ISSN (e): 2250 – 3005 || Vol, 04 || Issue, 6 || June– 2014 ||
International Journal of Computational Engineering Research (IJCER)

Modelling of Vapour Liquid Equilibrium by Artificial Neural Networks

[1,] Dr.B.Karunanithi, [2,] Sweta Shriniwasan, [3,] Bogeshwaran.K,
[1,] *Department of Chemical Engineering, SRM University, Kattankulathur, 603 203. India.*
[2,] *Department of Chemical Engineering, SRM University, Kattankulathur,603 203. India*
[3,] *Department of Chemical Engineering, VeltechHightechDr.RangarajanDr.Sakunthala Engineering College, Avadi.600 062.*

ABSTRACT:
Vapour liquid equilibrium is condition wherein the liquid and vapour state of the components of a system are in equilibrium with each other. Conventionally, the vapour liquid equilibrium data is evaluated using the thermodynamic models, namely the equation of state (EOS), and the activity co-efficient models. The models falling under these categories are Peng-Robinson model, Margules model, vaanLaar model, Wilson's model, NRTL, UNIQUAC and UNIFAC model. VLE data is required in designing distillation columns and any doubt or inaccuracy in the prediction of the VLE data results in variation in design parameters which leads to variations in purity of the distillate, number of theoretical plates, reflux ratio and energy consumption which consequently leads to variation in cost. The VLE data predicted by the existing thermodynamic models show deviations from the experimental data. Hence, an Artificial Neural Network (ANN) model has been developed to predict the VLE so as to minimize the deviations from the experimental values. Several binary systems (1 simple and 6 azeotropicsystems) have been considered and VLE data has been predicted using the ANN, Margules and the van Laar models. The Root Squared Mean Deviation (RSMD) of predicted values has been calculated with respect to the experimental values. It has been observed that the data predicted by the ANN model is more accurate as compared to the Margules and van Laar models.

KEYWORDS: *vapour liquid equilibrium, ANN, RSMD, binary systems, model.*

I. INTRODUCTION

Vapour liquid equilibrium is condition wherein the liquid and vapour state of the components of a system are in equilibrium with each other. In other words, it is the state of the system at which rate of condensation is equal to the rate of evaporation. Vapour liquid equilibrium data is required for the design, analysis and control of distillation columns.
Conventionally, the vapour liquid equilibrium data is evaluated using the thermodynamic models, namely the equation of state (EOS), and the activity co-efficient models. The models falling under these categories are Peng-Robinson model, Margules model, vaanLaar model, Wilson's model, NRTL, UNIQUAC and UNIFAC model.

The thermodynamic methods mentioned above use linear and nonlinear regression techniques to represent the relations among the variables of a given system. The relationship between the physical and thermodynamic properties is highly non-linear, and consequently an artificial neural network (ANN) can be a suitable alternative to model and develop a non-linear relation between the input and the output parameters. ANN is an efficient methodology to approximate any function with finite number of discontinuities by learning the relationships between input and output vectors.

VLE data is required in designing distillation columns and any doubt or inaccuracy in the prediction of the VLE data leads to design of distillation columns with variation in various parameters. The VLE data predicted by the existing thermodynamic models show deviations from the experimental data, though they are adequate for most engineering applications.

Moreover, with the increase in the use of software packages for data evaluations, the use of artificial neural networks can be integrated along with the existing software packages. Artificial neural network (ANN) is an evolutionary computation or optimization technique. The accuracy of the computed values is said to be better than many other mathematical models. In this thesis, it is intended to develop an artificial neural network model to predict the vapour liquid equilibrium values for 7 binary systems and compare these predicted values to that predicted by the existing mathematical models like the Margules and the van Laar models.

Existing models

Several empirical models have been developed to estimate the vapour liquid equilibrium data and the activity coefficients of various systems. The Margules and the van Laar are two of the empirical models that have been developed to estimate the activity coefficient. Modern activity coefficient models are based on the local-composition concept, which was introduced by Wilson (1964). Due to molecular size and intermolecular forces, local compositions are assumed to take into account the short range orders and non-random molecular orientations inside a liquid solution. Two of the most widely used models are the Non-Random-Two-Liquid (NRTL) developed by Renon and Prausnitz (1968) and the Universal QUAsi-Chemical (UNIQUAC) developed by Abrams and Prausnitz (1975). These models are capable of correlating experimental activity coefficients for a species in a liquid solution over a wide composition and temperature range. They are also capable of interpolating and/or extrapolating the experimental activity coefficients for a wide range of temperatures and compositions based on a few experimental points.

In the absence of experimental data, group contribution methods have been devised to predict the activity coefficients of a system. In these methods, atoms in a chemical compound are grouped to form functional groups that are assumed to have their own physical and chemical identity (Fredenslund*et al*., 1975). Wilson and Deal (1962) introduced the Analytical Solutions of Groups (ASOG) method. Fredenslund et al. (1975) developed the Universal Functional-group Activity Coefficients (UNIFAC) method to predict activity coefficients based on molecular functional groups contribution. UNIFAC is one of the most prominent methods that uses a combinatorial and a residual part with functional groups parameters such as: group volume, group surface area, and binary group interactions to predict the activity coefficients.

The group-contribution methods mentioned above use linear and non-linear regression techniques to represent the relations among the variables of a given system. The relationship between the physical and thermodynamic properties in a system is highly non-linear. Hence, an artificial neural network(ANN) can be a suitable alternative to model or to predict the vapour liquid equilibrium data.

Research so far

Maria Iliuta*et al* (1999), proposed artificial neural network correlations for the prediction of vapour-liquid equilibrium for mixed dual-solvent single electrolyte systems, and validated over an extensive VLE database (2900 data points, 16 binary solvents, 24 salts, 11 cations, 6 anions)[7]. Performance of these correlations to predict the vapour phase mole fraction, equilibrium temperature and total pressures was compared with the experimental data and the data generated by the UNIFAC model. The mean absolute deviations in the predicted data were found to be minimized.

Bilgin*et.al* (2003), employed a neural network model to predict VLE data for six different binary systems having different chemical structures and solution types in various conditions[1]. The VLE data was also predicted by the UNIFAC model. It was observed that the values predicted by the ANN model show close agreement with the experimental values.

Mehmet Bilgin (2004), employed a neural network model to calculate the isobaric vapour-liquid equilibrium of binary systems composed of different chemical structures, which do not show azeotropic behaviour[2]. Results generated by the ANN model were compared to those generated by the UNIFAC and the Margules model. In all cases, the deviations between the experimental activity co-efficients and those calculated by the neural network were less than those obtained by those obtained by the Margules and the UNIFAC models.

Govindarajan and Sabarathinam (2006), used the radial basis neural networks, a type of artificial neural networks to predict the VLE data for 4 binary systems and 1 ternary system[8]. A neural network based on the equation of state was use to predict the liquid phase composition and vapour phase compositions at the given conditions of temperature and pressure. Theperformance of the network was evaluated on the basis of an overall absolute error and root mean square error specified by the difference in the desired and the actual outputs. It was concluded that this technique to predict VLE data is efficient, reliable and robust.In 2006, Rajesh *et.al*, used ANN for the prediction of the prediction of equilibrium solubility of CO_2 in aqueous alkanolamines. An ANN model was employed to predict the VLE data of two systems, viz:- CO_2 – N-methyldiethanolamine(MDEA) – H_2O system and CO_2 – 2-amino-2-methyl-1-propanol(AMP)-H_2O system. The predictions made by the ANN model were found to be in the accuracy of $\pm$5% for 95% of the data[11].

Ghaemi*et.al* (2008), developed an ANN model for the prediction of VLE data in aqueous solutions of electrolytes[5]. VLE data for ternary system of NH_3- CO_2 – H_2O were predicted using the ANN model which was compared with the predictions of some thermodynamic models.

Moghadassi*et.al* (2009), developed an ANN model to predict the VLE data of high pressure systems[3]. Moghadassi*et.al* (2011), developed a model for predicting the VLE data for binary systems containing propane [4]. Four binary refrigerant systems containing propane were considered. Results generated by the ANN model were compared with those generated by Margules and vaanLaar models. The ANN model showed superiority over the other thermodynamic models.

