a higher AAD with the neural network, and for which the generalized McAllister model outperformed the ANN model. An additional model was developed in order to revalidate the performance of the neural network, with the data employed for the new model consisting of all the datasets combined and divided into two sets: 85% for training and 15% for testing. This approach produced remarkably improved results: the overall AAD with the new network was 0.77892%.

It can be concluded that the deviation from ideality represented by the ANN method provides a good estimate for predicting kinematic viscosity. The validity of the model was proved at temperatures of 293.15, 298.15, 308.15, and 313.15 K.

The ANN overcame the limitations of other prediction models, particularly the generalized McAllister model. This is mainly because the ANN technique can be developed without including the assumptions and estimation of variables which are essential in deriving the generalized McAllister model.

It is suggested that the developed neural network model for predicting kinetic viscosity to be retrained when more experimental data are available for quinary systems at a variety of temperatures.

Bibliography

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.

Aleksander, I., & Morton, H. (1990). *An introduction to neural computing.* New York,: Van Nostrand Reinhold Co.

Anderson, J. A. (1972). A simple neural network generating interactive memory. *Mathematical Biosciences* , 197-220.

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

Andrade, E., & Da, C. (1934). A theory of the viscosity of liquids. *Philosophical Magazine* .

Asfour, A., Cooper, E., Wu, J., & Zahran, R. (1991). Prediction of the McAllister model parameters from pure component properties for liquid binary n-alkane systems. *Industrial and Engineering Chemistry Research* , 1666}1669.

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.

Cheng, B., & Titterington, D. (1994). Neural Networks: a review from a staistical perspective. *Statistical Science, 9*, 2-54.

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

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., & Jackel, L. (1987). Large automatic learning rule extraction and generalization. *Complex systems* , 877-922.

Dowla, F., & Rogers, L. (1995). *Solving Problems in Environmental Engineering and Geosciences With Artificial Neural Networks.* Cambridge: MIT Press.

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.

Elkamel, A. (1998). An artificial neural network for predicting and optimizing immiscible flood performance in heterogeneous reservoirs. *Computers & Chemical Engineering , 22* (11), 1699-1709.

Elman, J. L. (1990). Finding Structure in time. *Cognitive Science,14(2),* 179-211.

Ewell, R., & Eyring, H. (1937). Theory of the Viscosity of Liquids as a Function of Temperature and Pressure. *The Journal of Chemical Physics, 5*, 726-736.

Fausett, L. (1994). *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications.* NJ.: Prentice-Hall.

Fukushima, K. (1975). A Self-Organizing Multilayered Neural Network. *Biological Cybernetics* (20), 121-136.

Glasston, S., Laidler, K., & Eyring, H. (1941). *The Theory of Rate Processes.* New York: McGraw-Hill .

Grossberg, S. (1987). Competitive Learning: from Interactive Activation to Adaptive Resonance. *Cognitive Science* (11), 23-63.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation.* New Yrok: MacMillan Publishing Company .

Hirschfelder, J., Curtiss, C., & Bird, R. (1954). *Molecular Theory of Gases and Liquids.* New York: John Wiley and Sons, Inc.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational properties. *Proceedings of the National Academy of Sciences , 79*, 2554-2588.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359-366.

Jasper, J. J. (1972). The Surface Tension of Pure Liquid Compounds. *Journal of Physical and Chemical Reference Data, 1*, 841-1009.

Jordan, M. I. (1986). Attractor Dynamics and Parallelism in a Connectionist Sequential Machine. *Eighth Annual Conference of the Cognitive Science Society* (pp. 531-546). Amherst, Massachusetts: Erlbaum.

Kesavan, J. G. (1996). Pharmaceutical granulation and tablet formulation using neural networks. *Pharmaceutical Development and Technology, 1* (4), 391-404.

Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers, 21*, 353-359.

Kohonen, T. (1988). *Self-organization and Associative Memory.* New York: Springer-Verlag.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics, 43*, 59-69.

Kohonen, T. (1988). The Neural Phonetic Typewriter. *IEEE Computer , 21* (3), 11-22.

Masters, T. (1994). *Practical Neural Network Recipes in C++*. Boston: Academic Press.

McAllister, R. (1960). The viscosity of liquid mixtures. *American Institute of Chemical Engineers, 6*, 427 - 431.

McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel distributed processing: explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

McCullough, W. S., & Pitts, W. (1943). A Logical Calculus Of ideas immanent in nervous activity. *Bulletin of mathematical biology , 5*, 115.

Minsky, N. a. (1969). *Perceptrons.* Cambridge Massachusetts: MIT Press.

Nelson, M., & Illinworth, W. (1990). *(Nelson, M. & Illinworth, W. (1990) A Practical Guide to Neural Nets. Addison-Wesley Publishing Company Inc.* Addison-Wesley Publishing Company Inc.

Network, D. N. (1988). *DARPA Neural Network Study.* AFCEA International Press.

Nhaesi, A., & Asfour, A. (1998). Prediction of the McAllister Model Parameters from Pure Component Properties of Regular Binary Liquid Mixtures. *Industrial & Engineering Chemistry Research, 37* (12), 4893–4897.

Nhaesi, A., & Asfour, A. (2000). Prediction of the viscosity of multi-component liquid mixtures:: a generalized McAllister three-body interaction model . *Chemical Engineering Science, 5* , 2861-2873.

Nhaesi, A., & Asfour, A. (2000). Predictive models for the viscosities of multicomponent liquid n-Alkane and regular solutions. *The Canadian Journal of Chemical Engineering, 78*, 355-362.

Nissan, A., Clark, L., & Nash, A. (1940). An improved rotary viscometer. *Journal of Scientific Instruments, 17*, 33-38 .

Paker, D. (1985). *Learning-Logic: Casting the Cortex of the Human Brain in Silicon.* Massachusetts Institute of Technology. Cambridge: Center for Computational Research in Economics and Management Science.

Pedersen, K., Fredenslund, A., Christensen, P., & Thomassen, P. (1984). Viscosity of crude oils. *Chemical Engineering Science, 39*, 1011-1016.

Rabaey, J. M. (1995). *Digital Integrated Circuits: A Design Perspective.* Prentice Hall.

Reid, R., Praunitz, J., & Sherwood, T. (1987). *The properties of gases and liquids.* New York: McGraw-Hill.

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.

Schalkoff, R. *Artificial neural networks.* McGraw-Hill Higher Education.

Swingler, K. (1996). *Applying Neural Networks: A Practical Guide.* New York: Academic Press.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* Ph.D Thesis, Harvard University, Cambridg, MA.

Widrow, B. a. (1960). Adaptive switching circuit. *IRE WESCON* , 96-104.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* , 270-280.

Zurada, J. M. (1992). *Introduction To Artificial Neural Systems, Boston:*. Boston: PWS Publishing Company.