Pandharipande*et.al* (2012) developed a model for the evaluation of VLE data for ten binary systems, results obtained indicated minimum error is obtained in the case of ANN models [9]. Pandharipande*et.al* (2012), modeled combined VLE of four quaternary mixtures using artificial neural network. It was observed that in the ANN model, the error difference between the predefined value and output calculated is minimized[6].Nasri et al (2012) developed an ANN model to predict the VLE of a carbon dioxide methanol system at high pressure. Predicted values using ANN are satisfactory.

Inference from literature review

A general overview of the literature shows that the experimental data or a set data points (experimental) available in literature are fed to the neural network for pattern recognition. Pattern recognition or the relation between the input and output is generated by the ANN. Thus, in other words, the ANN generates a mathematical model. Once this model is generated, it is tested for its accuracy with a set of new data points which have not been fed earlier to the ANN for pattern recognition. By entering a new set of data points for testing, the efficiency of prediction of the developed ANN model can be known.

As per the literary review, it was inferred that only a few azeotropic systems have been considered, hence in this report, six azeotropic systems have been considered to show the versatility of ANN in predicting VLE data. Systems considered in this report have not been considered so far. If ANN model is proved to be better than the existing thermodynamics models then, it can be easily integrated with the design and simulation softwares which are generally used for the VLE data estimation.

II. THEORETICAL BACKGROUND

Thermodynamic models

Basically two kinds of thermodynamic models are used to evaluate the VLE data.

They are:

1.Activity co-efficient models (Excess Gibbs Free Energy Models)

2.Equation of state models

Activity co-efficient models have been widely used for the evaluation of VLE data. In this thesis also, the values predicted by the ANN model will be compared with the values computed using the activity co-efficient models, viz: Margules and the van Laar models.

Basic equation for Vapour-liquid Equilibrium

Consider a closed system consisting of co-existing vapour and liquid phases, each phase containing 'c' each components in a state of equilibrium at constant temperature (T) and pressure (P). The criterion for equilibrium between the two phases is given by

$f_i^l = f_i^v \quad i = [i = 1,2,3,\ldots..,c]$ (1)

where, f_i^l , f_i^v are the fugacities of the pure components of the liquid and vapour.

Equation (1) can be rewritten as

$\gamma_i x_i f_i^* = \Phi_i^v y_i P$ (2)

The standard state fugacity is given by

$f_i^* = P_i^s \Phi_i^s \exp\{\frac{v_i^l(P - P_i^s)}{RT}\}$ (3)

where, P_i^s = saturation pressure of component i at temperature T

v_i^l = molar volume of liquid for component i

Φ_i^s= fugacity coefficient of component i at saturation pressure

Substituting (3) in (2)

$\gamma_i x_i P_i^s = \frac{\Phi_i^v}{\Phi_i^s} y_i P \exp\{-\frac{v_i^l(P - P_i^s)}{RT}\}$-- - (4)

The above equation is the basic equation for vapour liquid equilibrium. It provides a relation between among the variables T, P, x_i's and y_i's.

At low pressures (uptoatleast 1 bar), the vapour phase can be assumed to behave like an ideal gas and hence$\Phi_i^v = 1$, and$\Phi_i^s = 1$. At low pressures the Poynting correction factor $\exp\{\frac{v_i^l(P - P_i^s)}{RT}\}$ is negligibly small and it is approximately equal to unity. At low to moderate pressures (upto 10 bar), Φ_i^v and Φ_i^s are equal to each other and hence it is reasonable to assume $\Phi_i^v / \Phi_i^s = 1$.

Thus at low to moderate pressures

$$\gamma_i x_i P_i^s = y_i P \qquad (5)$$

or $\gamma_i = \frac{y_i P}{x_i P_i^s} (I = 1,2,3,\ldots..,c)$

The value of γ_i is evaluated by the thermodynamic models like Margules, van Laar, etc.

For a binary system, the activity coefficients can be evaluated as

Margules equation

$$\ln \gamma_1 = x_2^2 \{ A_{12} + 2 *(A_{21} - A_{12}) * x_1\} \qquad (6)$$

$$\ln \gamma_2 = x_1^2\{ A_{21} + 2 *(A_{12} - A_{21}) *x_2 \} \qquad (7)$$

where, A_{12},A_{21} are the Margules interaction parameters for the binary system consisting of components 1 and 2.

vanLaar equation

$$\ln \gamma_1 = \frac{A}{(1+\frac{Ax_1}{Bx_2})^2} \qquad (8)$$

$$\ln \gamma_2 = \frac{B}{(1+\frac{Bx_2}{Ax_1})^2} \qquad (9)$$

where, A and B are van Laar constants and subscripts 1 and 2 stand for components 1 and 2 respectively in the binary system.

Artificial neural networks (ANN): General Overview

Artificial neural networks (ANN) are non-linear information processing paradigm, which are built from interconnected elementary processing devices called neurons. They are inspired by the way the human brain processes information. ANNs like people, learn by an example. ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections/weights that exist between the neurons, which is applied to ANNs as well. ANNs can also be defined as parameterized computational nonlinear algorithms for data/signal/image processing. These algorithms are either implemented on a general purpose computer or built on a dedicated hardware.

An efficient way of solving a complex problem is to divide or decompose it into simpler elements in order to be able to understand it. Also, simple elements may be gathered to produce a complex system. Use of networks is one of the approaches to achieve this. There are a large number of networks of different types. They are all characterized by the following components: a set of nodes, and connections between the nodes.

The nodes, which are analogous to the neurons in the biological nervous system are computational units. They receive inputs and process them to obtain an output. This processing may be simple (summing of input) or complex (a node contains another network). The connections determine the flow of information between the nodes. The interactions of the nodes through the connections lead to a global behaviour of the network which cannot be observed in the elements of the network. In other words, abilities of the network supercede the ones of its elements.

Elements called neurons (nodes), process the information. The signals are transmitted by means of connection links. The links possess an associated weight, which is multiplied along with the input signal(net input) for any typical neural net. The output signal is obtained by applying activations to the net input.

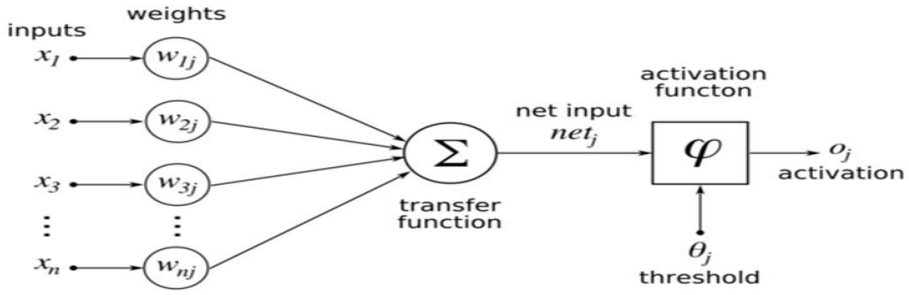


Figure 3.1 – General representation of an artificial neural network [iv]

The figure above represents a simple structure of an artificial neural network. It has 'n' input neurons (x_1, x_2,……x_n), w stands for the interconnected weights. The suffix 'ij' is added to the weights where, i stands for the corresponding input neuron number, j stands for the number corresponding to the weighted connections. In the above figure, there is only a single layer, there can be as many layers as the user defines. When there are more than one layers, the neural network is called a multilayer net. The value o_j stands for the output which is processed by the net.

The transfer function is the summation of the weighted inputs or also called as the net input. It can be represented mathematically as

$net_j = \sum_{i=0}^{n} WijXi$ + bias;

The output is obtained by the activation of the net input. It is given by,

$o_j = f(net_j)$;

The activation functions are of several types. In this case, two activation functions have been used namely the log-sigmoid and the linear activation functions. The mathematical representations of these functions are,

Log-sigmoid function:

$f(x) = \frac{1}{1+\exp(-x)}$, where x is any variable.

Linear function:

f(x) = x, where x is any variable.

The arrangement of neurons into the layers and the pattern of connection within and in-between layer are generally called as the architecture of the net. The neurons within a layer are found to be fully interconnected or not interconnected. The number of layers in the net can be defined to be the number of layers of weighted interconnected links between particular slabs of neurons. If two layers of interconnected weights are present, then it is found to have hidden layers. The various types of network architecture are feed-forward, feedback, fully interconnected net, competitive net etc.The feed-forward network architecture has been used to evaluate the network. It can be represented as:

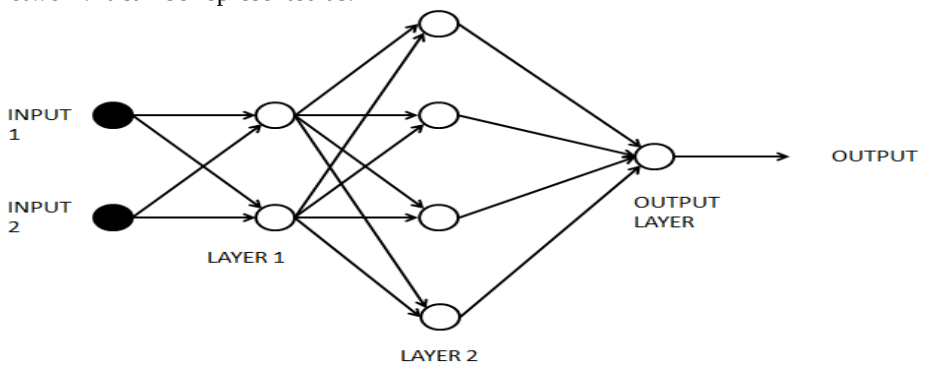


Figure 3.2 – General representation of a feed forward neural network architecture

Feed forward networks may have a single layer of weights where the inputs are directly connected to the outputs or it may consist of multiple layers with intervening sets of hidden unit units. Neural networks use hidden units to create internal representations of the input patterns. The figure 3.2, represents a multi-layer feed-forward network consisting of an input layer, two hidden layers and an output layer. In multilayer nets, signal flow from input units to output units in a forward direction. It can be used to solve complex problems.

The neural net learns or recognises a pattern by the process of learning or training. In the learning/training process, the network is presented with a set of values of the input and its corresponding output. In this process, the network learns or establishes a relation between the input and the output by setting weights and biases accordingly. This process is also called as supervised learning as for each input value, the value of the desired input is provided by the user and the network establishes a non-linear relation between the input and the output.

A training algorithm and function is employed for the training of the values. The back-propagation training algorithm and the resilient back-propagation training function is used. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by the user. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Properly trained back-propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

There are four steps in the process of predicting the VLE data using neural networks:
1.Assemble the training data.
2.Create the network object.
3.Train the network.
4.Simulate the network response to new inputs

In this learning pattern, the back-propagation of errors takes place. Initially the random values of weights are assumed. The evaluation of the network takes place which is compared with the predefined mean squared error (MSE) value or error difference between the output and the target. MSE is given by the equation:

$$\text{MSE} = \frac{1}{N}\sum_0^N (y_c - y_t)^2$$

i.e. Error (E) = F(weighted inputs, target output)
If the desired value of MSE is reached then the evaluation of the network is stopped, else the weights are updated to new values to evaluate the network and check the value of MSE. If MSE is reached the evaluation stops or else the weights are updated and the cycle continues till the predefined MSE is reached or till the maximum limit of predefined iterations (epochs) are reached.
The updation of weights takes place by the method of resilient propagation[16][17]. Resilient propagation is an effective learning scheme. It performs a direct adaption of weight step based on local gradient information. For each weight an individual update value Δ_{ij} is used which determines the size of the weight update. An adaptive update-value evolves during the training process based on local sight of the error function.
Every time the partial derivative of the corresponding weight w_{ij}changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update-value $\boldsymbol{\Delta_{ij}}$is decreased by the factor η**-.** If the derivative retains its sign, the update-value is slightly increased in order to accelerate convergence in shallow regions. Given as:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ \eta^- * \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}^{(t-1)}, & \text{else.} \end{cases}$$

Once the update-value for each weight is adapted, the weight-update itself follows a very simple rule:

- if the derivative is positive (increasing error), the weight is decreased by its update-value
- if the derivative is negative, the update-value is added

$$\Delta w_{ij} = \begin{cases} -\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & \text{else} \end{cases}$$

Updation is carried out by the formula,
$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$.
There is one exception: If the partial derivative changes sign, i.e. the previous step was too large and the minimum was missed, the previous weight-update is reverted.

$$w_{ij}^{(t+1)} = -\Delta w_{ij}^{(t-1)}, \quad \text{if } \frac{\partial E^{(t-1)}}{\partial w_{ij}} * \frac{\partial E^{(t)}}{\partial w_{ij}} < 0$$

Due to that 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting the differential of the error w.r.t the weights to zero in the $\boldsymbol{\Delta_{ij}}$adaptation rule. The update-values and the weights are changed every time the whole pattern set has been presented once to the network (learning by epoch/iteration). In other words, the update values and weights are changed after every epoch/iteration till the maximum number of iterations or till the error between the computed and the target output reach the predefined MSE value.

III. METHODOLOGY

The main aim of this project was to predict the VLE data based on the non-linear correlation generated by the artificial neural network. The artificial neural network model was developed with the help of a code using Matlab 7.0. Firstly, a basic code was executed and it was tested for a binary system of Methanol(1) – Acetone(2) at 101.325 kPa. There were modifications made in the basic code to obtain a suitable code which has been used to evaluate VLE data of all the systems considered in this report.

As mentioned in the previous chapter, the neural network learns the patterns for a set of values and it applies it for a new set values. Hence, an input data is provided with its corresponding target or desired target value. The neural network studies the set of input and its corresponding output and it generates a non-linear model relating the input to the output.

Steps in developing the model

The development of the ANN model consists of the following steps which have been written as a Matlab code

1.Identification of the number of inputs and outputs

2.Designing an architecture for the neural network

3.Training the neural network with a set of data points

4.Simulating the trained neural network with a new set of data points which have not been used in the training stage.

The data simulated by the neural network for the new set of data points is compared with the data obtained by solving this set using the Margules and the van Laar models. The comparison of the data is reported in the form of a graph.

1.Identifying the number of inputs and outputs

In this step, the number of inputs and outputs are decided. For eg: in this work, binary systems have been considered and according to the phase rule, the number of degrees of freedom is given as :

$F = C - \pi + 2$

Where, F = number of degrees of freedom, C = number of components, π = number of phases. For a binary system in vapour liquid equilibrium conditions, C = 2, π = 2, so number of degrees of freedom is F = 2.

Thus two known parameters are taken as the input to the neural networks. There can be one or two outputs considered. Thus, the input to the ANN can be either T, x1 data or P, x1 data and y1 data is the desired output.

2. Designing an architecture for the neural network

Optimum network architecture has to be designed such that the convergence of the values or the training of the network is fast and the output obtained does not show much deviations from the experimental values. The number of layers, the number of neurons in each layer and the activation function for each layer is to be set to form a neural network.

3. Training of the neural networks for a set of data points

The set of data points used for training the neural network serves as the input to the system. The data points are trained using the following steps

a) Input data – the input set and the corresponding target output set is fed to the ANN.

b).Initialize training – the network structure is defined and the training function is defined. The parameter associated with the training function like the maximum number of epochs, minimum gradient, mean squared errors (MSE) limit is defined.

c).Epoch is set as 1 and the training is started.

d).Weights and bias of the network are initialized to random values

e).With the entered input value and the value of weights and bias the output values are calculated.

f).Deviation of the calculated output and the desired target output is calculated using MSE.

g).If MSE $\leq MSE_{minimum}$ evaluation is stopped else do step (h).

f) If number epochs $\leq epochs_{maximum}$ go to step (i), else stop.

g) Weights are updated on the basis of the training function and the number of epoch is increased by 1 and the steps are repeated from step (e).

Once, the training is stopped, a model is generated with fixed values of weights and biases forming a numerical non-linear model (relation) between the input and output.

1. Simulating the trained neural network with a new set of data points which have not been used in the training stage.

A new set of data input is provided to the trained network to evaluate the output value or the value of the mole fraction of the components in the vapour phase. This simulated data has been compared with the experimental values and the values of the mole fraction of the vapour phase computed by the Margules and the van Laar's model.

The VLE data required for training and testing has been taken from the explorer edition of the Dortmund Data Bank (DDBST) online. For each system, considered, a number of data points have been used for training and a different set has been used for simulation and testing.

Systems considered:
The systems considered and number of data sets used for training and testing are as given below
i.Chloroform(1) – Ethanol(2) system (308.15 K) (Data sets: Training – 23, Testing – 5)
ii. Methanol(1) – Acetone(2) system (101.325 kPa)(Data sets: Training – 18, Testing – 6)
iii.Methanol(1) – Hexane(2) system (333.15 K)(Data sets: Training – 25, Testing – 5)
iv.Methanol(1) – Benzene (2) system (101.33 kPa)Data sets: Training – 36, Testing – 6)
v.Benzene(1) – Acetonitrile(2) system (293.15 K)(Data sets: Training – 38, Testing – 7)
vi.Water(1) – m-Xylene(2) system (101.3 kPa)
(Data sets: Training – 15, Testing – 5)
vii.Hexane(1) – Cyclohexane(2) system (101.33 kPa)(Data sets: Training – 28, Testing – 6)
All the data sets considered have been mentioned in the appendices

Example

Considering the Methanol (1) – Acetone (2) system (101.325 kPa). The neural network is trained by the input and target data. The Matlab code given in appendix (A1) is executed, the neural network architecture is a feed-forward architecture consisting of input T-x_1 data and output y_1 data. The neural net architecture is as given in figure (4.1). The net consists of 1 input layer, 2 hidden layers with 2 and 4 neurons respectively and the output layer with one neuron. The activation function in the hidden layers is logsig and in the output layer is purelin. MSE set is 1e-6. Minimum gradient set is 1e-6, maximum weight change is set as 100. The maximum number of epochs is set as 100000. After the training the relational bias and weights obtained can be represented in the net as given below.

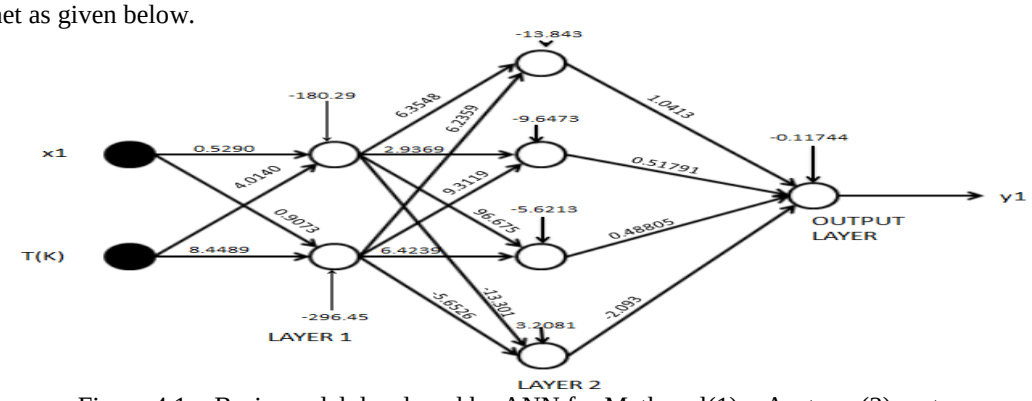


Figure 4.1 – Basic model developed by ANN for Methanol(1) – Acetone (2) system

With a new set of input values, the output is simulated and it is compared with the values given in literature and those calculated using a thermodynamic model. It is represented in the form of a graph as shown in appendix.

The training of the data takes 41656 epochs or iterations.But since the number of epochs taken is more, a neural network has to be designed in such a way that it predicts data with better accuracy and the epochs taken are less.

After designing and testing many neural network models, a neural network model was optimized to give accurate and faster convergence of results. The code written in Matlab 7.0 is as given in Appendix(A.2). For all the 7 systems considered, the neural network has been trained and the values are simulated using the trained network in each of the cases. The values have been calculated by the neural network, Margules and the van Laar models and a graph has been plotted to represent the result. The graphical representation of the calculated parameters has been presented in Appendix(B). For each system, the deviation between the calculated and the experimental values has been evaluated by the Root Mean Squared Deviation(RSMD) formula which is given by the expression.

$$\text{RSMD} = \sqrt{MSE}$$

$$\text{RSMD} = \sqrt{\frac{\sum_{i=1}^{N}(y_c - y_t)^2}{N}}$$

For all the systems, the model has been developed as per the Matlab code given in Appendix (A.2) (Optimized model). The model consists of four layers viz : an input layer with two inputs, two hidden layer with 30 neurons each and an output layer with one neurons. The activation function used in all the layers is the log-sigmoid (logsig) and the Resilient Propagation (trainrp) training algorithm is used. The parameters set for this training this network are:

1) Maximum number of epochs/ iterations = 100000

2) Minimum gradient = 1e-10
3) Mean squared error = 1e-9
4) Maximum weight change = 100

Using these parameters the systems have been trained to recognise the input patterns and to develop a non-relation between the input and output values. This relation developed is used to simulate the test data sets.

NAME OF THE SYSTEM	ROOT SQUARED MEAN DEVIATION (RSMD)		
	MARGULES	VAN LAAR	ANN
Chloroform(1) – Ethanol(2) system (308.15 K)	0.00790	0.02862	0.00500
Methanol(1) – Acetone(2) system (101.325 kPa)	0.00847	0.00885	0.00099
Methanol(1) – Hexane(2) system (333.15 K)	0.06728	0.05852	0.00569
Methanol(1) – Benzene (2) system (101.33 kPa)	0.06879	0.05889	0.00699
Benzene(1) – Acetonitrile(2) system (293.15 K)	0.01811	0.01981	0.00607
Water(1) – m-Xylene(2) system (101.3 kPa)	0.17764	0.19103	0.01671
Hexane(1) – Cyclohexane(2) system (101.33 kPa)	0.00563	0.00865	0.00201

IV. RESULTS AND DISCUSSIONS

In this study, 7 binary systems have been considered. Out of these 7 systems, 6 systems are azeotropes and 1 is a binary mixture. The values of the output (y1) data has been calculated using the different models (ANN, Margules, van Laar). The values of Root Mean Squared Deviation (RSMD) have been calculated for various systems. It is obtained as shown in the table given below.

Table 5.1. Root Squared Mean Deviation from experimental values

The Root Squared Mean Deviation (RSMD) calculated for all the systems considered have been represented in the table above. The comparison of the RSMD values of all the models, shows that the deviations shown by the ANN model is lesser as compared to the Margules and van Laar models. In other words, for all the systems considered, the values of mole fraction of the vapour phase predicted by the ANN model show lesser deviations from the experimental values as compared to the thermodynamic models. The graphical representation (Appendix) of the mole fractions in the liquid phase versus the mole fractions in the vapour phase shows the closer proximity of ANN values to the experimental values as compared to those of thermodynamic models considered.

In the Methanol (1) – Acetone (2) system, the RSMD value obtained for the mole fractions of vapour phase calculated by the ANN model is the least. Also in the Chloroform(1) – Ethanol(2) system, Methanol(1) – Hexane(2) system, Methanol(1) – Benzene (2) system, Benzene(1) – Acetonitrile(2) system, Water(1) – m-Xylene(2) system, the values of the RSMD obtained for ANN values is considerably lesser than the van Laar and the Margules model values. Hence, the VLE data prediction by ANN for these systems is much accurate as compared to the thermodynamic models considered. While, for Hexane(1) – Cyclohexane(2) system the accuracy of prediction of ANN model is as much as the thermodynamic models.

V. CONCLUSIONS

An artificial neural network (ANN) model has been developed and used for the prediction of Vapour Liquid Equilibrium (VLE) data. This model developed is a non-linear, non-thermodynamic model. The results of VLE data prediction using this model for various systems show satisfactory results. For all the systems considered the data predicted by the ANN shows closer agreement with the experimental literature as compared to the Margules and vaanLaar models, especially for the azeotropic systems considered, the VLE data prediction by the ANN model is better than that predicted by the Margules and van Laar models.

APPENDICES

APPENDIX A: MATLAB CODE

I) BASIC MATLAB CODE FOR METHANOL (1) – ACETONE (2) SYSTEM

```
clc, clear;
% asking the user the Excel file name containing the data
R=input('training set file name:','s');
T=input('target file name: ','s');
S=input('input set file name: ','s');
```

```
%Reading the Excel file
RR=xlsread(R);
TT=xlsread(T);
SS=xlsread(S);
 % Taking transpose of the input matrix
RR1 = RR';
TT1 = TT';
SS1 =SS';
 % Defining function and parameters for the creation of the ANN
net = newff(minmax(RR1),[2 4 1], {'logsig'  'logsig'      'purelin'},' trainrp'; net.trainParam.epochs =100000;
net.trainParam.show=5000;
net.trainParam.lr=0.1;
net.trainParam.lr_inc = 1.05;
net.trainParam.deltamax=100.0;
net.trainParam.goal = 1e-6;
%training the network
net = train(net,RR1,TT1);
%saving the net
save(['m1.mat'],'net');
% Simulation of new sets of data
y=sim(net,SS1)
```

II) MATLAB CODE USED FOR ALL THE SYSTEMS

```
clc,
clear;
R=input('training set file name:','s');
T=input('target file name: ','s');
S=input('input set file name: ','s');
RR=xlsread(R);
TT=xlsread(T);
SS=xlsread(S);
RR1=RR';
TT1=TT';
SS1=SS';
% new network with 1 input layer, 2 hidden layer with 30 neurons each
% output layer with one neuron
% training function ‘trainrp’ Resilient Propagation
%activation function in each layer Log-sigmoid (logsig)
net = newff(minmax(RR1),[30 30 1], {'logsig' 'logsig' 'logsig'},'trainrp');
net.trainParam.epochs =100000; % maximum epochs/iterations
net.trainParam.show=5000; %parameters shown on matlab screen
net.trainParam.lr=0.1;% learning rate
net.trainParam.min_grad = 1e-10; %minimum gradient
net.trainParam.lr_inc = 1.05; %learning rate increment
net.trainParam.deltamax=100.0; %maximum weight increase
net.trainParam.goal = 1e-9; % goal to be achieved (MSE)
net = train(net,RR1,TT1); %network training command
save(['NETNAME.mat'],'net'); % saving the neural network developed
y=sim(net,SS1)  % simulating values using the developed network
```

APPENDIX B: TRAINING DATA FOR ALL THE SYSTEMS

These sets of data points have been obtained from the Explorer edition of the Dortmund Data Bank (DDBST). The y1 data is the target output for network. And P, x1 or T, x1data is the input.

TABLE B.1 CHLOROFORM(1) - ETHANOL (2) SYSTEM (308.15 K) – TRAINING DATA

P [kPa]	x1	y1
13.703	0	0
13.982	0.0062	0.0254
14.84	0.0241	0.0991
15.147	0.0297	0.121
16.775	0.0594	0.2343
19.766	0.1109	0.3885
23.678	0.173	0.5304
27.422	0.2361	0.6207
30.563	0.3014	0.687
31.531	0.3227	0.7009
33.783	0.3845	0.737
34.035	0.3922	0.7412
35.684	0.4384	0.7646
38.923	0.6185	0.8181
39.587	0.6783	0.8327
40.403	0.7746	0.8554
40.715	0.8265	0.8698
40.83	0.8483	0.8783
40.803	0.9315	0.9161
40.646	0.956	0.9363
40.553	0.9586	0.9385
40.489	0.96	0.9403
39.345	1	1

TABLE B.2 METHANOL (1) – ACETONE (2) SYSTEM (101.325kPa) – TRAINING DATA

T(K)	x1	y1
328.82	0.07	0.082
328.46	0.181	0.188
328.39	0.217	0.218
328.45	0.265	0.255
328.54	0.34	0.311
328.89	0.406	0.356
329.3	0.481	0.406
330.05	0.593	0.486
330.2	0.606	0.496
330.44	0.631	0.515
331.47	0.719	0.59
331.64	0.737	0.608
332.12	0.771	0.643
332.72	0.805	0.681
334.68	0.9	0.809
335.36	0.926	0.852
335.94	0.947	0.89
336.84	0.976	0.947

TABLE B.3 METHANOL (1) – HEXANE (2) SYSTEM (333.15 K) – TRAINING DATA

P [kPa]	x1	y1
76.047	0	0
107.484	0.005	0.285
130.869	0.016	0.407
138.002	0.033	0.438
141.868	0.06	0.463
145.108	0.098	0.482
147.148	0.145	0.488
148.308	0.175	0.496
148.988	0.218	0.499
149.348	0.399	0.512
149.641	0.522	0.517
149.494	0.57	0.52
149.388	0.613	0.521
149.308	0.657	0.522
148.961	0.79	0.524
147.761	0.828	0.533
145.028	0.877	0.54
144.241	0.883	0.543
138.495	0.919	0.574
131.603	0.938	0.616
126.47	0.951	0.645
121.59	0.962	0.681
111.484	0.979	0.752
110.471	0.98	0.763
83.9	1	1

TABLE B.4 METHANOL(1) – BENZENE(2) SYSTEM (101.33 kPa) – TRAINING DATA

T [K]	x1	y1
351.76	0.002	0.041
350.8	0.0028	0.066
350.67	0.003	0.0684
350.43	0.003	0.0812
350.37	0.0041	0.082
348.45	0.0058	0.1386
346.92	0.0134	0.1924
346.21	0.019	0.206
344.11	0.0276	0.2658
341.1	0.046	0.3392
340.9	0.048	0.344
334.23	0.1437	0.4998
334.19	0.17	0.518
333.07	0.1858	0.5212
332.55	0.276	0.5442
331.58	0.364	0.5736
331.3	0.4558	0.587
331.19	0.524	0.5981
331.12	0.637	0.62
331.13	0.6739	0.629
331.14	0.68	0.6322
331.2	0.7024	0.6408
331.21	0.7071	0.6416
331.37	0.744	0.6531
331.6	0.7768	0.6691
331.8	0.8031	0.6838
332.14	0.838	0.709
332.75	0.8754	0.748
333.75	0.9178	0.8002
334.05	0.925	0.8118
334.79	0.9424	0.851
335.75	0.9652	0.8998
336.72	0.9849	0.9492
336.87	0.988	0.9572
337.1	0.9918	0.9684
337.36	0.9949	0.9799

TABLE B.5
BENZENE(1) – ACETONITRILE(2) SYSTEM (293.15 K) – TRAINING DATA

P [kPa]	x1	y1
9.799	0.018	0.054
10.106	0.033	0.096
10.599	0.063	0.161
11.106	0.098	0.221
11.386	0.128	0.256
11.719	0.168	0.3
11.852	0.187	0.321
11.932	0.198	0.33
12.226	0.256	0.376
12.359	0.284	0.395
12.439	0.31	0.412
12.466	0.32	0.417
12.612	0.368	0.442
12.666	0.424	0.473
12.679	0.446	0.484
12.692	0.446	0.486
12.719	0.47	0.498
12.692	0.481	0.501
12.746	0.511	0.518
12.746	0.535	0.532
12.719	0.572	0.551
12.719	0.607	0.571
12.666	0.648	0.593
12.639	0.657	0.597
12.612	0.679	0.612
12.519	0.71	0.632
12.186	0.793	0.695
12.119	0.815	0.709
12.012	0.841	0.734
11.866	0.857	0.757
11.706	0.874	0.767
11.252	0.92	0.829
11.106	0.933	0.849
11.026	0.939	0.86
10.879	0.953	0.883
10.506	0.975	0.93
10.319	0.983	0.95
10.186	0.991	0.972

TABLE B.6
WATER(1) - m-XYLENE(2) SYSTEM (101.3 kPa) – TRAINING DATA

T [K]	x1	y1
408.15	0.0022	0.1216
407.59	0.0029	0.1396
399.85	0.0082	0.3082
395.63	0.0122	0.4013
391.15	0.0163	0.4807
382.35	0.0231	0.6023
377.35	0.0349	0.6704
374.89	0.0864	0.6921
372.65	0.1294	0.7126
369.87	0.2435	0.7378
367.16	0.9992	0.7924
367.46	0.9994	0.8138
368.06	0.9996	0.8313
369.32	0.9998	0.8723
372.84	1	0.9897

TABLE B.7
HEXANE (1) – CYCLOHEXANE(2) SYSTEM (101.33 kPa) –TRAINING DATA

T [K]	x1	y1
353.95	0	0
353.75	0.008	0.018
353.5	0.019	0.032
352.9	0.063	0.0975
352.45	0.094	0.135
351.8	0.133	0.187
350.3	0.239	0.316
349.65	0.287	0.373
349.2	0.318	0.4
348.5	0.369	0.4575
347.65	0.443	0.5335
347.4	0.462	0.553
346.9	0.498	0.5835
346.55	0.536	0.6175
346.4	0.549	0.63
345.9	0.596	0.673
345.25	0.655	0.724
344.85	0.693	0.7575
344.1	0.757	0.808
344.05	0.769	0.818
343.95	0.777	0.8295
343.7	0.807	0.852
343.5	0.831	0.8715
343.3	0.85	0.883
343.1	0.874	0.902
342.5	0.935	0.95
342.2	0.969	0.976
341.95	1	1

APPENDIX B: GRAPHS (x1 v/s y1)

GRAPHICAL REPRESENTATION OF OUTPUT

The calculated vapour mole fraction obtained from the various models is plotted against the liquid mole fractions to compare between the values calculated by the different models.

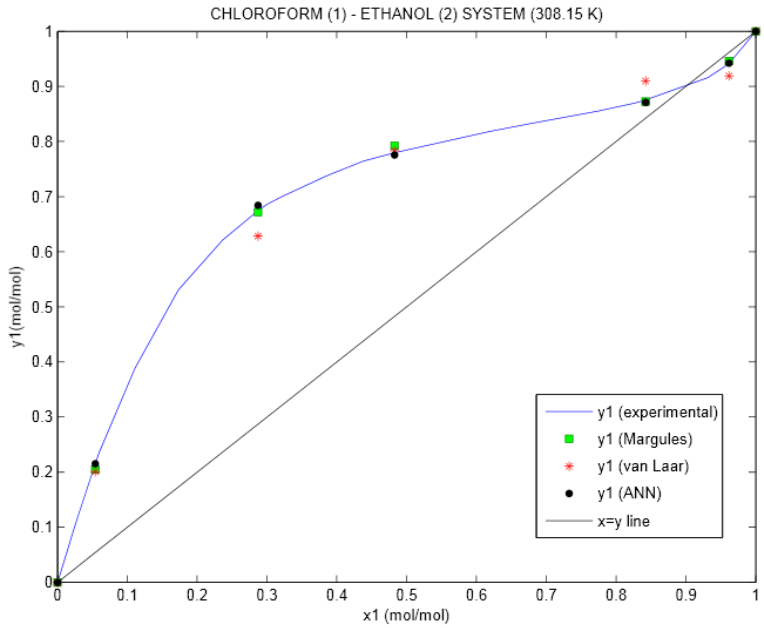


FIGURE B.1: Chloroform (1) -Ethanol (2) System (308.15 K) – (x-y Graph)

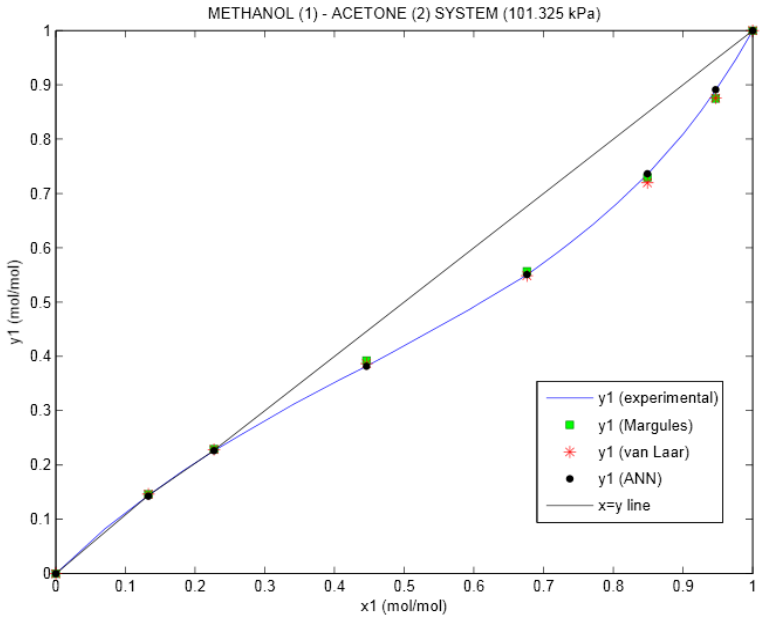


FIGURE B.2 : Methanol (1) – Acetone (2) System (101.325 kPa) – (x-y Graph)

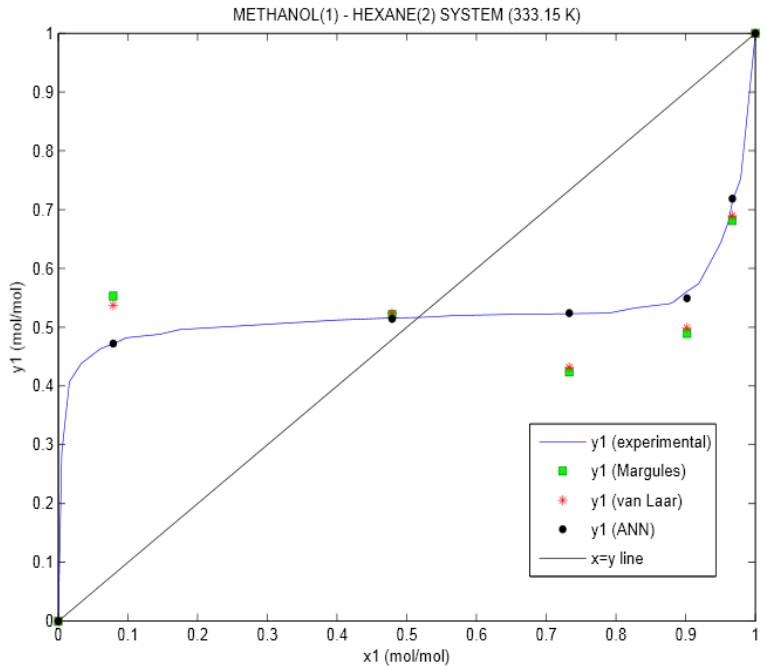


FIGURE B.3: Methanol (1) – Hexane (2) System (333.15 K) – (x-y Graph)

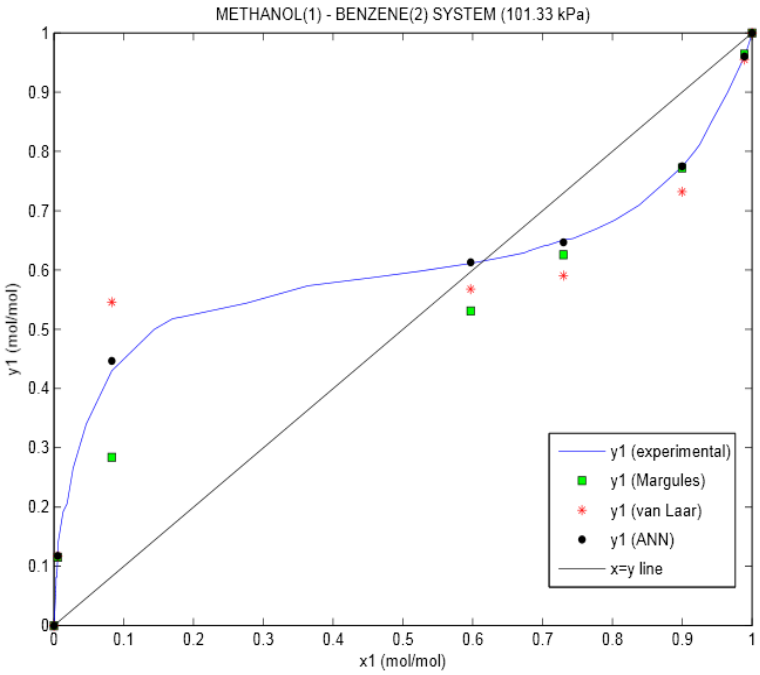


FIGURE B.4: Methanol (1) –Benzene (2) System (333.15 K) – (x-y Graph)

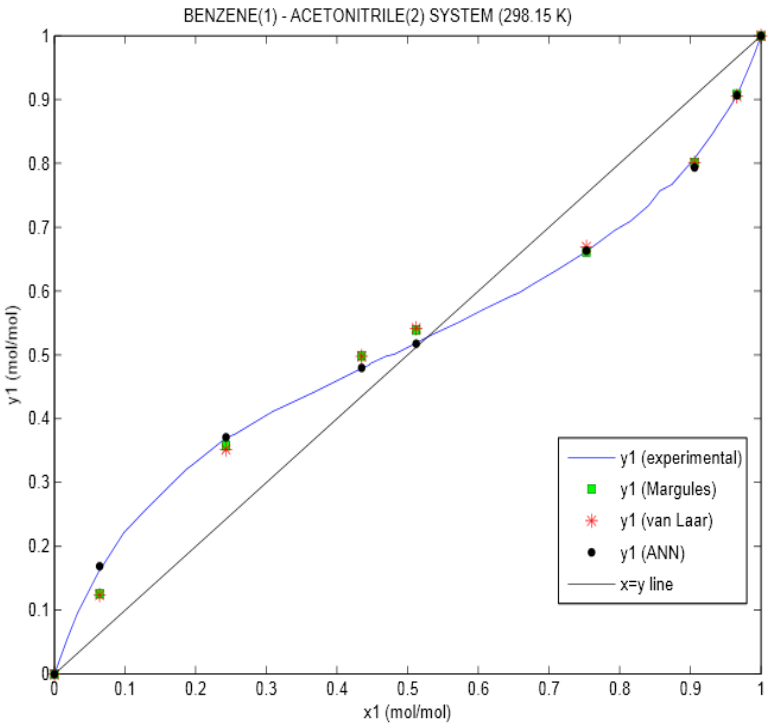


FIGURE B.5: Benzene (1) – Acetonitrile (2) System (293.15 K) – (x-y Graph)

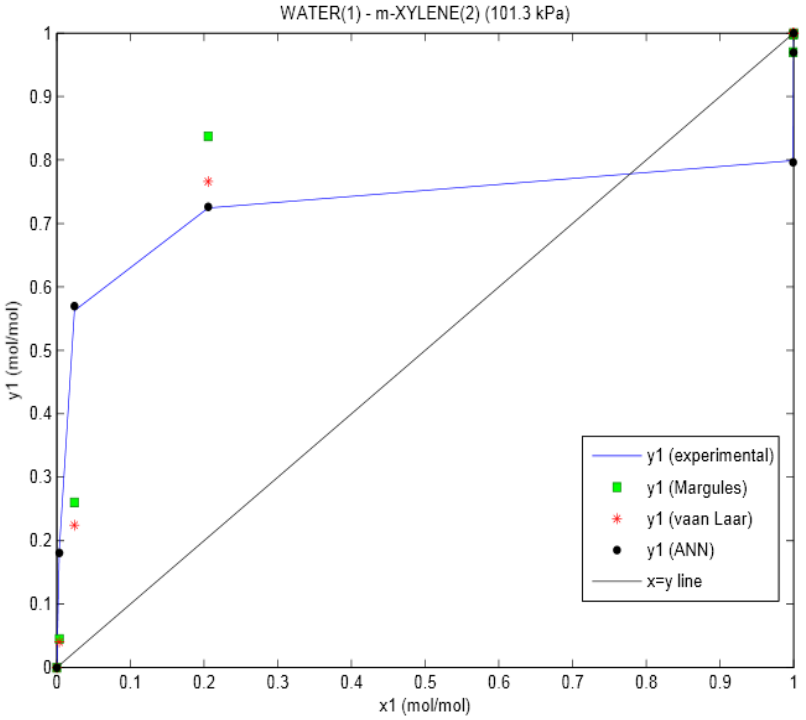


FIGURE B.6: Water (1) – m-Xylene (2) System (101.3 kPa) – (x-y Graph)

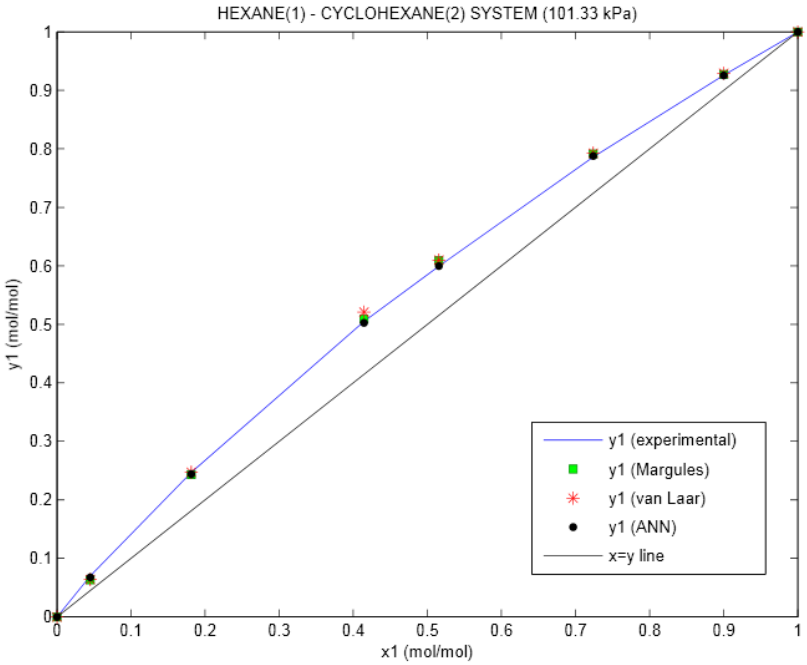


FIGURE B.7: Hexane (1) – Cyclohexane (2) System (101.33 kPa) – (x-y Graph)

TABLE C.1
CHLOROFORM(1) – ETHANOL(2) SYSTEM (308.15 K) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
P [kPa]	x1	y1	y1	y1	y1
16.471	0.0542	0.2154	0.2047	0.201	0.2151
30.006	0.2873	0.6747	0.672	0.6282	0.6842
36.536	0.4827	0.7797	0.7924	0.7831	0.7755
40.679	0.8423	0.8752	0.8722	0.91	0.871
40.518	0.9616	0.9414	0.9459	0.9189	0.9424
		RSMD	0.0079	0.0286	0.005

TABLE C.2
METHANOL(1) – ACETONE(2) SYSTEM (101.325 kPa) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
T(K)	x1	y1	y1	y1	y1
328.58	0.133	0.144	0.1461	0.1463	0.1427
328.4	0.227	0.226	0.2293	0.2284	0.2265
329.11	0.446	0.382	0.3924	0.3868	0.3818
330.84	0.676	0.55	0.5566	0.5486	0.5505
333.56	0.849	0.735	0.7295	0.7197	0.7365
335.94	0.947	0.89	0.8747	0.8759	0.8912
		RSMD	0.0085	0.0089	0.000993

TABLE C.3
METHANOL(1) – HEXANE(2) SYSTEM (333.15 K) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
P [kPa]	x1	y1	y1	y1	y1
143.468	0.079	0.472	0.553	0.5373	0.4724
149.561	0.479	0.516	0.5212	0.5234	0.5141
149.254	0.733	0.523	0.4238	0.4318	0.5241
141.762	0.902	0.561	0.4893	0.4987	0.5494
116.67	0.967	0.714	0.6815	0.6895	0.7186
		RSMD	0.0673	0.0585	0.0057

TABLE C.4
METHANOL(1) – BENZENE(2) SYSTEM (101.33 kPa) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
T [K]	x1	y1	y1	y1	y1
349.2	0.0055	0.1172	0.1153	0.1176	0.11776
337.17	0.083	0.43	0.2842	0.5458	0.44651
331.12	0.597	0.6114	0.5309	0.5677	0.61283
331.28	0.7298	0.6509	0.6257	0.5903	0.64674
333.33	0.8998	0.7742	0.7723	0.7319	0.77495
336.93	0.989	0.961	0.9642	0.955	0.96013
		RSMD	0.0688	0.0589	0.006995

TABLE C.5
BENZENE(1) – ACETONITRILE(2) SYSTEM (293.15 K) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
P	x1	y1	y1	y1	y1
10.639	0.064	0.162	0.1257	0.1239	0.169
12.186	0.243	0.37	0.3577	0.352	0.3709
12.679	0.435	0.479	0.4981	0.4976	0.4798
12.732	0.512	0.519	0.5391	0.5418	0.5177
12.359	0.753	0.662	0.6615	0.6693	0.6631
11.386	0.906	0.808	0.801	0.801	0.7937
10.692	0.966	0.906	0.908	0.905	0.9068
		RSMD	0.0181	0.0198	0.0061

TABLE C.6
WATER(1) – m-XYLENE(1) SYSTEM (101.3 kPa) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
T [K]	x1	y1	y1	y1	y1
405.01	0.0035	0.1875	0.0451	0.0404	0.18079
385.91	0.0242	0.5627	0.2601	0.2246	0.56964
370.25	0.2057	0.7243	0.8373	0.7659	0.72621
367.2	0.9993	0.7992	0.9698	1	0.79638
371.21	0.9999	0.9334	0.997	1	0.96934
		RSMD	0.1776	0.191	0.016712

TABLE C.7
HEXANE(1) – CYCLOHEXANE(2) SYSTEM (101.33 kPa) – OUTPUT DATA

		EXPERIMENTAL	MARGULES	VAN LAAR	ANN
T [K]	x1	y1	y1	y1	y1
353.15	0.045	0.07	0.0632	0.0639	0.0676
351.15	0.1815	0.247	0.2433	0.2473	0.2442
347.95	0.4145	0.505	0.509	0.5209	0.5024
346.75	0.5155	0.5995	0.6088	0.6098	0.5996
344.55	0.724	0.786	0.7912	0.7926	0.7879
342.9	0.9	0.926	0.9272	0.9291	0.9254
		RSMD	0.0056	0.0086	0.002

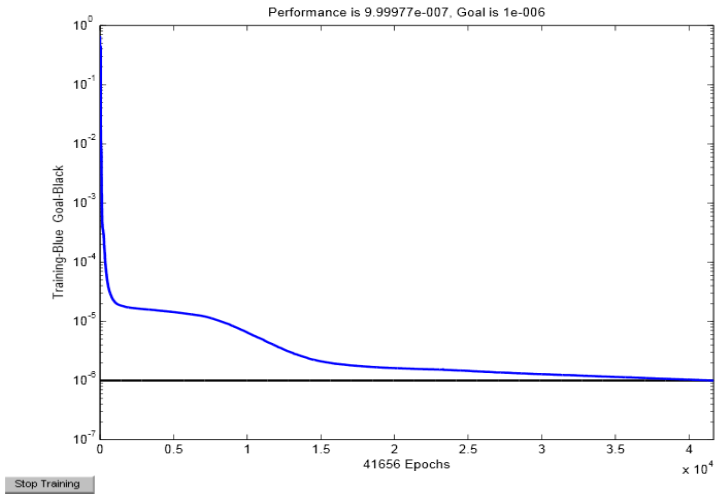
APPENDIX C: TRAINING OF DATA AND TRAINING GRAPH

The Methanol (1) – Acetone(1) system was trained with the training data given in appendix (B.2) by implementing the Matlab code given in appendix (A.1)

The training output obtained is training set file name: maa-input target file name: maa-target input set file name: maa-test

TRAINRP, Epoch 0/100000, MSE 0.660786/1e-006, Gradient 36.7999/1e-006

TRAINRP, Epoch 5000/100000, MSE 1.43502e-005/1e-006, Gradient 0.00193372/1e-006
TRAINRP, Epoch 10000/100000, MSE 6.49147e-006/1e-006, Gradient 0.0037737/1e-006
TRAINRP, Epoch 15000/100000, MSE 2.11517e-006/1e-006, Gradient 0.000254585/1e-006
TRAINRP, Epoch 20000/100000, MSE 1.62195e-006/1e-006, Gradient 0.000552213/1e-006
TRAINRP, Epoch 25000/100000, MSE 1.45938e-006/1e-006, Gradient 0.000329364/1e-006
TRAINRP, Epoch 30000/100000, MSE 1.27897e-006/1e-006, Gradient 0.000208013/1e-006
TRAINRP, Epoch 35000/100000, MSE 1.15176e-006/1e-006, Gradient 0.000179174/1e-006
TRAINRP, Epoch 40000/100000, MSE 1.03867e-006/1e-006, Gradient 0.00016271/1e-006
TRAINRP, Epoch 41656/100000, MSE 9.99977e-007/1e-006, Gradient 0.000333166/1e-006
TRAINRP, Performance goal met.
The training graph obtained is



C.1. Training graph for Methanol (1) – Acetone (2) system 101.325kPa

REFERENCES

[1]. Mehmet Bilgin, I.MetinHasdemir and OguzhanOztas, "The use of neural networks on VLE data prediction", Journal of Scientific and Industrial Research, April 2004, Volume: 63, Pages: 336-343.
[2]. Mehmet Bilgin, "Isobaric vapour–liquid equilibrium calculations of binary systems using a neural network", Journal Of The Serbian Chemical Society, 2004, Volume: 69, Number:8-9, Pages: 669 – 674.
[3]. AbdolrezaMoghadassi, FahimeParvizian, SayedMohsenHosseini, "A New Approach Based on Artificial Neural Networks for Prediction of High Pressure Vapor-liquid Equilibrium", Australian Journal of Basic and Applied Sciences, 2009, Volume: 3 Number: 3, Pages: 1851-1862.
[4]. A. R. Moghadassi, M. R. Nikkholgh, S. M. Hosseini, F. Parvizian and A. Sanaeirad, "Vapour Liquid Equilibrium Data Prediction For Binary Systems Containing Propane", ARPN Journal of Engineering and Applied Sciences, September 2011, Volume: 6, Number: 9, Pages: 94 - 103.
[5]. A.Ghaemi, Sh.Shahhoseini, M.GhannadiMarageh, M.Farrokhi, "Prediction of Vapour Liquid Equilibrium for Aqueous Solution of Electrolytes Using Artificial Neural Networks", Journal of Applied Sciences, 2008, 8(4):615-621.
[6]. ShekharPandharipande*et. al.*, "Modeling Combined VLE Of Four Quaternary Mixtures Using Artificial Neural Network", International Journal of Advances in Engineering, Science and Technology (IJAEST), May-July 2012, Vol. 2 No. 2.
[7]. Maria C. Iliuta, Ion Iliut, FaicalLarachi, "Vapour liquid equilibrium data analysis for mixed solvent electrolyte systems using neural network models", Chemical Engineering Science, 2000, Volume:55, Pages: 2813-2825.
[8]. L. Govindarajan and PL. Sabarathinam, "Prediction of Vapor-liquid Equilibrium Data by Using Radial Basis Neural Networks", Chemical Biochemical Engineering Journal, 2006, Volume: 20, Number: 6, Pages: 319-323.
[9]. S.L.Pandharipande*et. al*, "Modeling Combined VLE of Ten Binary Mixtures using Artificial Neural Networks", International Conference on Intuitive Systems & Solutions(ICISS) 2012, Proceedings published by International Journal of Computer Applications (IJCA).
[10]. FariborzNasri, HiuaDaraei, TahmasbHatami And AfshinMaleki, "Phase Equilibrium of Binary System Carbon Dioxide - Methanol at High Pressure Using Artificial Neural Network", April 2012.
[11]. R. Rajesh, S. Chattopadhyay, M. Kundu, "Prediction Of Equilibrium Solubility Of CO2 in Aqueous Alkanolamines Through Artificial Neural Network", Paper presented at CHEMECA'06, Auckland, New Zealand from 17-20 September 2006.
[12]. ManishVashishtha, "Application of Artificial Neural Networks in the Prediction of Vapour Liquid Equilibrium Data", Proceedings 25th European Conference on Modeling and Simulation (ECMS), 2011.
[13]. J.M.Smith, H.C.Van Ness, Introduction to Chemical Engineering Thermodynamics, Seventh Edition, Tata McGraw Hill Publications.
[14]. S.N.Sivanandam, Introduction to Neural Networks using Matlab 6.0, Tata McGraw Hill Publications.
[15]. Y.V.C. Rao, Chemical Engineering Thermodynamics, University Press.
[16]. MartienRiedmiller, "A Direct Adaptive Method for Faster BackpropagationLearning : The RPROP Algorithm".
[17]. MartienRiedmiller, "RPROP – Description and Implementation details".
[18]. Frauke Gunther and Stefan Fritsch, 'Neuralnet: Training of Neural Networks', The R Journal, June 2010, Pages: 30-38.